



iCluster User Guide V9.2.1

Notices

Copyright

© 1996-2024 Rocket Software, Inc. or its affiliates. All Rights Reserved.

Trademarks

Rocket is a registered trademark of Rocket Software, Inc. For a list of Rocket registered trademarks go to: www.rocketsoftware.com/about/legal. All other products or services mentioned in this document may be covered by the trademarks, service marks, or product names of their respective owners.

Examples

This information might contain examples of data and reports. 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.

License agreement

This software and the associated documentation are proprietary and confidential to Rocket Software, Inc. or its affiliates, are furnished under license, and may be used and copied only in accordance with the terms of such license.

Note: This product may contain encryption technology. Many countries prohibit or restrict the use, import, or export of encryption technologies, and current use, import, and export regulations should be followed when exporting this product.

Corporate information

Rocket Software, Inc. develops enterprise infrastructure products in four key areas: storage, networks, and compliance; database servers and tools; business information and analytics; and application development, integration, and modernization.

Website: www.rocketsoftware.com

Rocket Global Headquarters

77 4th Avenue, Suite 100

Waltham, MA 02451-1468

USA

To contact Rocket Software by telephone for any reason, including obtaining pre-sales information and technical support, use one of the following telephone numbers.

Country and Toll-free telephone number

- United States: 1-855-577-4323
- Australia: 1-800-823-405
- Belgium: 0800-266-65
- Canada: 1-855-577-4323
- China: 400-120-9242
- France: 08-05-08-05-62
- Germany: 0800-180-0882
- Italy: 800-878-295
- Japan: 0800-170-5464
- Netherlands: 0-800-022-2961
- New Zealand: 0800-003210
- South Africa: 0-800-980-818
- United Kingdom: 0800-520-0439

Contacting Technical Support

The Rocket Community is the primary method of obtaining support. If you have current support and maintenance agreements with Rocket Software, you can access the Rocket Community and report a problem, download an update, or read answers to FAQs. To log in to the Rocket Community or to request a Rocket Community account, go to www.rocketsoftware.com/support. In addition to using the Rocket Community to obtain support, you can use one of the telephone numbers that are listed above or send an email to support@rocketsoftware.com.

Table of contents

Rocket® iCluster overview	14
Starting iCluster.....	14
Accessing the job logs	15
Starting iCluster after a system IPL	15
Ending iCluster before a system IPL.....	16
iCluster user registration.....	18
Registering iCluster users	18
Changing registered users.....	19
Removing registered users	19
Using the Work with Users screen	20
Server authentication for secure DDM connections	20
Verifying DDM authentication.....	20
Enabling DDM access	21
Node operations	25
Adding a node to a cluster.....	25
Viewing the node settings	26
Removing a node from a cluster.....	26
Changing node attributes	26
Renaming a node	27
Assigning a failover message queue to a node	28
Specifying a failover user exit on a node.....	28
Changing the status of a node	29
Viewing nodes in the cluster	29
Checking for communications failure on a node.....	30
Verifying communications between nodes in a cluster	30
Changing the IP address of a node	31
Group operations	32
Setting up groups	32
Adding a group	32
Changing group attributes.....	34
Removing a group.....	34
Viewing groups in a cluster	34
Working with recovery domains.....	35
Adding a backup node to the recovery domain of a group.....	35
Removing a backup node from the recovery domain	35
Changing the primary node to the backup node.....	36
Viewing current settings for a specified group.....	36
Starting replication.....	37
Initial synchronization	37
Start replication using marked journal positions.....	37
Starting replication groups	38
Resilient devices	39

Adding a switchable device entry to a group	39
Viewing the attributes of a switchable device replication group.....	39
Changing the attributes of a switchable device entry in a group	40
Removing a switchable device entry from a group	40
Object specifiers and path object specifiers	41
Understanding object specifiers.....	41
Generic object specifiers	41
Rules of precedence for object specifiers	42
Working with object specifiers	42
Selecting an object specifier to a replication group.....	42
Selecting a directory entry to a group	46
Selecting a SQL object specifier to a group	47
Changing object specifier attributes.....	48
Deselecting an object specifier from a group	51
Deselecting a directory entry from a group.....	52
Reconfiguring object specifiers	52
Viewing object specifiers.....	53
Path object specifiers	53
Naming conventions for path object specifiers	54
Non-journalized and journalized path object specifiers	54
Working with path object specifiers.....	54
Adding a path object specifier to a group	55
Removing a path object specifier	56
Changing path object specifier attributes	56
Replicating database *FILE objects	57
Replicating database *FILE objects	59
Replicating BSF objects	61
Replicating configuration objects	63
Replicating user profiles	64
Replicating LOBs	64
Large object (LOB) constraints	65
Replicating triggers	65
Using the SAVFEXIT user exit program when replicating save files	65
Replicating objects with save files and the SAVFEXIT program	67
Replicating SQL objects.....	67
Replicating QDLS or DLO objects	69
Including the QDLS folder and QDLS sub-folders in replication	69
Replicating WebSphere MQ objects	70
Creating a local relational database directory entry.....	71
Enabling WebSphere MQ replication.....	71
WebSphere MQ recommendations	71
Supported object types for replication.....	72
Configuring high availability for external applications.....	77
Configuring replication for external applications	77
Using iCluster with external applications	77

Recreating groups after Lifecycle Manager upgrade from Version 7.5	79
Configuring replication for Rocket Lifecycle Manager.....	81
Configuring replication of objects stored outside of Lifecycle Manager.....	81
Staging objects.....	82
Clearing the staging store	83
Changing the size of a staging store	83
Changing the name of a staging store library	83
Starting the replication apply process.....	84
Starting the replication apply process for groups on the backup node.....	85
Ending the replication apply process.....	85
Ending the replication apply process on a backup node	86
Locked objects and locked object contention	86
Commitment control	87
Specifying the commitment control level for a group.....	87
Cluster operations.....	89
Starting cluster operations at a node	89
Starting cluster operations for groups	89
Ending cluster operations for groups with a specific primary node	90
Restarting cluster node operations with current machine rejoining the cluster	91
Ending cluster operations at a node	91
Ending cluster operations for all groups.....	91
Restarting iCluster after restarting a node	91
Upgrading the version of a cluster.....	92
Deleting a cluster	92
Suspended objects	93
Suspending an object.....	93
Suspending a BSF object	94
Activating objects	94
Activating a suspended object.....	94
Activating BSF objects	95
Automatic object reactivation	96
Removing user profiles from suspension.....	97
Journaling considerations	98
Uncommitted transactions	98
Remote journaling	99
Configuring remote journaling	100
Role switching with remote journals	101
Setting journal positions	101
Setting a journal start position	102
Marking the current journal positions for replication	103
Changing journal receivers.....	104
Displaying replication status of journal receivers.....	105
Verifying the audit journal.....	105
Synchronization points.....	106
Setting a synchronization point.....	106

Passing arguments to sync point user exit programs	107
Recovery checkpoints	108
Retrieving a recovery checkpoint from the command line	108
Performance of replication groups	108
Logging a journal entry	109
Sync check operations	110
Setting attributes for sync checks	110
Selecting sync check attributes for a group	111
Starting a sync check for a group	111
Starting a user-specified sync check	114
Starting a continuous sync check for a group	117
Ending a sync check	118
Viewing sync check results	118
Generating sync check reports	118
Purging sync check results	119
Event log	120
Displaying the event log	120
Clearing the event log	121
Failover and switchover mechanisms	123
What happens during a switchover or failover	123
Failover mechanisms	124
Key differences in failover mechanisms	124
SwitchOver System	126
User actions	126
Detecting node failures	126
Defining user exit programs	127
Detecting problems in processing a role switch	128
Node failure examples	129
Manual failover example	130
Automatic failover example	131
Performing a manual failover	132
Performing a manual failover after the primary node fails	133
Recovering from failovers	133
Restoring a group whose primary node has failed	133
Testing switchover for a single group	134
Changing an existing cluster to use SwitchOver System	135
Cluster limits for SwitchOver System	135
Cluster Resource Services failover mechanism	136
Clustering overview and concepts	136
iCluster and Cluster Resource Services	137
iCluster groups and role switching	137
ROLESWITCH parameter	138
Preparing the cluster to handle switchovers and failovers	139
Journal entry processing	139
Failover message queue	139

Line controller heartbeat	139
Passing arguments to role switch user exit programs	140
FAILED state overview	140
FAILED state scenarios	142
Detecting a FAILED state.....	142
Group failover	143
Recovery paths for a FAILED state.....	143
PARTITION state overview	146
False and True PARTITIONS	147
PARTITION state scenarios.....	147
Detecting a PARTITION state	148
Verifying a PARTITION state	148
Recovery paths for a PARTITION state	149
Restarting active groups	151
Restarting active groups with mirroring in the original direction with ROLESWITCH *NO	151
Restarting active groups with mirroring in the original direction with ROLESWITCH *YES	152
Restarting active groups with mirroring in the opposite direction with ROLESWITCH *NO	152
Restarting active groups with mirroring in the opposite direction with ROLESWITCH *YES	153
Rejoining non-functioning nodes	154
Rejoin a non-functioning node with a role switch	154
Rejoin a non-functioning node without a role switch	154
Removing a failed node from the cluster	155
Alertable messages for clustering	156
Switching the ownership of the iCluster metadata to another node	157
Configuring a new cluster to use Cluster Resource Services	157
Changing an existing cluster to use Cluster Resource Services	158
Modifying default cluster system values	159
Role switch operations	160
Changing a primary node to backup node	160
Starting a role switch	160
Role switch simulator	161
Simulating role switches.....	161
Ending role switch simulation	162
Monitoring	164
Monitoring overview.....	164
Understanding latency.....	165
Status Monitor views	165
Real Time Overall Latency view	166
Real Time Object Latency view	166
Real Time Object Position And Totals view	167
Real Time Object Throughput view.....	168
Status column information	168
Monitoring the replication environment.....	171
Performing tasks from the status monitors	172
Searching while monitoring	173

Configuring replication monitoring.....	173
Configuring email for replication monitoring alerts.....	174
Configuring replication monitoring alerts	175
Object and group status views	175
Work with Object Status views.....	175
Work with BSF/SQL Status views.....	177
Object status reason codes.....	177
Out-of-sync reason codes	180
Viewing and changing object status.....	184
Working with BSF and SQL objects.....	184
Historical Status Monitor views	185
Historical Latency view	185
Historical Object Position and Totals view.....	185
Historical Object Throughput view	186
Viewing the Historical Status Monitor	186
Changing the Historical Latency collection process	187
Deleting historical information from the Historical Latency	188
Reporting	189
iCluster Configuration Report (DMCFGRPT command).....	190
iCluster System Report (DMSYSINF command)	191
iCluster Group Status Report (DMGRPSTS command)	192
iCluster Sync Check Report (DMSCRPT command)	193
iCluster Replication Coverage Report (DMRPLCVRPT command)	194
iCluster Event Log Report (DMDSPLOG command).....	195
Files with Triggers Report (DMRPTTRG command)	195
iCluster Audit Report (DMAUDITRPT command)	196
Files with Constraints Report (DMRPTCST command)	198
Journal Analysis Report (DMANZJRN command)	199
Cross-Library Logical File Report (DMRPTXLF command).....	202
Object Counts Report (DMOBJCNT command).....	203
iCluster Suspended Objects Report (DMSUSOBRPT command)	204
iCluster commands.....	206
Node commands.....	206
DMADDNODE (Add Node command).....	207
DMCHGNODE (Change Node command).....	213
DMRMVNODENODE (Remove Node command)	219
DMRNMNODENODE (Rename Node command).....	220
DMDSPNODE (Display Node command)	221
DMWRKNODE (Work with Nodes command)	222
DMCHGNODIP (Change Node IP Address command)	223
DMINZNODE (Initialize Node for Recovery command)	224
Group commands.....	225
DMADDGRP (Add Group command).....	226
DMCHGGGRP (Change Group command)	243
DMRMVGRP (Remove Group command)	260

DMADDBACK (Add Backup Node to Recovery Domain command).....	261
DMRMVBACK (Remove Backup Node from Recovery Domain command).....	262
DMDSPGRP (Display Group command)	262
DMWRKGRP (Work with Groups command)	263
Native IBM i object commands	266
DMSELOBJ (Select Objects to Group command).....	266
DMDSELOBJ (Deselect Objects from Group command)	273
DMCHGOBJSL (Change Object Selection command)	275
DMWRKOBJ (Work with Object Specifiers command)	282
DMSNDOBJ (Send Object Immediately command).....	283
DMSELDIRE (Select Directory Entry command).....	286
DMDSELDIRE (Deselect Directory Entry command).....	288
DMSUSDIR (Suspend Directory Entry command).....	289
DMACTDIRE (Activate Directory Entry command)	290
SQL object commands	292
DMSELSQLO (Select SQL Objects command)	292
DMDSELSQLO (Deselect SQL Objects from Group command).....	294
DMCHGSQLSL (Change SQL Object Selection command).....	296
DMWRKSQLO (Work with SQL Object Specifiers command)	298
Byte Stream File commands	299
DMADDBSF (Add Path Specifier to Group command).....	300
DMRMVBSF (Remove Path Specifier from Group command).....	304
DMCHGBSF (Change Path Specifier to Group command).....	306
DMWRKBSF (Work with Path Object Specifiers command)	309
External application commands	310
CFGHAXAPP (Configure HA for External Application command)	310
RMVHAXAPP (Remove HA for External Application command)	313
Administration commands	314
DMSETSVAL (Set Cluster System Values) command.....	314
DMCHGTIME (Change System Date and Time command).....	337
DMCLRLOG (Clear Cluster Event Log command)	339
HAPNGTCP (Ping Using TCP command).....	341
STRHATCP (Start TCP/IP Listener command)	342
WRKHAJOB (Work with Active Cluster Jobs command)	343
DMDLTCLSTR (Delete Cluster command).....	344
DMSETMSTR (Set Metadata Owner command).....	345
Report commands	346
DMCFGRT (Configuration Report command)	346
DMSYSINF (System Information command)	347
DMGRPSTS (Group Status Report command)	349
DMSCRPT (Sync Check Report command)	350
DMRPLCVRPT (iCluster Replication Coverage command)	352
DMDSPLLOG (Display Cluster Event Log command)	354
DMRPTRG (Report Files with Triggers command).....	359
DMRPTCST (Report Files with Constraints command)	360

DMANZJRN (Analyze Journals command).....	361
DMRPTXLF (Report Cross-library LFs command)	363
DMOBJCNT (Report Object Counts command)	364
DMAUDITRPT (Audit Report command).....	366
DMSUSOBRPT (Suspended Object Report command). .	367
Cluster operation commands	368
DMSTRCST (Start iCluster Processes command).....	369
DMENDCST (End iCluster Processes command).....	371
DMSTRNODE (Start Node command).....	374
DMENDNODE (End Node command)	375
DMREJOIN (Rejoin Cluster command)	376
DMSTRGRP (Start Group command).....	377
DMENDGRP (End Group command)	381
DMSTRSWO (Switchover Group command).....	383
DMSETPRIM (Prepare Primary command).....	385
DMCHGROLE (Change Primary command)	386
DMSWTCHRDY (Switchover Ready command)	388
RTVSWCHRDY (Retrieve Switchover Ready Status command)	390
DMSTRRSS (Start Roleswitch Simulator command)	392
DMENDRSS (End Roleswitch Simulator command)	393
DMINZSWO (Initialize System Switchover command)	395
DMFNZSWO (Finalize System Switchover command)	398
DMGENEXC (Generate Command Exceptions command).....	402
Replication operation commands	403
DMSETPOS (Set Journal Start Position command)	403
DMMRKPOS (Mark Journal Positions command)	407
DSPHAPOS (Display Journal Information command)	410
RTVHAPOS (Retrieve iCluster Journal Position command)	411
VFYHAJRN (Verify Audit Journal command)	414
DMSTRAPY (Start Replication Apply Process command)	415
DMENDAPY (End Replication Apply Process command)	417
RTVAPYSTS (Retrieve Apply Status command)	419
DMACTOBJ (Activate Object command)	421
DMACTBSF (Activate BSF Object command)	424
DMACTOOS (Activate Out-of-Sync Objects command)	425
DMACTSUS (Activate Suspended Objects command)	428
DMACTSQLO (Activate SQL Object command)	430
DMSUSOBJ (Suspend Object command)	431
DMSUSBSF (Suspend BSF Object command)	433
CRTCFGOBJ (Create Configuration Objects command)	434
INITHAOBJ (Initialize Objects command)	436
DMLOGENT (Log Journal Entry command)	437
DMACTSPLF (Activate Suspended Spooled Files command)	438
DMWRKSSPLF (Work with Suspended Spooled Files command)	440
RTVRCVPT (Retrieve Recovery Checkpoint command)	441

RTVRCVPTR (Retrieve Recovery Checkpoint (CL Program) command)	443
SYNCHATRG (Synchronize Triggers and SQL Triggers command)	445
SYNCHACST (Synchronize constraints command)	446
DMENDRFSH (End Current Refresh command)	448
Monitor commands	450
WRKCSMON (Work with Full Cluster Monitor command)	450
WRKHASMON (Work with Status Monitor on Primary Node command)	451
WRKHATMON (Work with Status Monitor on Backup Node command).....	452
DSPHASMON (Display Source Monitor command).....	453
CHGHASMON (Change History Monitor on Primary Node command).....	454
PRGHASMON (Purge History Monitor on Primary Node command).....	456
WRKHAOBJST (Work with Object Status command).....	457
WRKHABSFST (Work with BSF Status command)	458
DMWRKOBJST (Work with Object Status by Group command)	460
Sync check commands	461
SELSCATTR (Select Sync Check Attributes command)	461
DMSCATTR (Select Sync Check Attributes for a Group command)	470
DMSTRSC (Start Group Sync Check command).....	478
DMSTRSCUSR (Start User-specified Sync Check command)	484
STRCNSC (Start Continuous Sync Check command)	490
DMSCRPT (Sync Check Report command).....	492
DSPHASC (Display Sync Check Results command).....	494
ENDHASC (End Sync Check command).....	494
PRGHASC (Purge Sync Check Results command)	495
DMSYNCACT (Sync Check and Activate command)	496
STRHASC (Start Group Sync Check command)	497
STRHASCUSR (Start User-Specified Sync Check command)	503
Registered iCluster user commands	509
DMWRKUSR (Work with Users command).....	509
DMADDUSR (Add User command)	510
DMCHGUSR (Change User command)	512
DMRMVUSR (Remove User command).....	513
Utility commands	514
DMAUTOCFG (Auto-configure iCluster command).....	514
DMMONHA (Monitor iCluster Replication command)	517
ENDMONHA (End iCluster Replication Monitoring command).....	520
DMWRKALR (iCluster Alert Destinations command)	521
DMADDALR (Add iCluster Alert Destination command)	521
DMCHGALR (Change iCluster Alert Destination command)	523
DMRMVALR (Remove iCluster Alert Destination command).....	524
RTVHACFGS (Retrieve HA Configuration Source command).....	525
DMMRKSYNC (Mark Journal Positions and Synchronize Objects command).....	526
DMCHKALRD (Check alert destinations command)	530
Switchable device commands	530
DMADDSWDEV (Add Switchable Device Entry to Group command)	531

DMCHGSWDEV (Change Switchable Device Entry to Group command).....	532
DMRMVSWDEV (Remove Switchable Device Entry to Group command)	533
Other commands	533
SETHAREG (Restore iCluster Communications Registry command)	534
DMSETSYNC (Set Sync Point command).....	534
DMSYNCCMD (Run Command at Sync Point command).....	539
DMCHGUPRF (Change user profile command).....	542
Journal management commands.....	544
DMWRKJRN (Work with Journals command).....	545
CHGHAJRN (Change Journal Receiver command).....	547
DLTHAJRCV (Delete Journal Receiver command)	548
ENDHADJRCV (End Journal Management command).....	553
DMRMVJNMNG (Remove Journal Management Record command)	555
DMSTRJNMNG (Restart Journal Management Jobs command)	556
STRHARTJRN (Start Remote Journals command)	556
ENDHARTJRN (End Remote Journals command).....	557
Access path maintenance commands.....	558
DMWRKAPLST (Work with Access Path Lists command).....	559
DMCRTAPLST (Create Access Path List command).....	560
DMDLTAPLST (Delete Access Path List command).....	561
DMDSPAPLST (Display Access Path List command).....	561
DMADDAPL (Add Access Path to List command)	562
DMRMVAPL (Remove Access Path from List command)	563
DMADDPFAPL (Add PF Access Paths to List command)	564
DMRMVPFAPL (Remove PF Access Paths from List command)	565
DMSTRAPM (Start Access Path Maintenance command)	566
DMENDAPM (End Access Path Maintenance command).....	568
DMCHGAPM (Change Access Path Maintenance command)	569
DMDSPAPM (Display Access Path Maintenance Details command)	570
DMREGAPLST (Register Access Path List command)	572
Tools for replicating job schedule entries.....	573
DMRTVJSE (Retrieve Job Schedule Entries command)	573
DMAPYJSE (Apply Job Schedule Entries command)	574
DMRMVJSE (Remove Job Schedule Entries command).....	576
Tools for testing files on the backup node	577
DMCHGTRGBK (Change Trigger State On Backup command)	577
DMCHGCSTBK (Change Referential Constraint State on Backup command).....	578
Rocket® iCluster resources.....	580
iCluster port requirements	582

Rocket® iCluster overview

Rocket iCluster is a logical replication solution that operates on IBM Power or PureFlex systems that run IBM i. iCluster uses journaling to provide high availability and disaster recovery for business applications and enables almost continuous access through real-time monitoring, notifying, and self-correcting replication.

Plan before installation

Logical replication products like iCluster require appropriate planning, configuration, and deployment. Before you install iCluster, ensure that you conduct a careful analysis to determine what applications and objects should be replicated to the backup system or systems in your cluster. Your high availability and disaster recovery environment must meet the needs of the business while optimizing performance.

Tune after installation

After iCluster is installed and replication is active, perform initial and ongoing administration to ensure that your replication solution meets your business needs. Administration includes the following tasks:

- Monitoring for delays in replication (latency)
- Monitoring for objects that are suspended from replication, due to errors on the primary or backup system
- Performing sync checks to verify that the primary and backup nodes are synchronized

Administration is important during the initial business cycles to ensure that your configuration is tuned to optimize performance and includes or excludes the required objects. After iCluster is tuned, your day-to-day administration is minimized and can be mostly automated through alarms and alerts, continuous sync checking, and email alert notifications.

Plan and perform ongoing administration

As with all high availability and disaster recovery solutions, you must verify the integrity of the replicated backup environment before a disaster occurs. Perform regular switchover verification testing to ensure that all critical data is available. Switchover, also known as role switch, is the action of switching the roles of the primary and backup systems in your cluster so that the backup system takes over the role of the primary system. Tape backups recover only the data that is saved from the last backup and are not a viable replacement for replication solutions.

To test the integrity of the backup system in a failover, use the role switch simulator to test role switches without impacting applications on the primary node. For more information, see [Simulating role switches](#). You can engage Rocket consulting to schedule and run switchover tests.

Related information

[Simulating role switches](#)

Starting iCluster

After you complete installing and configuring iCluster, you can start iCluster to work in the active replication environment.

1. Log on to a node as a user with iCluster *ADMIN authority.
2. Go to the iCluster main menu using any of the following options:
Option 1
 - a. To change the current library to the iCluster product library, enter the following command:

CHGCURLIB ICLUSTER

- b. To start iCluster, enter the following command:

Go DMCLUSTER

Option 2

Copy ICMENU *CMD and ICMENU *PGM from the iCluster library to any library (for example, QGPL), and then run ICMENU to go to the iCluster main menu.

 **Note:**

The copy operation is only a one-time operation, and in the future, you do not need to perform this operation before running ICMENU.

3. To monitor the replication environment from the Full Cluster Status Monitor, enter the following command:
WRKCSMON
4. To view a list of supported commands, navigate to the iCluster main menu and press F22 (Shift + F10).

Related information

[WRKCSMON \(Work with Full Cluster Monitor command\)](#)

Accessing the job logs

Use the primary and backup jobs to perform communications and other batch functions. Produce job logs that contain messages about the operation of each function.

From the primary or backup node, enter one of the following commands on the command line:

- To access the default job logs:

```
WRKSPLF SELECT(DMCLUSTER)
```

- To access the job logs for a specified user:

```
WRKSPLF SELECT(<user name>)
```

Starting iCluster after a system IPL

After an initial program load (IPL), use the DMSTRCST (Start iCluster Processes) command to start the iCluster processes. You can run the command from a job scheduler or an IPL startup program.

Before you begin: If the current node that is being restarted after a system IPL does not own the iCluster metadata, a communications link is opened to the node that owns the iCluster metadata so that the node can be started.

To open the communications link, the other node must meet the following prerequisites:

- TCP/IP is running.
- The **XDMCLUSTER** subsystem is active.
- The **DMCHATL** job is running in the **XDMCLUSTER** subsystem.

Use the DMSTRCST (Start iCluster Processes) command to start the current node and the groups that are configured to autostart.

The **DMSTRCST** command performs the following:

- If specified, waits for the TCP/IP to be active.
 - Starts the **XDMCLUSTER** subsystem on the current node, if it is not started yet.
 - Starts the node that owns the iCluster metadata.
 - Starts the current node, if it is not started yet.
 - Waits for the specified time for the node to be active. The command ends if the node does not change to *ACTIVE status in the specified time.
 - Starts the other nodes of the cluster and waits until they are active.
 - Starts the groups that are configured to autostart.
1. To start iCluster processes after a system IPL, run the following command from a job scheduler or an IPL startup program:
DMSTRCST
 2. In the **STRGRP** field, enter one of the following values to define the group autostart:
 - *YES—Starts the groups that are configured to automatically start if their primary and backup nodes are active. By default, this value is used. Refresh-only groups are not automatically restarted.
 - *NO—Does not start groups.
 3. In the **STRAPY** field, enter one of the following values for the apply process:
 - *NOCHG—Preserves the last operational status of the apply jobs on the backup node. By default, this value is used.
 - *YES—Starts the apply jobs on the backup node for the groups that are being started.
 - *NO—Does not start the apply jobs on the backup node for the groups that are being started.
 4. **Optional:** In the remaining fields, define values to configure how to start the iCluster processes:
 - MSGQUEUE—Specifies the message queue that receives status and alert messages from the **DMSTRCST** command process.
 - MAXWAITTCP—Specifies the maximum time that the **DMSTRCST** command process waits for the TCP/IP communications on the current node to become active.
 - ATTEMPTS—Specifies the number of attempts to start a node.
 - MAXWAITNOD—Specifies the maximum time that the **DMSTRCST** command process waits for a node to become active after the node is started.
 - GRPSTRDLY—Specifies the time that the **DMSTRCST** command process waits to start the groups after starting the nodes. Only groups that are configured to auto-start are started. This wait time ensures that the required iCluster processes are fully active.
 - MAXWAITGRP—Specifies the maximum time that the **DMSTRCST** command process waits for a group to become active after the group is started.

Ending iCluster before a system IPL

Before an initial program load (IPL), use the DMENDCST (End iCluster Processes) command to end the iCluster processes. You can run the command from a job scheduler or a CL shutdown program.

Use the DMENDCST (End iCluster Processes) command to end the active groups that are using the current node, and then end the node and the **XDMCLUSTER** subsystem.

The **DMENDCST** command performs the following steps:

- If the groups that are using the current node are in *ACTIVE status, ends the groups.

- Waits until the groups end their operations.
 - If the current node is in *ACTIVE status, ends the current node.
 - Waits the specified time for the node to be inactive. The command ends if the node does not change to *INACTIVE status in the specified time.
 - If the XDMCLUSTER subsystem is running on the current node, ends the XDMCLUSTER subsystem.
1. To end iCluster processes before a system IPL, run the following command:
DMENDCST
 2. In the **OPTION** field, enter one of the following values to define how to stop the replication process:
 - *CNTRLD—Ends replication for the groups in a controlled manner. By default, this value is used. A controlled stop completes tasks to ensure that replication ends gracefully. However, the completion of these tasks may take some time. Controlled stops are recommended when possible.
 - *IMMED—Ends replication for the group immediately. Operations might not stop on a transaction boundary.
 3. **Optional:** In the remaining fields, define values to configure how to end the iCluster processes:
 - DELAY—Specifies the maximum amount of time, in seconds, for replication to stop in a controlled manner without intervention. If replication cannot be completed within the timeout period, replication is immediately stopped after the timeout period expires. This parameter applies only when the **OPTION** parameter is set to *CNTRLD.
 - MSGQUEUE—Specifies the message queue that receives status and alert messages from the **DMENDCST** command process.
 - MAXWAITGRP—Specifies the maximum time that the **DMENDCST** command process waits for a group to end operations.
 - ENDNODES—Specifies whether to end nodes as part of the command processing.
 - MAXWAITNOD—Specifies the maximum time, in minutes, that the **DMSTRCST** command process waits for a node to become *INACTIVE.
 - ENDSBS—Specifies whether to end the **XDMCLUSTER** subsystem as part of the command processing.
 - WAITSBSEND—Specifies whether to wait for the **XDMCLUSTER** subsystem to end before returning control to its caller. This parameter applies only when the value of the **ENDSBS** parameter is *YES.

Related information

[DMENDCST \(End iCluster Processes command\)](#)

iCluster user registration

Add one or more iCluster users on each node. Grant administrative (*ADMIN) privileges to at least one iCluster user.

Add users to each node to work with iCluster without having to sign on to the system as QSECOFR or a user with QSECOFR authority.

If the number of iCluster users is large, or if the users change frequently, it is recommended to include the iCluster users as members of an IBM i group profile. Instead of registering the user, register the group profile with iCluster. Users inherit the iCluster authority of the group profile.

To ensure operational security, restrict the ability to run commands that are based on user access levels. Define the appropriate user security level (Administrator, Operator, or User) to define the cluster operations that the user can perform.

The specified user or group profile is registered only on the node where you run the DMADDUSR (Add User) command.

Related information

[Registering iCluster users](#)

[DMADDUSR \(Add User command\)](#)

Registering iCluster users

A registered iCluster user must have an IBM i user profile that is defined with *IOSYSCFG authority. Add one or more iCluster users on each node.

Registered users can work with iCluster without having to log on to the system as the QSECOFR service tools user profile or a user with QSECOFR authority.

To define which cluster operations can be performed, register an IBM i user profile or group profile, and identify the authority level (administrator, operator, user) for each profile.

If the number of iCluster users is large or if the users change frequently, add the iCluster users as members of an IBM i group profile. Instead of registering each user, register the group profile. Users inherit the iCluster authority of the group profile.

To ensure operational security, restrict the ability to run commands that are based on user access levels. Grant administrative (*ADMIN) privileges to at least one iCluster user.

The specified user or group profile is registered only on the node where you run the DMADDUSR (Add User) command.

1. To register a user, enter the following command:
DMADDUSR
2. In the **User** field, enter an IBM i user profile or group profile that is defined on the node.
3. In the **Authority** field, enter one of the following security levels for the profile:
 - *USER: Grants user privileges. By default, this value is used.
 - *OPERATOR: Grants operator privileges.
 - *ADMIN: Grants administrative privileges.
4. **Optional:** In the **Description** field, enter a short description that identifies the full name of the user or group profile.

-
5. **Optional:** In the **iCA password** field, set the password for the user.
* (asterisk) indicates that no password has to be specified with the user name.

A user who does not have a password can still use iCluster commands in the native IBM i environment.

6. Repeat these steps on each node in the cluster for each user profile and group profile.

Related information

[DMADDUSR \(Add User command\)](#)

Changing registered users

Change the user authority, description, and passwords for registered users.

For group profiles, users inherit the authority and iCluster rights of the group profile.

1. Enter the following command:
DMCHGUSR
2. In the **User** field, specify an IBM i user profile or group profile that is defined on the node.
3. In the **Authority** field, specify the security level:
 - *SAME—Keeps the current setting for this parameter. By default, this value is used.
 - *USER—Grants user privileges to the user profile.
 - *OPERATOR—Grants operator privileges to the user profile.
 - *ADMIN—Grants administrative privileges to the user profile.
4. To add a short description that identifies the full name of the user, type up to 50 characters in the **Description** field.
5. To set a password, specify a value in the **iCA password** field:
 - <password>—Specifies the password for the user. The password is case-sensitive, uses a maximum of 10 characters, and must contain only alphanumeric characters.
 - NONE—Specifies no password. A user with no password can still use iCluster commands in the native IBM i environment.
 - *—Keeps the current password for this user. By default, this value is used.

Related information

[DMCHGUSR \(Change User command\)](#)

Removing registered users

Remove registered users from iCluster.

1. Enter the following command:
DMRMVUSR
2. In the **User** field, type the user profile name.

Related information

[DMRMVUSR \(Remove User command\)](#)

Using the Work with Users screen

Use the **Work with Users** screen to add, remove, and change attributes for users on the local node.

1. From the node where the users are registered, enter the following command:
`DMWRKUSR`
2. Repeat step 1 on each node to add, remove, and change attributes for users on that node.

Related information

[DMWRKUSR \(Work with Users command\)](#)

Server authentication for secure DDM connections

For secure DDM connections that require password validation, a server authentication entry is required on the primary and backup nodes for some users.

The following user profiles require a server authentication entry on the primary and backup nodes:

- The DMCLUSTER user profile
- The user profiles for registered iCluster users that use the Status Monitors

DDM access to iCluster metadata is required for many replication activities, including the following operations:

- Remote journal replication
- DMMONHA—replication monitoring
- Status Monitors
- File content sync check
- File activation of files that are in an out-of-sync (ACTOOS) state
- WebSphere MQ replication

For non-secured DDM connections, password validation and server authentication entries are not required for the DMCLUSTER user profile and other user profiles for registered iCluster users that use the Status Monitors.

Related information

[Verifying DDM authentication](#)

[Enabling DDM access](#)

Verifying DDM authentication

DDM might be configured as secure and require password validation. Verify how DDM is used in your environment.

When the PWDRQD parameter for the IBM i Change DDM TCP/IP Attributes (CHGDDMTCPA) command requires a user ID and password, then secure DDM server authentication is in use.

1. Log in as a user with *IOSYSCFG special authority.
2. Issue the IBM i command:
`CHGDDMTCPA`
3. Review the values that are displayed for the **PWDRQD** parameter. To determine whether secure DDM is in use, verify whether a password and user ID are required. Values include:

Values	DDM server authentication
PWDRQD parameter is *USRID or *VLDONLY	No secure DDM server authentication is in use.
PWDRQD parameter is a value other than *USRID or *VLDONLY	Secure DDM server authentication is in use.

Related information

[Enabling DDM access](#)

For secure DDM connections that require password validation, use the QDDMDRDASERVER or QDDMSERVER server authentication entry, which is required on the primary and backup nodes for the DMCLUSTER user profile and the user profiles for registered iCluster users that use Status Monitor.

Users can choose either Procedure A or Procedure B for the configuration. However, if both procedures are configured, the QDDMSERVER server authentication entry takes precedence over the QDDMDRDASERVER server authentication entry.

DDM access to iCluster metadata is required for many replication activities, including the following operations:

- Remote journal replication.
- DMMONHA—replication monitoring.
- Status Monitors.
- File content sync check.
- File activation of files that are in an out-of-sync (ACTOOS) state.
- WebSphere MQ replication.

Note: For non-secured DDM connections, password validation and server authentication entries are not required for the DMCLUSTER user profile and other user profiles.

1. To use Procedure A and add a server authentication entry with the QDDMDRDASERVER server name, complete these steps:

Note: QDDMDRDASERVER can handle both RDB server authentication and non-RDB server authentication for a user profile, so that users do not need to add each RDB server authentication entry for the DMCLUSTER user profile and specific user profiles when each new RDB is used for the remote journal or a secure DDM connection.

- a. Log on as a user with *IOSYSCFG special authority.
- b. Issue the following IBM i command: CHGDDMTCPA
- c. To determine whether a secure DDM server authentication is in use, review the value for the **PWDRQD** parameter. The parameter value determines if you need to complete the procedure steps.

Value of PWDRQD parameter	DDM server authentication
*USRID or	Authentication is not in use.

Value of PWDRQD parameter	DDM server authentication
*VLDONLY	<p style="color: red;">⚠️ Important: Do not complete the following steps.</p>
A value other than *USRID or *VLDONLY	<p>Secure DDM server authentication is in use. The DMCLUSTER user profile and user profiles for registered iCluster users that use the Status Monitor must have a valid server authentication entry on each node.</p> <p>Complete the following steps.</p>

- d. To create the QDDMDRDASERVER authentication entry for the DMCLUSTER user profile, enter the following command:

```
ADDSVRAUTE USRPRF(DMCLUSTER) SERVER(QDDMDRDASERVER) USRID(<name>) PASSWORD(<password>)
```

where:

<name> is the user ID that exists on all nodes of the cluster.

<password> is a valid password for the user ID.

Enter the SERVER name as **QDDMDRDASERVER** in uppercase. If the password changes, use the following IBM i command to update the QDDMDRDASERVER authentication entry:

CHGSVRAUTE (Change Server Authentication Entry)

- e. To create the QDDMDRDASERVER authentication entry for user profiles for registered iCluster users that use the Status Monitors, enter the following IBM i command:

```
ADDSVRAUTE USRPRF(<userprf>) SERVER(QDDMDRDASERVER) USRID(*USRPRF) PASSWORD(<password>)
```

where:

<userprf> is the user profile for registered iCluster users who use the Status Monitors.

<password> is a valid password for the user profile.

Enter the SERVER name as **QDDMDRDASERVER** in uppercase. If the password changes, use the following IBM i command to update the QDDMDRDASERVER authentication entry:

CHGSVRAUTE (Change Server Authentication Entry)

2. To use Procedure B and add a server authentication entry with the QDDMSERVER server name, complete these steps:

- Note:** QDDMSERVER is a special server name that can handle non-RDB server authentication for a user profile, but it cannot handle RDB server authentication. Users must add each RDB server authentication entry manually for a DMCLUSTER user profile or specific user profiles when each new RDB entry is used for a remote journal or secure DDM connection.

- a. Log on as a user with *IOSYSCFG special authority.
- b. Issue the following IBM i command:
CHGDDMTCPA
- c. To determine whether secure DDM server authentication is in use, review the value that is displayed for the **PWDRQD** parameter. The parameter value determines if you need to complete the procedure steps.

Value of PWDRQD parameter	DDM server authentication
*USRID or *VLDONLY	Authentication is not in use. Do not complete the following steps.
A value other than *USRID or *VLDONLY	Secure DDM server authentication is in use. The DMCLUSTER user profile and user profiles for registered iCluster users that use the Status Monitor must have a valid server authentication entry on each node. Complete the following steps.

- d. To create the QDDMSERVER authentication entry for the DMCLUSTER user profile, enter the following IBM i command:

```
ADDSVRAUTE USRPRF(DMCLUSTER) SERVER(QDDMSERVER) USRID(<name>)
PASSWORD(<password>)
```

where:

<name> is the user ID that exists on all nodes of the cluster.

<password> is a valid password for the user ID.

Enter the SERVER name as **QDDMSERVER** in uppercase. If the password changes, use the following IBM i command to update the QDDMSERVER authentication entry.

CHGSVRAUTE (Change Server Authentication Entry)

- e. To create the QDDMSERVER authentication entry for user profiles for registered iCluster users that use the Status Monitors, enter the following IBM i command:

```
ADDSVRAUTE USRPRF(<userprf>) SERVER(QDDMSERVER) USRID(*USRPRF) PASSWORD(<password>)
```

where:

<userprf> is the user profile for registered iCluster users who use the Status Monitors.

<password> is a valid password for the user profile.

Enter the SERVER name as **QDDMSERVER** in uppercase. If the password changes, use the following IBM i command to update the QDDMSERVER authentication entry:

CHGSVRAUTE (Change Server Authentication Entry)

3. **Optional:** Remote journaling and IBM WebSphere MQ replication require authentication entries. If you use remote journaling or WebSphere MQ replication, run these commands for the automatically created remote database directory entries (RDBDIRE) on each node and for the remote database directory entries (RDBDIRE) that are used for remote journaling.

- a. On the primary node of the group, enter the following IBM i command:

```
ADDSVRAUTE USRPRF(DMCLUSTER) SERVER(<rdbentry_bn>) USRID(<name>)  
PASSWORD(<password>)
```

where:

<rdbentry_bn> is the name of the remote database directory entry for the backup node.

<name> is the user ID that exists on all nodes of the cluster.

<password> is a valid password for the user ID.

- b. On the backup node of the group, enter the following IBM i command:

```
ADDSVRAUTE USRPRF(DMCLUSTER) SERVER(<rdbentry_pn>) USRID(<name>)  
PASSWORD(<password>)
```

where:

<rdbentry_pn> is the name of the remote database directory entry for the primary node.

<name> is the user ID that exists on all nodes of the cluster.

<password> is a valid password for the user ID.

Node operations

To create an environment for high availability or disaster recovery, add two or more nodes to a cluster.

A node is considered to be a single computer and can be a member of only one cluster. Each node in a cluster must be able to communicate with every other node in the cluster. You can add up to 128 nodes to a cluster.

The first node that you add to a cluster becomes the node that owns the iCluster metadata. The iCluster metadata maintains the configuration information for the cluster.

The following node guidelines apply to node operations:

- Node information is shared among the active nodes in a cluster.
- Never stop the XDMCLUSTER subsystem on a node that is active in a cluster.

Adding a node to a cluster

Change your cluster to include another node. For example, add a node in a different location to enable disaster recovery.

Before you begin: Verify that the first node that is added to a cluster can communicate with every other node in the cluster.

The IP address can be entered in domain name format or dotted quad format. The domain name format is resolved to dotted quad format when the node is added to the cluster. The dotted quad format address is used for communications.

1. Ensure that the XDMCLUSTER subsystem is running on all of the nodes that will be defined in the cluster.
2. Ensure that the DMCHATL listener job is running in the subsystem. To start the XDMCLUSTER subsystem, use the STRHATCP (Start TCP/IP Listener) command.
3. To start the internet daemon (INETD), enter the following IBM i command:
`STRTCPSVR SERVER(*INETD)`

 **Tip:** Add this command to the autostart job that runs when the machine starts.

4. To ensure that all cluster information is recorded if node operations fail, enter the following command to set the event log message generation level to *ALL:
`DMSETSVAL EVNTLOG(*SAME *SAME *SAME *ALL)`
5. To add a node, enter the following command:
`DMADDNODE`
6. In the **Node** field, type a unique name, up to eight characters, for the node.
7. In the **IP Address** field, type the primary IP address of the node in dotted quad notation or in domain name form.
8. In the **Alternate IP Address** field, type the alternate IP address of the node if the failover mechanism is Cluster Resource Services.
9. In the **Port** field, type a TCP/IP port number for the dmcluster service entry on the node.
By default, the value is 4545. Valid values are from 1-65535.
10. In the **Staging store size (in MB)** field, type the size of the staging store.
Valid values are from 512 to 1048576.
11. Specify the library where the staging store resides in the **Staging store library** file.
This library is created by the product. Choose one of the following values:

- <name>—Specifies the staging store library name.
 - DMSTORE—By default, this value is used.
12. For the remaining values, accept the default values. If needed, enter values that are applicable for your environment.

Related information

[DMADDNODE \(Add Node command\)](#)
[STRHATCP \(Start TCP/IP Listener command\)](#)

Viewing the node settings

Use the DMDSPNODE (Display Node) command to view the current settings for a node.

1. Enter the following command:
DMDSPNODE
2. Type the name of the node in the **Node** field.
3. To view the command prompt, press F4.

Related information

[DMDSPNODE \(Display Node command\)](#)

Removing a node from a cluster

Use the DMRMVNODE (Remove Node) command to remove a node from a cluster.

You might need to remove a node for different reasons. For example, remove a node when you replace the hardware. At least one node in the cluster must be active before nodes can be removed from the cluster.

1. Ensure that at least one node in the cluster is active.
2. Enter the following command:
DMRMVNOD
3. In the **Node** field, type the name of the node.

Related information

[DMRMVNOD \(Remove Node command\)](#)

Changing node attributes

Use the DMCHGNODE (Change Node) command to change the attributes of a node.

Except for the staging store library changes, new values that apply to replication take effect when new replication jobs are started. Existing replication jobs continue to use the previous node attributes.

1. On an inactive node of the group, enter the DMCHGNODE command.
2. In the **Node** field, type the name of the node to change.
3. On the **iCluster Change Node** screen, change one or more values.
The following table lists several of the fields that you might want to change:

Option	Description
Job description	It is recommended that you retain the default CSJOBDB job description in the iCluster product library. However, you can change the default job description to alter performance parameters.
User profile status	By default, user profiles are replicated to the backup node with *DISABLED status. The *DISABLED status prevents users from signing on. If your business needs for the backup node require users to sign on for queries, reporting, and so on, you can change how user profile statuses are replicated.
Hold config object source	Specify whether to create configuration objects as they are received on the node that is added to the cluster, or to hold the commands for creating the configuration objects in source physical files for later use.
Enable switchable resources	When the failover mechanism is Cluster Resource Services, you can configure a node to use switchable resources. IBM PowerHA requires switchable resources. HA Switchable Resources (Option 41 of IBM i) is required.

4. Press F1 to see help on any field.

Related information

[DMCHGNODE \(Change Node command\)](#)

Renaming a node

Use the DMRNMNODE (Rename Node) command to rename a node.

Before you begin: Verify that all nodes in the cluster are active. The node to be renamed cannot be the node where the command is run. Groups that have the node that is being renamed in their recovery domain must be inactive and have another node in their recovery domain. The groups must have a current backup node. The groups cannot be local loopback groups.

1. Ensure that all nodes in the cluster are active.
2. Ensure that groups that have the node that is being renamed in their recovery domain are inactive, and that the groups have another node in their recovery domain.
3. On a node that is not being renamed, enter the following command:
DMRNMNODE
4. In the **Node** field, type the name of the node to change.
5. In the **New node name** field, type the new name of the node and press Enter.
6. Restart the groups that have the renamed node in their recovery domain, as if the groups were new groups. Use the DMMRKPOS command, or do a refresh, depending on how much data is being replicated.
The renamed node is removed from the cluster and added back to the cluster with the same parameters.
Historical replication information is not retained.

Related information

[DMRMVNOD \(Remove Node command\)](#)

Assigning a failover message queue to a node

Node failures produce specific messages. When third-party software is used to monitor message queues, assign a failover message queue to a node.

1. Enter the following command:

DMCHGNODE

2. Type the name of the node in the **Node** field.
3. Press **PgDn**.
4. Specify the name of the message queue to receive messages after the set number of communication failures occurs in the **Message queue name** field.
Choose one of the following values:
 - <name>—Specifies the name of the message queue.
 - *SAME—Keeps the current setting for this parameter.
 - *NONE—Specifies that no message queue are used.
5. Type the name of the library where the message queue resides in the **Library** field.
6. Specify the maximum number of messages to send after a failure is detected in the **Number of message actions** field.
By default, the value is *SAME.
7. Specify the number of minutes to wait before a message is sent in the **Wait time after action (min)** field.
By default, the value is *SAME.

Related information

[DMCHGNODE \(Change Node command\)](#)

Specifying a failover user exit on a node

Automate diagnostic or recovery tasks to run when a node fails.

When a node fails, diagnostic or recovery tasks such as restarting communications might be required. Automate the tasks in a user exit that runs automatically when the node fails.

1. Enter the following command:

DMCHGNODE

2. Type the name of the node in the **Node** field.
3. Press **PgDn**.
4. Assign the name of the user exit program to run after the set number of communication failures has occurred in the **Message action user exit** field.
Choose one of the following values:
 - <name>—Specifies the user exit program name.
 - *NONE—Specifies that no user exit program is called. By default, this value is used.
5. In the **Library** field, type the name of the library where the user exit program resides.

Related information

[DMCHGNODE \(Change Node command\)](#)

Changing the status of a node

Manually change the status of a node so that it can be restarted and made active.

For example, if contact is lost between one or more nodes in the cluster, and the failure of the lost nodes cannot be confirmed, the node might have *PARTITION status. This status might be caused by a power failure, hardware failure, or a communications failure. After the problem is resolved, you can change the status of the node to *FAILED so that the node can be restarted and made *ACTIVE.

1. Enter the following command:

DMCHGNODE

2. Type the name of the node in the **Node** field.
3. Press **PgDn**.
4. Specify the status of the node in the **Change node status** field.

Choose one of the following values:

- *FAILED—Identifies that the node failed. Use this value when a node failed but its status in the cluster is *PARTITION.
- *SAME—Keeps the current setting for this parameter.

After the status of the node is changed to *FAILED, it can either be removed from the cluster or restarted if it is capable of rejoining the cluster.

Related information

[DMCHGNODE \(Change Node command\)](#)

Viewing nodes in the cluster

Use the **Work with Cluster Nodes** screen to view a list of nodes in the current cluster.

You can choose one of the following methods to view nodes in the cluster:

- All defined nodes.
- All nodes that are in the recovery domain of a specific group.

1. Enter the following command:
DMWRKNODE
2. Use the following parameter or parameters to view nodes:
 - To include all nodes, use:

BY(*NONE)

- To include all nodes by group, use:

BY(*GROUP) GROUP(<group name>)

For example, to view all nodes by group:

```
DMWRKNODE BY(*GROUP) GROUP(<group name>)
```

Related information

[DMWRKNODE \(Work with Nodes command\)](#)

Checking for communications failure on a node

When the failover mechanism is SwitchOver System, verify communications failures on a node.

1. Enter the following command:

```
DMCHGNODE
```

2. Type the name of the node in the **Node** field.
3. Press **PgDn**.
4. To test the primary IP address for communication failures between the primary and backup nodes, specify one of the following values in the **Check primary link** field:
 - *SAME—Keeps the current setting for this parameter.
 - *YES—Tests the primary IP address.
 - *NO—Does not test the primary IP address.
5. To change how often to poll the primary and alternate links for communication failures, specify a new value in the **Link check frequency (sec)** field.
This value can range from 1 - 300. The default value is *SAME.
6. To change how long you want to wait for a response when polling links for failures, specify a new value in the **Link check reply timeout (sec)** field.
This value can range from 1 - 300. The default value is *SAME.
7. To change the maximum number of consecutive communication failures to allow before the primary node is considered failed, specify a new value in the **Maximum link check retries** field.
The default value is *SAME.

Related information

[DMCHGNODE \(Change Node command\)](#)

Verifying communications between nodes in a cluster

Use the HAPNGTCP (Ping Using TCP) command to contact a remote node in the cluster and to determine whether the node is available and operational.

1. Enter the following command:

```
HAPNGTCP
```
2. To specify the host name of a remote node in the cluster, use the **RMTNME(<host name>)** parameter.
3. To specify the port number on the remote node in the cluster, use the **RMTPRT(<port number>)** parameter.
4545 is the default port. Use the port number that was specified in the TCP/IP Services Table entry during the installation process.
4. To specify the length of packets, use the **PKTLEN(<value>)** parameter.
Valid values are 32-32760. The default value is 256.
5. To specify the number of packets to send, use the **NBRPKT(<value>)** parameter.
Valid values are from 1-999. The default value is 5.

6. To specify the name of the iCluster installation library on the remote node, use the **RMTLIB(<library name>)** parameter.
The default value is ICLUSTER.
7. To specify the job description that is associated with iCluster jobs running on the remote node, use the **JOBD(<job description>)** parameter.
The default value is CSJOB.

Related information

[HAPNGTCP \(Ping Using TCP command\)](#)

Changing the IP address of a node

Change the node IP address after the IP addresses in the network are changed. Update only one node's IP address at a time.

Before you begin: The following prerequisites apply to changing the IP address of a node:

- The system must be running, including TCP/IP.
 - The new IP address must already be in use on the node.
 - The node whose IP address is being changed must be inactive in the cluster.
 - All groups that use the node in their recovery domain must be inactive.
1. To ensure that the groups that use the node in their recovery domain are inactive, enter the DMENDGRP (iCluster End Group) command to stop all groups that use the IP address you are changing.
 2. To ensure that the node is inactive, end the XDMCLUSTER subsystem on that node.

Enter the following IBM i command at the command line:

`ENDSBS SBS(XDMCLUSTER) OPTION(*IMMED)`

3. To change the node IP address, enter the DMCHGNODIP (Change Node IP Address) command on an active node in the cluster.

Enter this information:

- a. In the **Node** field, type the node name of the node on which the IP address is being changed.
 - b. In the **New IP address** field, type the new IP address in dotted quad notation or in domain name form.
The domain name format resolves to the dotted quad format address that is used for communications.
4. To restart the XDMCLUSTER subsystem on the node for which the IP address was changed, enter the following IBM i command:
`STRSBS SBSD(XDMCLUSTER)`
 5. To restart the node, enter the following command on an active node:
`DMSTRNODE NODE(<node name>)`

where <node name> is the node whose IP address was changed.

6. To start replication for all groups that use the node, enter the DMSTRGRPD MSTRGRP (iCluster Start Group) command.

Related information

[DMSTRNODE \(Start Node command\)](#)

Group operations

A group is a replication relationship between nodes in a cluster. A group is also a logical construct that bundles libraries and objects together for replication management.

For example, libraries and objects that are used by a business application to provide general ledger functions should be replicated together and be in the same group.

Backups and switchovers are typically done at the group level; multiple groups enable better granularity to manage these processes. A group can have one or two nodes, but a primary node is always required. For replication groups, a backup node is also required.

Setting up groups

After creating a group, specify the objects that the group replicates from the primary to the backup node, or specify multiple objects for replication in the same group. You can also specify the same object in multiple groups from the same primary node.

Nodes in a cluster belong to one or more groups. Nodes in a cluster can belong to multiple groups and can replicate the same or different objects in these groups.

The replication process copies data and applications from the primary node to the backup node. Replication groups operate independently of each other. Start, stop, and configure one group without affecting the other groups in the cluster.

Local loopback replication occurs when objects are replicated between environments on the same physical server or logical partition. Because replication is within the same physical server, an object cannot replicate on top of itself and recursive updates are not supported. You must specify a different library for local loopback replication.

Local loopback replication is available only for native library-based objects, and it is not supported for BSF objects, WebSphere MQ groups, or switchable device groups.

SwitchOver System

All group types except switchable device groups (*SWDEV) and application groups (*APPL) are supported when the failover mechanism is SwitchOver System.

In a cluster that uses SwitchOver System as the failover mechanism, each active node in the cluster regularly polls its connections with the other nodes in the cluster. Configure node and group properties to control the action that the backup node takes when it loses its connections to the other nodes in the cluster. Define how many attempts are made to reestablish a connection before giving up and declaring the node as *FAILED.

Adding a group

Add a group, which is a replication relationship between nodes in a cluster.

The following types of groups can be defined:

- Replication group—Supports the replication of objects and data for high availability.
- MQSeries group—Provides a high availability solution for WebSphere MQ.
- Refresh-only group—Performs a refresh of the objects that are selected to the group and then shuts down. These groups are not eligible for role switch.

- Stand-by group—Cannot be used for replication until converted to a replication group or a refresh-only group.
- *HADR group—Supports the replication of objects and data to a high availability backup node and a disaster recovery backup node.

Additionally, when the failover mechanism is Cluster Resource Services, the switchable device group type is supported.

Auto-configuration creates only two groups. Follow these steps to create more groups.

1. Enter the following command:
`DMADDGRP`
2. Type a unique name for the group in the **Group** field.
Use up to 10 characters.
3. In the **Group type** field, type one of the following values:

Value	Description
*REPL	Replication groups enable the replication of objects and data for high availability. Add and configure a replication group without affecting other groups in the cluster.
*RFSH	Refresh-only groups perform a refresh of the objects that are selected to the group. Refresh of the objects are from the primary node to the backup node. Refresh-only groups are eligible for role switching only if the Allow roleswitch for *RFSH groups system value is set to *YES.
*SWDEV	Switchable device groups support switchable disks when the failover mechanism is Cluster Resource Services.
*MQSERIES	MQSeries groups provide a high availability solution for WebSphere MQ
*STDBY	Standby groups cannot be used for replication until converted to a replication group or a refresh-only group.
*HADR	*HADR groups support the replication of objects and data to a high availability backup node and a disaster recovery backup node.

4. In the **Primary node** field, specify the name of the node.
The node must be added to the cluster with the `DMADDNODE` command. You can add a backup node later.
5. Accept the default values, change values, or enter values as applicable for your environment.
6. Press Enter to save the group.
7. Press F1 to see the help on any field.

Related information

[DMADDGRP \(Add Group command\)](#)

[DMADDNODE \(Add Node command\)](#)

Changing group attributes

After a group is created, use the DMCHGGRP (Change Group) command to change the attributes of the group.

Before you begin: Verify that the group is inactive before the changes are applied.

1. On an inactive node of the group, enter the following command:

DMCHGGRP

2. In the **Group** field, type the name of the group to change.

3. On the **iCluster Change Group** screen, change one or more values.

These are several of the fields that you can change:

- **Staging store library**—The name of the staging store library for the group. To improve latency, create a separate staging store for the replication group.
- **Check journaling at roleswitch**—If the role switch is taking too long with CHKJRN(*YES), change the value to *NO.
- **Journal location**—The location of the database journal where scraping occurs. The default is *LOCAL. To use remote journaling, specify *REMOTE.
- **Delay between objects (msec)**—Decrease this value if the default value of 200 milliseconds causes the sync checks to take too long.
- **Optimize apply for updates**—If you discover apply latency after the initial configuration, change this value to *YES to enable optimization for database apply updates. Optimization can increase the apply performance for large files that are significantly larger than the shared memory pool.
- **Max. record level errors**—Work with Technical Support and change this value to get a replication group back in sync.

4. Press F1 to see the help on any field.

Related information

[Understanding object specifiers](#)

[DMCHGGRP \(Change Group command\)](#)

Removing a group

Use the DMRMVGRP (Remove Group) command to remove a group. For example, remove the group when an application is no longer required for high available or disaster recovery.

If a group is active when the command is issued, verify that the group is ended before it is removed.

1. Enter the following command:

DMRMVGRP

2. Type the name of the replication group to remove in the **Group** field.

3. Specify whether to delete an active group in the **Remove active group** field.

Related information

[DMRMVGRP \(Remove Group command\)](#)

Viewing groups in a cluster

Use the DMWRKGRP (Work with Groups) command to list the groups in a cluster. Use the **BY** and **NODE** parameters to filter the list.

-
1. On an active node in the cluster, enter the **DMWRKGRP** command.
 2. Press Enter to see all groups in the cluster, or specify the filter and value and then press Enter.
 3. In the **Output** field, specify whether to display the group list on the console or write the group status report to a spooled file.

Related information

[DMWRKGRP \(Work with Groups command\)](#)

Working with recovery domains

Use a recovery domain, which consists of the primary and backup nodes in a group.

A recovery domain has only two nodes: one primary node and one backup node. An *HADR group's recovery domain has three nodes: one primary node and two backup nodes.

Adding a backup node to the recovery domain of a group

If a group does not have a backup node (for example, when the previous backup node has been removed from the cluster), add a backup node to the recovery domain of the group.

All groups require a primary node and a backup node to replicate or start.

1. Enter the following command:
DMADDBACK
2. Type the name of the group in the **Group or resilient application** field.
3. Type the name of the node in the **Node** field.
4. In the **Backup IASP device name** field, specify the name of the independent auxiliary storage pool (IASP) on which the group replicates on the backup node.
 - <name>—Specifies the IASP name.
 - *SYSBAS—Uses a system ASP rather than an IASP. By default, this value is used.
5. The **Backup node type** field is only applicable to an *HADR group. Unless the group is an *HADR group, do not change the default value *ONLY in this field.

Related information

[DMADDBACK \(Add Backup Node to Recovery Domain command\)](#)

Removing a backup node from the recovery domain

When a node is being replaced or is no longer needed, use the DMRMVBAC (Remove Backup Node from Recovery Domain) command to remove it from the recovery domain of the groups where it is the backup node.

Before you begin: If the node to remove is the primary node of a group, you must first do a roleswitch (DMCHGROLE) to make it the backup node of the group.

Verify that the groups are inactive.

1. Enter the following command:
DMRMVBAC
2. Type the name of the group in the **Group or resilient application** field.
3. Type the name of the node in the **Node** field.

Related information

[DMRMVBACK \(Remove Backup Node from Recovery Domain command\)](#)

Changing the primary node to the backup node

As part of a planned switchover, change the primary node of an inactive group to the current backup node. To perform a role switch on an active group, use the **DMSTRSWO** command.

The group remains *INACTIVE until the DMSTRGRP (iCluster Start Group) or DMSTRAPP (iCluster Start Application) command restart it.

1. Ensure that groups have *INACTIVE status.
2. On an active node in the cluster, enter the following command:
DMCHGROLE
3. Specify the name of the group in the **Group or resilient application** field.
Choose one of the following values:
 - <name>—Specifies the group name.
 - *ALL—Changes the primary node of all groups in the cluster to the current first backup node.
 - *PRIMNODE—Performs a role switch for all groups with the specified primary node.
4. If *PRIMNODE was selected, type the name of the node in the **Primary node** field.

Related information

[DMCHGROLE \(Change Primary command\)](#)

[DMSTRSWO \(Switchover Group command\)](#)

Viewing current settings for a specified group

Display and work with the current settings for replication groups.

On an active node in the cluster, use one of the following actions to view the group settings.

- a. To view replication groups, including refresh groups, enter the following command:
DMDSPGRP GROUP(<group name>)
Press F4 to prompt.
- b. To view any group, enter the following command:
DMKWRKGRP
Type 5 in the **Option** field next to the group, and press Enter.
- c. To view groups with a specified note, enter the following command:
DMKWRKGRP BY(*NODE) NODE(<node name>)
Type 5 in the **Option** field beside the group, and press Enter.

Related information

[DMDSPGRP \(Display Group command\)](#)

[DMWKRKGRP \(Work with Groups command\)](#)

Starting replication

After you install and configure iCluster, start replication from the primary node to the backup node of the groups that are configured.

Following are the available options to start replication:

- [Initial synchronization](#)
- [Start replication using marked journal positions](#)
- [Starting replication groups](#)

Initial synchronization

Consider the possible scenarios that might occur before you start replication.

Refresh (Before Mirroring)

Initiating a refresh operation copies the selected objects in a group from the primary node to the backup node before replication begins. Since replication only captures changes made to objects on the primary node, refreshing the data ensures the object on the backup node is the same as the object on the primary node. Use the **REFRESH** parameter of the DMSTRGRP (iCluster Start Group) command to initiate this operation.

Replication and Mirroring

After synchronizing objects on primary and backup nodes through a refresh, replication ensures that updates applied to objects on the primary node are replicated as soon as possible to the backup node. Replication occurs automatically in any replication group that is started and has a replication status of *ACTIVE. You can start a group with the DMSTRGRP (iCluster Start Group) command.

Replication maintains an up-to-date image of objects on the backup node to ensure that a switchover or failover to the backup node can be performed with minimal or no disruption to business operations. Replication is a conceptually continuous operation that is stopped only as a result of anticipated or unplanned interruptions.

Marking positions and synchronizing

Saving a group's replicated objects (or a subset of its replicated objects) on the primary node, restoring them on the backup node, and restarting replications with the marked position can be considerably faster than using the Refresh *YES option on the DMSTRGRP (iCluster Start Group) command, particularly if the group is replicating mostly database files. Use the DMMRKSYNC (Mark Journal Positions and Synchronize Objects) command to perform this function.

Related information

[DMSTRGRP \(Start Group command\)](#)

[DMMRKSYNC \(Mark Journal Positions and Synchronize Objects command\)](#)

Start replication using marked journal positions

Refresh the selected objects using save-restore, run the DMMRKPOS command, and then start the group using marked journal positions instead of refreshing the selected objects through the group replication process.

This method can be considerably faster than using the **Refresh** *YES option on the DMSTRGRP (iCluster Start Group) command, particularly if the group is replicating mostly database files.

1. Save the objects using one of the following options:
 - **Entire libraries are replicated**—To save the entire library or libraries on the primary node, save the libraries to tape or to an optical device. Run the DMMRKPOS command for the group.

For example, when the only object specifier is *ALL objects of *ALL types and no objects are excluded.

 - **Some objects in libraries are excluded, or only specific objects from a library are selected**—To save only the objects that are in replication scope to a save file, use the SAVOBJ parameter on the DMMRKPOS command.
2. Restore the objects to the backup node.
3. Start the group with the **Use Marked Positions** parameter set to *YES.

Related information

[DMMRKPOS \(Mark Journal Positions command\)](#)

Starting replication groups

After initial synchronization, start the replication groups.

1. From the iCluster main menu, enter 2 for **Work with Groups**.
2. On **Work with iCluster Groups**, type 1 for the replication group. Then press Enter.
3. On **iCluster Start group (DMSTRGRP)**, type *YES for **Start replication apply jobs** and press F10 to display additional parameters.
4. Depending on how the initial refresh is performed, take one of the following actions:
 - Refresh before mirroring:

Type *YES for **Refresh selected objects** and press **Enter**.

The following actions occur:

- The replication group is started.
- The remote journal is created on the BACKUP node, if necessary.
- The staging store library is created on the BACKUP node, if the group has its own staging store.
- The libraries are created on the backup node.
- A full refresh of the selected objects on the primary node is saved and sent to the backup node.
- Monitoring and mirroring of the selected objects from the primary node to the backup node is started.
- Use marked journal positions:
 - Ensure that all objects are successfully restored from either the tape or the save files that are created by the DMMRKPOS command.
 - Type *YES for **Use marked journal positions** and press enter.

The replication group is started from the journal positions that are marked with the DMMRKPOS command. You can start monitoring the group.

Related information

[DMMRKPOS \(Mark Journal Positions command\)](#)

Resilient devices

Resilient devices are supported when Cluster Resource Services is the failover mechanism.

A resilient device is a physical resource that is accessible from more than one node in a cluster. The switchable independent auxiliary storage pools (IASP) device is the only type of device that can be defined as resilient. An IASP can go offline or online independently of the rest of the system storage.

A resilient device cluster resource group can contain a list of up to 256 switchable devices. Each device in the list identifies a switchable IASP. When a failure of the primary node occurs, the entire collection of devices is switched to the backup node. Resilient device groups can be added only by using the DMADDGRP (Add Group) command.

Related information

[DMADDGRP \(Add Group command\)](#)

Adding a switchable device entry to a group

Use the DMADDSWDEV (Add Switchable Device Entry to Group) command to add a switchable device entry to a group. The IASP device is assigned to the primary node of a group, but it is reassigned to the new primary node after a switchover or failover of the group.

Nodes that use the resilient device must be included in the cluster.

1. Enter the following command:

DMADDSWDEV

2. Press Enter.
3. Type the name of the switchable device resource group in the **Group** field.
4. Type the name of the device in the **Switchable device name** field.
5. Specify if the device is varied on or varied off when a switchover or failover occurs in the **Online at switchover** field.
Choose one of the following values:
 - *YES—Specifies that the device will be varied on when the group is switched over from one node to another or when it is failed over to another node. By default, this value is used.
 - *NO—Specifies that the device will not be varied on when the group is switched over from one node to another or when it is failed over to another node.

6. Press Enter.

Related information

[DMADDSWDEV \(Add Switchable Device Entry to Group command\)](#)

Viewing the attributes of a switchable device replication group

Use the DMDSPASPGP (Display Switchable Device Group) command to view the attributes of a switchable device replication group.

1. Enter the following command:
DMDSPASPGP
2. In the **Group** field, type the name of the switchable resource group.

-
3. To prompt, press F4.

Changing the attributes of a switchable device entry in a group

To change whether the device is varied on or varied off at switchover or failover, use the DMCHGSWDEV (Change Switchable Device Entry to Group) command.

1. Enter the following command:
DMCHGSWDEV
2. Type the name of the switchable resource group in the **Group** field.
3. Type the name of the switchable device in the **Switchable device name** field.
4. Specify if the device is varied on or varied off when a switchover or failover occurs in the **Online at switchover** field.
Choose one of the following values:
 - *YES—Specifies that the device will be varied on when the group is switched over from one node to another or when it is failed over to another node. By default, this value is used.
 - *NO—Specifies that the device will not be varied on when the group is switched over from one node to another or when it is failed over to another node.

Related information

[DMCHGSWDEV \(Change Switchable Device Entry to Group command\)](#)

Removing a switchable device entry from a group

To remove a switchable device entry from a group, use the DMRMVSWDEV (Remove Switchable Device Entry to Group) command.

1. Enter the following command:
DMRMVSWDEV
2. Type the name of the switchable resource group in the **Group** field.
3. Type the name of the switchable device in the **Switchable device name** field.

Related information

[DMRMVSWDEV \(Remove Switchable Device Entry to Group command\)](#)

Object specifiers and path object specifiers

Replication objects are the data, applications, and other objects on a node that you want to replicate. Replicate data is stored in native IBM i objects or Integrated File System (IFS) files.

Object specifiers are definitions that reference the objects to replicate within a recovery domain. Define any number of object specifiers. Through generic names and special values, each object specifier can reference more than one object on a node.

Path object specifiers identify the location of Byte Stream File (BSF) objects in the Integrated File System (IFS) or the IBM i document library services file system (QDLS). Document library objects (DLO) reside in the IBM i document library services file system.

Directory entry specifiers and objects are listed in the iCluster native object panels (DMWRKOBJ, WRKHAOBJST, DMWRKOBJST and others) with the object type *DIRE. This is not a true IBM i object type, but is simply used by iCluster to indicate directory entry specifiers and objects in the native object panels.

Understanding object specifiers

Define any number of object specifiers to reference the objects to replicate. You can define object specifiers with generic names and special values. Each object specifier can reference more than one object on a node.

An object specifier consists of the following elements:

- Object name—Generic and special values for the object name can be specified to reference multiple objects.
- Library name—The name of the library where these objects are stored. Generic library names are supported.
- Object type—The object type of the specifier to select. Specify *ALL to reference all object types that can be replicated.
- Object attribute—Object attributes specify the type of *FILE object. Specify *ALL to reference all types of *FILE objects that can be replicated.



Note: Names for objects and libraries must conform to IBM i naming rules. Names may begin with Roman letters and these special characters: \$ (dollar sign), # (number sign/hash tag), @ (at sign), and " (double quote). Names may also contain these same characters, in addition to _ (underscore) and . (dot) as long as these last two are not the first characters of the name.

When replication is active, the following rules apply to selecting native IBM i objects and BSF objects for replication:

- You can add *INCLUDE object specifiers while replication is active.
- Objects that come into replication scope because of new *INCLUDE object specifiers can be refreshed or activated without refresh.

Generic object specifiers

Generic object specifiers refer to a group of objects.

A generic object specifier can have one or more of the following properties:

- A generic name component; for example, AB*
- A generic library component; for example, AC*

- Name component of *ALL
- Type component of *ALL
- Type component of *FILE and attribute component of *ALL

! **Note:** The attribute component is ignored for object types other than *FILE.

Rules of precedence for object specifiers

Rules of precedence apply when an object matches more than one object specifier.

For example:

- Exact object specifiers (non-generic) have the highest precedence. INCLUDE and EXCLUDE specifiers have the same precedence when they are exact specifiers.
- Generic EXCLUDE specifiers have the next highest precedence.
- Generic INCLUDE specifiers have the lowest precedence.

Precedence among generic specifiers is based on the length of the non-generic part. A less generic name has higher precedence.

Working with object specifiers

Through the use of generic object specifiers, reference any number of objects through a single object specifier.

Before you use a save file or tape transfer to synchronize objects on the primary and backup nodes, select the object specifiers that reference these objects before you save the objects. Selecting the object specifiers first ensures that changes to the objects are audited between the time of the save and the time when replication is started. Object specifiers do not apply to WebSphere MQ groups.

When you define object specifiers for *FILE objects, an attribute of *ALL matches all types of files, including database files and device files. To include only database files, create two object specifiers, one with an attribute of PFDTA (to include physical database files) and the other with an attribute of LF (to include logical files). Note that the attribute PF (physical files) includes both source physical files and database files.

When you create an object specifier for a physical file, specify an object attribute of PFDTA to access certain iCluster parameters, such as the update method (by relative record number or by unique key), that cannot be accessed if PF or PFSRC (source physical file) is specified.

System directory entries are special object specifiers and are not selected to a replication group using the DMSELOBJ command or deselected using DMDSELOBJ. For system directory entries use the DMSELDIR and DMDSELDIR command instead.

Related information

[DMSELDIR \(Select Directory Entry command\)](#)

[DMDSELDIR \(Deselect Directory Entry command\)](#)

Selecting an object specifier to a replication group

Use the DMSELOBJ (Select Objects to Group) command to select an object specifier for a replication group or a refresh-only group.

When you define a new object specifier and select it for a group, the objects that are referenced by the selected object specifiers are replicated from the primary node to the backup node in the group.

1. On an active node in the cluster, enter the **DMSELOBJ** command.
2. Type the name of the group in the **Group** field.
3. In the **Object** field, specify the object name component of the specifier to select.
Choose one of the following values:
 - *ALL—Specifies all objects in a library.
 - <name>—Specifies the object name.
 - <generic*>—Specifies a generic object name, to identify multiple objects in a library.
4. In the **Library** field, type the name of the library where the objects reside.
You can use a generic name for the library component (for example, AC*).
5. In the **Object Type** field, specify the object type component of the object specifier.
Choose one of the following values:
 - *ALL—Specifies all object types.
 - <name>—Specifies the object type. Press F4 for a list of all values.
6. In the **Object extended attribute** field, specify the attributes of the chosen specifier.
Choose one of the following values:
 - *ALL—Specifies all object attributes. By default, this value is used.
 - <name>—Specifies the attribute type. Press F4 for a list of all values.
7. In the **Target library** field, specify the name of the library on the backup node that will receive the replicated objects.
Choose one of the following values:
 - <name>—Specifies the library name.
 - *PRIMARY—Specifies the same library as the primary node library where the object resides.
 - *GROUP—Uses the library that is specified in the DMADDGRP (Add Group) or DMCHGGRP (Change Group) commands. By default, this value is used.
8. In the **Description** field, add a short description to identify the specifier.
Type up to 50 characters.
9. In the **Include or exclude** field, specify whether the objects referenced by the specifier will be replicated within the group.
Choose one of the following values:
 - *INCLUDE—Replicates referenced objects within the group when cluster operations are started. By default, this value is used.
 - *EXCLUDE—Does not replicate referenced objects within the group when cluster operations are started.
10. In the **Object polling interval** field, specify the polling interval for monitoring content changes to source physical files and user spaces.
Choose one of the following values:
 - *CLUSTER—Uses the system value that is assigned in the DMSETSVAL (Set Cluster System Values) command.
 - *GROUP—Uses the group value specified in the DMADDGRP (Add Group) or DMCHGGRP (Change Group) commands. By default, this value is used.
 - *NONE—Specifies that source physical files and user spaces will not be polled.
11. In the **Activation of new objects** field, choose the method by which replication will begin for the objects that come into replication scope when an object specifier is added.
Choose one of the following values:
 - *NONE—Preserves the replication status of new, in-scope objects. By default, this value is used. This is the only value that can be used when the object specifier is added while the group is inactive.
 - *CURRENT—Begins replication of journal entries for new, in-scope objects at the time the object specifier is added. This value can be used only when the object specifier is added while the group is active.

- *REFRESH—Begins replication of journal entries for new, in-scope objects after an initial refresh of the objects. This value can be used only when the object specifier is added while the group is active.

12. In the **File update method** field, specify the method to update physical files.

Choose one of the following values:

- *RRN—Updates by relative record number. By default, this value is used.
- *UKEY—Updates by unique index. If you are updating by unique index, specify the name of the unique index (logical file) in the **PFKEY** parameter. If a unique index cannot be specified, then choose update by relative record number.

Note: This parameter is applicable when *FILE object types with an attribute of **PFDTA** are selected and when the **INCFLG** parameter is set to *INCLUDE.

13. In the **File unique key** field, specify the physical file's unique key.

This field is available only if the **PFUPDMTD** parameter is set to *UKEY and the **INCFLG** parameter is set to *INCLUDE.

The unique key must be specified when the update method is by unique key and the unique key is not specified as *AUTO.

Choose one of the following values:

- <name>—Specifies the unique key value.
- *AUTO—Specifies that a unique key is automatically determined for every physical file being replicated by the object specifier.

14. In the **File unique key** field, type the name of the library for the **File unique key**.

This field is available only if the **PFUPDMTD** parameter is set to *UKEY and the **INCFLG** parameter is set to *INCLUDE. The library must be specified when the update method is by unique key and the unique key is not specified as *AUTO.

15. In the **Mirror contents** field, specify whether to replicate the contents of the object are replicated.

The parameter applies only if the **OBJTYPE** parameter is *DTAARA, *DTAQ, or *FILE with the PFDTA attribute specified. Choose one of the following values:

- *YES—Refreshes and replicates the contents of the object. By default, this value is used.
- *NO—Does not refresh and replicate the contents of the object.

Note: For PFDTA files, iCluster does not force journaling them if you decide not to refresh or mirror the object contents with **MIRRCNTS(*NO)**.

16. When the automatic reactivation product system value is *YES, specify settings in the **Automatic Reactivation** field for the automatic reactivation of suspended objects that are replicated by this group for the object specifier.

Choose one of the following values:

- *GROUP—Keeps the group setting for this parameter. By default, this value is used.
- *NO—Does not attempt reactivation.

17. In the **Maximum reactivation attempts** field, specify the number of automatic retries before halting reactivation.

Choose one of the following values:

- <number>—Specifies the maximum number of retries. Valid entries range between 1 - 32767.
- *CLUSTER—Uses the system value that is assigned in the DMSETSVAL (Set Cluster System Values) command.
- *GROUP—Keeps the group setting for this parameter. By default, this value is used.
- *NOMAX—Never gives up on automatic reactivation; no limit for automatic retries.

-
18. In the **Maximum reactivation size** field, specify the largest object size to be included in the reactivation. Choose one of the following values:
- *CLUSTER—Uses the system value that is assigned in the DMSETSVAL (Set Cluster System Values) command.
 - *GROUP—Keeps the group setting for this parameter. By default, this value is used.
 - *NOMAX—Includes objects of all sizes in the reactivation. This value could have serious performance issues on the primary node if very large objects are locked frequently by other jobs.
 - <size>—Specifies the maximum size (in bytes) of an object.
19. In the **Activate files in ACTOOS state** field, specify how to reactivate files that are in an out-of-sync (ACTOOS) state. Choose one of the following values:
- *GROUP—Keeps the group setting for this parameter. By default, this value is used.
 - *CLUSTER—Uses the system value that is assigned in the DMSETSVAL (Set Cluster System Values) command.
 - *NO—Does not attempt to refresh records that are in an out-of-sync (ACTOOS) state.
 - *YES—Attempts to refresh records in error for files in an out-of-sync (ACTOOS) state.
20. In the **Maximum Record-Level Errors** field, specify the maximum number of record-level errors that are allowed to occur for a particular file before replication of the physical file stops and the file becomes suspended. Choose one of the following values:
- *CLUSTER—Uses the system value that is assigned in the DMSETSVAL (Set Cluster System Values) command.
 - *GROUP—Keeps the group setting for this parameter. By default, this value is used.
 - *NOMAX—Specifies that replication of physical files will continue regardless of the number of record-level errors generated.
 - *FIX—Specifies to continue replication of physical files when record-level errors occur during the backup apply process and to record the relative record number (RRN) for each failed record to enable reactivation at a later time. Files that become out-of-sync with record-level errors are marked as being in ACTOOS state and are listed as both active and suspended with a reason code of RLE.
 - <number>—Specifies the maximum number of record-level errors.
21. In the **Replicate *OUTQ Contents** fields, specify whether to mirror the contents of output queues. Specify the value of a refreshed output queue's AUTOSTRWTR option on the backup node in the **AUTOSTRWTR option** field. Choose one of the following values:
- *PRIMARY—Sets the same value on the backup node that is used on the primary node.
 - *NONE—Sets the value on the backup node to none, regardless of the value on the primary node.
22. In the **Replicate *OUTQ Contents** field, specify whether to mirror the contents of output queues. Choose one of the following values:
- *CLUSTER—Uses the system value that is assigned in the DMSETSVAL (Set Cluster System Values) command.
 - *NO—Does not mirror the spooled files. Replicates only the *OUTQ objects, but not the spooled files in them.
23. In the **File member(s) to exclude** field, specify the name of a file member or multiple file members to exclude. Set this parameter to a value other than *NONE only for an object specifier that references a single object with type *FILE.
- This parameter is available only for *INCLUDE specifiers. Choose one of the following values:
- *NONE—Exclude no members of the file. By default, this value is used.
 - <name>—Specifies the member name.
 - <generic*>—Specifies a generic object name to identify multiple members of the file.

-
24. In the **Hold *JOBSCD ent. on backup** field, specify whether to hold all of the job schedule entries for the replicated *JOBSCD objects on the backup node.

Choose one of the following values:

- *NO—Do not hold the job schedule entries on the backup node. The entries have the same status on the backup node as the primary node. By default, this value is used.
- *YES—Hold all of the job schedule entries for the replicated *JOBSCD objects on the backup node.

Related information

[DMSELOBJ \(Select Objects to Group command\)](#)

Selecting a directory entry to a group

Use the DMSELDIRE (Select Directory Entry) command to select system directory entries as shown through the IBM DSPDIRE command, to a replication group or a refresh-only group.

When you select directory entries based on the **userid** parameter to a specified group, the directory entries referenced are replicated from the primary node to the backup node in the replication group.

The directory entry specifiers and objects are listed in the iCluster native object panels (DMWRKOBJ, WRKHAOBJSTS, DMWRKOBJST and others) as object type *DIRE. These are not official IBM i object types but are used as a logical construct within iCluster to identify these entries.

The group must be inactive when you issue the DMSELDIRE (Select Directory Entry) command.

1. Enter the following command:

DMSELDIRE

2. Press Enter.

3. In the **Group** field, type the name of the group.

4. In the **Userid** field, specify the userid and address of the directory entry to select.

Choose one of the following values for **Userid**:

- *ALL—Specifies all directory entries with a specific address.
- *ANY—Specifies a directory entry with the special user identifier *ANY.
- <name>—Specifies a single user identifier.
- <generic*>—Specifies a generic user identifier.

Choose one of the following values for address:

- *ALL—Specifies all addresses.
- <name>—Specifies a single address.

5. In the **TGTADDR** field, specify the user address, system name and network userid that will be given to the replicated directory entries on the backup node.

Choose one of the following values:

- *PRIMARY—Specifies that the address and system name will be the same as on the primary node.
- *BACKUP—Specifies that address and system name will be the system name of the backup node. If this option is selected, the "Network userID" parameter for the directory entry will be the same as the user ID and address associated with the directory entry.
- <name>—Specifies the address and system name that will be given to the replicated directory entries matching this object specifier on the backup node.

6. In the **Description** field, specify a short description of up to 50 characters to identify the specifier.

7. In the **Include or exclude** field, specify whether the objects referenced by the specifier will be replicated within the group.

Choose one of the following values:

- *INCLUDE—Replicates referenced objects within the group when cluster operations are started. By default, this value is used.
- *EXCLUDE—Does not replicate referenced objects within the group when cluster operations are started.

Related information

[DMSELDIRE \(Select Directory Entry command\)](#)

Selecting a SQL object specifier to a group

Use the DMSELSQLO (Select SQL Objects) command to select a SQL object specifier to a replication group or a refresh-only group.

When you define a new SQL object specifier and select it to a group, the SQL objects that are referenced by the selected SQL object specifiers are replicated from the primary node to the backup node in the group.

1. On an active node in the cluster, enter the **DMSELSQLO** command.
2. In the **Group** field, type the name of the group.
3. In the **OBJ** field, specify the SQL object name component of the specifier to select.
A SQL object name can be up to 128 characters long. The possible values are:
 - <name>—Specifies the SQL object name.
 - <generic*>—Specifies a generic SQL object name to identify multiple SQL objects in a library. A generic SQL name is a character string that contains one or more characters followed by an asterisk (*).
 - *ALL—Specifies all SQL objects in a library.

The library where the SQL objects reside must be identified. Prefix the SQL object specification with the name of the library where the SQL objects are located (for example, LIB1/SQLOBJ1).

The possible values are:

- <name>—The library name.
- <generic*>—A generic library name to identify multiple libraries. For example, JDE* identifies all libraries that begin with JDE, such as JDEPROD, JDEDEV, and so on.
- *ALL—Specifies all libraries. All non-system-included libraries that contain SQL objects are listed in the iCluster SQL object panel, **DMWRKSQLO**.

Confirm if the specifiers meet your needs after running the command with the parameter Library set to *ALL.

If any specifiers are selected that you do not need, you must use the **DMCHGSQLSL INCFLG(*EXCLUDE)** to avoid the SQL objects that you do not need to be replicated. If you use the **DMDSELSQLO** command to delete the specifier, it may be reselected when the group starts.

Not all libraries are auto-selected. A library is not selected due to the following:

- Owned by the **DMCLUSTER** user profile.
- Owned by one of the **Aldon License Manager** user profiles and is not eligible for replication for Aldon LM high availability. (ACMSCTL and ACMSUSR libraries are eligible for replication).
- Licensed program product libraries and directories.
- iCluster work libraries.
- iCluster staging store libraries.
- Libraries and directories that are iCluster target libraries have **Library created by iCluster** as the

descriptive text (which means that the library was created by iCluster as a target library for replication). Earlier forms of the text are the **Library created by iCluster for i** and the **Library created by IBM iCluster**.

- Contains only *JRN and *JRNRCV.
 - System-owned library.
4. In the **OBJTYPE** field, type the SQL object type component of the specifier to select or *ALL.
 5. In the **DESC** field, enter a short description that identifies this SQL object specifier selection. Use up to 50 characters.
 6. In the **INCFLG** field, indicate whether to replicate the SQL objects that are referenced by the specifier. Specifying *INCLUDE means that the referenced SQL objects are replicated to the backup node when replication is started. Specifying *EXCLUDE prevents the referenced SQL objects from being replicated to the backup node.

Related information

[Replicating SQL objects](#)

[DMSELSQLO \(Select SQL Objects command\)](#)

Changing object specifier attributes

Change specific attributes of an existing object specifier at any time. Run the command on any node. The changes are propagated to all other nodes.

1. Enter the following command:
DMCHGOBJSL
2. In the **Group** field, type the name of the group.
3. In the **Object** field, specify the object name component of the specifier to select.
Choose one of the following values:
 - *ALL—Specifies all objects in a library.
 - <name>—Specifies the object name.
 - <generic*>—Specifies a generic object name, to identify multiple objects in a library.
4. In the **Library** field, type the name of the library where the objects reside.
5. In the **Object type** field, specify an object type.
Choose one of the following values:
 - *ALL—Specifies all objects in the library which has the same object name component as set in the previous step but different types.
 - <name>—Specifies an object type.
6. In the **Object extended attribute** field., specify the attributes of the chosen specifier.
Choose one of the following values:
 - *ALL—Specifies all object attributes. By default, this value is used.
 - <name>—Specifies the attribute type. Press F4 for a list of all values.
7. In the **Target library** field, specify the name of the library on the backup node.
Choose one of the following values:
 - *SAME—Keeps the current setting for this parameter.
 - <name>—Specifies the backup library name.
 - *PRIMARY—Uses the same backup library as the primary node library where the object resides.
 - *GROUP—Uses the backup library that is specified in the DMSTRRSS (Start Roleswitch Simulator) or DMCHGGRP (Change Group) commands. By default, this value is used.
8. In the **Description** field, specify a description. Choose one of the following actions:

- *SAME—Keeps the current setting for this parameter.
 - Type a short description that is up to 50 characters.
 - 9. In the **Include or exclude** field, specify whether the objects that are referenced by the specifier are replicated within the group.
Choose one of the following values:
 - *SAME—Keeps the current setting for this parameter.
 - *INCLUDE—Replicates referenced objects within the group when cluster operations are started.
 - *EXCLUDE—Does not replicate reference objects within the group when cluster operations are started.
 - 10. In the **Object polling interval** field, specify the polling interval for monitoring content changes to source physical files and user spaces.
Choose one of the following values:
 - *SAME—Keeps the current setting for this parameter.
 - *CLUSTER—Uses the system value assigned in the DMSETSVAL (Set Cluster System Values) command.
 - *GROUP—Uses the group value that is specified in the DMADDGRP (Add Group) or DMCHGGP (Change Group) commands.
 - *NONE—Specifies that pollable objects that match this specifier are not polled for content changes.
 - 11. In the **Mirror contents** field, specify if contents of the object are replicated.
Choose one of the following values:
 - *SAME—Keeps the current setting for this parameter.
 - *YES—Refreshes and replicates the contents of the object.
 - *NO—Does not refresh and replicate the contents of the object.
- Note:** For PFDTA files, iCluster does not force journaling them if you decide not to refresh or mirror the object contents with **MIRRCNTS(*NO)**.
- 12. In the **File update method** field, specify the method to update physical files.
Choose one of the following values:
 - *RRN—Updates by relative record number. By default, this value is used.
 - *UKEY—Updates by unique index. If you are updating by unique index, specify the name of the unique index (logical file) in the **PFKEY** parameter. If a unique index cannot be specified, then choose update by relative record number.

Note: This parameter is applicable when *FILE object types with an attribute of **PFDTA** are selected and when the **INCFLG** parameter is set to *INCLUDE.

 - 13. In the **File unique key** field, specify the unique key for the physical file.
Choose one of the following values:
 - *SAME—Keeps the current setting for this parameter.
 - *AUTO—Automatically determines a unique key for every physical file being replicated by the object specifier.
 - <key>—Specifies a unique key value.
 - 14. Ignore the **Data conflict winner** field and the **Conflict res. user exit** field. These parameters are no longer used.
 - 15. When the automatic reactivation product system value is *YES, specify settings in the **Automatic Reactivation** field for the automatic reactivation of suspended objects that are replicated by this group for this object specifier.
Choose one of the following values:
 - *SAME—Keeps the current setting for this parameter.

- *GROUP—Keeps the group setting for this parameter.
 - *NO—Does not attempt reactivation.
16. In the **Maximum reactivation attempts** field, specify the number of automatic retries before halting reactivation.
Choose one of the following values:
- *SAME—Keeps the current setting for this parameter.
 - <number>—Specifies the maximum number of retries. Valid entries range between 1 - 32767.
 - *CLUSTER—Uses the system value assigned in the DMSETSVAL (Set Cluster System Values) command.
 - *GROUP—Keeps the group setting for this parameter.
 - *NOMAX—Never gives up on automatic reactivation; no limit for automatic retries.
17. In the **Maximum reactivation size** field, specify the largest object size to be included in the reactivation.
Choose one of the following values:
- *SAME—Keeps the current setting for this parameter.
 - *CLUSTER—Uses the system value assigned in the DMSETSVAL (Set Cluster System Values) command.
 - GROUP—Keeps the group setting for the parameter.
 - *NOMAX—Includes objects of all sizes in the reactivation. This value could have serious performance issues on the primary node if very large objects are locked frequently by other jobs.
 - <size>—Specifies the maximum size (in bytes) of an object.
18. In the **Activate files in ACTOOS state** field, specify how to reactivate files that are in an out-of-sync (ACTOOS) state.
Choose one of the following values:
- *CLUSTER—Uses the system value assigned in the DMSETSVAL (Set Cluster System Values) command.
 - GROUP—Keeps the group setting for the parameter.
 - *SAME—Keeps the current setting for this parameter.
 - *NO—Does not attempt to refresh records that are in an out-of-sync (ACTOOS) state.
 - *YES—Attempts to refresh records in error for files in an out-of-sync (ACTOOS) state.
19. In the **Maximum Record-Level Errors** field, specify the maximum number of record-level errors that are allowed to occur for a particular file before replication of the physical file stops and the file becomes suspended.
Choose one of the following values:
- *SAME—Keeps the current setting for this parameter.
 - *CLUSTER—Uses the system value assigned in the DMSETSVAL (Set Cluster System Values) command.
 - GROUP—Keeps the group setting for the parameter.
 - *NOMAX—Specifies that replication of physical files will continue regardless of the number of record-level errors generated.
 - *FIX—Specifies to continue replication of physical files when record-level errors occur during the backup apply process and to record the relative record number (RRN) for each failed record to enable reactivation at a later time. Files that become out-of-sync with record-level errors are marked as being in ACTOOS state and are listed as both active and suspended with a reason code of RLE.
 - <number>—Specifies the maximum number of record-level errors.
20. In the **AUTOSTRWTR option** field, specify whether to mirror the contents of output queues in the **Replicate *OUTQ Contents** fields. Specify the value of a refreshed output queue's AUTOSTRWTR option on the backup node.
Choose one of the following values:
- *SAME—Keeps the current setting for this parameter.
 - *PRIMARY—Sets the same value on the backup node that is used on the primary node.
 - *NONE—Sets the value on the backup node to none, regardless of the value on the primary node.
21. In the **Replicate *OUTQ Contents** field, specify whether to mirror the contents of output queues.

- Choose one of the following values:
- *SAME—Keeps the current setting for this parameter.
 - *CLUSTER—Uses the system value assigned in the DMSETSVAL (Set Cluster System Values) command.
 - *NO—Does not mirror the spooled files. Replicates only the *OUTQ objects, but not the spooled files in them.
22. In the **File member(s) to exclude** field, specify the name of a file member or multiple file members to exclude.
Choose one of the following values:
- *SAME—Keeps the current setting for this parameter.
 - *NONE—Exclude no members of the file.
 - <name>—Specifies the member name.
 - <generic*>—Specifies a generic object name to identify multiple members of the file.
23. In the **Hold *JOBSCD ent. on backup** field, specify whether to hold all of the job schedule entries for the replicated *JOBSCD objects on the backup node.
Choose one of the following values:
- *NO—Do not hold the job schedule entries on the backup node. The entries have the same status on the backup node as the primary node. By default, this value is used.
 - *YES—Hold all of the job schedule entries for the replicated *JOBSCD objects on the backup node.

Related information

[DMCHGOBJSL \(Change Object Selection command\)](#)

Deselecting an object specifier from a group

Deselecting an object specifier from a group prevents referenced objects from being replicated or excluded from replication within the group. Use the DMDSELOBJ (Deselect Objects from Group) command to deselect object specifiers from a group.

The group must be inactive when you issue the DMDSELOBJ (Deselect Objects from Group) command.

1. Enter the following command:
`DMDSELOBJ`
2. Press Enter.
3. In the **Group** field, type the name of the inactive group.
4. In the **Object** field, specify the object name component of the specifier to select.
Choose one of the following values:
 - *ALL—Specifies all objects in a library.
 - <name>—Specifies an object name.
 - <generic*>—Specifies a generic object name, to identify multiple objects in a library.
5. In the **Library** field, type the name of the library where the objects reside.
6. In the **Object type** field, specify an object type.
Choose one of the following values:
 - *ALL—Specifies all objects in the library which has the same object name component as set in the previous step but different types.
 - <name>—Specifies an object type.
7. In the **Object extended attribute** field, specify the attributes of the chosen specifier.
Choose one of the following values:
 - *ALL—Specifies all object attributes. By default, this value is used.
 - <name>—Specifies an attribute type. Press F4 for a list of all values.

Related information

[DMDSELOBJ \(Deselect Objects from Group command\)](#)

Deselecting a directory entry from a group

Use the DMDSELDIRE (Deselect Directory Entry) command to deselect system directory entries from a replication group or refresh-only group. When you deselect directory entries from a specified group, the replication of the referenced directory entries is stopped.

The directory entry specifiers and objects are listed in the iCluster native object panels (DMWRKOBJ, WRKHAOBJSTS, DMWRKOBJST and others) as object type *DIRE. These are not official IBM i object types but are used as a logical construct within iCluster to identify these entries.

This command must be invoked on an active node in the cluster. Also, the specified replication group must be inactive when you issue the DMDSELDIRE (Deselect Directory Entry) command.

1. Enter the following command:

DMDSELDIRE

2. Press Enter.

3. In the **Group** field, type the name of the group.

4. In the **Userid** field, specify the userid and address of the directory entry to deselect.
Choose one of the following values for **Userid**:

- *ALL—Specifies all directory entries with a specific address.
- *ANY—Specifies a directory entry with the special user identifier *ANY.
- <name>—Specifies a single user identifier.
- <generic*>—Specifies a generic user identifier.

Choose one of the following values for **Address**:

- *ALL—Specifies all addresses.
- <name>—Specifies a single address.

Related information

[DMDSELDIRE \(Deselect Directory Entry command\)](#)

Reconfiguring object specifiers

To automatically reconfigure object specifiers for each library, default journals, and set the product system values, use the DMAUTOCFG (Auto-configure iCluster) command as an option to using the DMSELOBJ (Select Objects to Group) command.

Use an automatic configuration to change your object specifier configuration to create and select the following object specifiers to the SYSTEM replication group:

- All user profiles (excluding DMCLUSTER)
- All authorization lists (excluding lists with names that begin with the letter 'Q')
- All output queues (excluding queues with name that begin with the letter 'Q')
- The QGPL library (excluding objects with names that begin with the letter 'Q')
- Adds object specifiers to the SYSTEM group

1. For each node, ensure that the XDMCLUSTER subsystem is active on both nodes where iCluster is installed.

2. For each node, ensure that the DMCHATL job is active in the XDMCLUSTER subsystem.
 3. Enter the following command:
DMAUTOCFG
 4. To select object specifiers only, specify *YES in the **Select object specifiers only** field. Selecting *YES runs the DMAUTOCFG (Auto-configure iCluster) command to select object specifiers only. The groups, nodes, and other objects must exist. The command assumes all nodes, groups and journals have been created and displays user libraries so that you can select which libraries should be selected to which group.
 5. In the **Object spec. library size** field, specify whether to display the size of object specifier libraries and the number of objects in each library when entering this command.
Choose one of the following values:
 - *YES—Displays the size of each object specifier library and the number of objects in each library.
- Note:** The time to display this information depends on the number of libraries on the system.
- *NO—Does not display the size and number of objects. By default, this value is used.

Related information

[DMAUTOCFG \(Auto-configure iCluster command\)](#)

Viewing object specifiers

Use the **Work with Object Specifiers** screen to view a list of defined object specifiers.

Use one of the following commands to display the **Work with Object Specifiers** screen to view object specifiers or object specifiers in a group:

- List of all object specifiers:

```
DMWRKOBJ BY(*NONE)
```

- List of object specifiers for a group:

```
DMWRKOBJ BY(*GROUP) GROUP(<group name>)
```

Related information

[DMWRKOBJ \(Work with Object Specifiers command\)](#)

Path object specifiers

Use path object specifiers to identify the location of Byte Stream File (BSF) objects in the Integrated File System (IFS) or QDLS file system.

IFS is an IBM i feature that supports a hierarchical directory structure similar to that found on personal computers and UNIX systems. The IFS integrates support for a number of file systems, such as QDLS, the "root" file system, and the QOpenSys file system.

Use IFS to port PC or UNIX-based applications to a Power System running IBM i. IFS is the name used to refer to the file system, but references to the actual BSF objects that reside in this system will be through path object specifiers.

Select path object specifiers to groups. You can select object specifiers and path object specifiers to the same group, although it recommended to use separate groups for native and path object specifiers.

IFS Limitations

The following limitations apply to replicating objects in the IFS:

- Hard links are not supported.
- Replicating to a different directory on the backup node is not supported.
- DLO extended object attributes are not replicated.
- The maximum length of path names is 5000 bytes.
- The **Lock Files on Backup Node** and **Maximum Refresh Size** system values are not supported for path object specifiers.

Naming conventions for path object specifiers

A path object specifier is the path name to the BSF objects on the primary node (for example, '/Dir1/Dir2/Dir3/Dir4/file' or '/Dir1/Dir2/Dir5/Dir6/*').

You must use the following naming conventions for path object specifiers:

- Enclose the path name in single quotation marks (').
- The path must start with a forward slash (/) character ('/Dir1').
- The path cannot end with a forward slash (/) character ('/Dir3/Dir4/file').
- The path can have a maximum of 5000 characters. The path must be a minimum of two characters in length.
- Generic path names of the form '/mydir*' are supported.
- Generic path names of the form '//*' are supported and match the entire IFS subdirectory or directories in the path.
- Generic path names support 1 - 4 character file name extensions.
- To achieve the best results, use the longest non-generic path name as possible.

Non-journalized and journaled path object specifiers

Non-journalized BSF support is applicable for applications that create large numbers of files and replace existing files with a new version. In addition, objects in the /QDLS file system, or Document Library Objects (DLO), cannot be journaled to a database journal.

Journaled BSF support is applicable for applications that store modifiable data in BSF objects. These applications change the contents of the data contained in the BSF objects. Journaling is required to replicate those changes in real time.

Working with path object specifiers

To add a path object specifier to a group, supply the full path where one or more Byte Stream File (BSF) objects reside in the Integrated File System (IFS). It is important to identify the correct path so that you can address the correct set of objects.

The parameters of a path object specifier can be changed at any time. However, the path specifier must be selected to the specified group.

When a path specifier is deselected from a group, the path objects referenced by selected specifiers are no longer replicated from the primary node to the backup node in the group.

Adding a path object specifier to a group

Use the DMADDBSF (Add Path Specifier to Group) command to add a path object specifier to a group.

1. Enter the following command:

DMADDBSF

2. In the **Group** field, type the name of the group.

3. In the **Path specifier** field, specify the path that identifies the location of the BSF objects. Enclose the path in single quotes and start with a forward slash character. Use up to 5000 characters.

4. In the **Description** field, add a short description that identifies the specifier. Use up to 50 characters.

5. In the **Include or exclude** field, specify whether the BSF objects referenced by the path object specifier will be replicated within the group.

Choose one of the following values:

- *INCLUDE—Replicates referenced BSF objects within the group when cluster operations are started. By default, this value is used.
- *EXCLUDE—Does not replicate referenced BSF objects within the group when cluster operations are started.

6. In the **Journal** field, specify how the BSF objects referenced by the path object specifier will be replicated within the group.

Choose one of the following values:

- *NONE—Does not journal BSF objects matching this object specifier. By default, this value is used.
- *GROUP—Journals the BSF objects matching this object specifier using the default BSF journal specified at the group level.

7. In the **Object polling interval** field, specify the polling interval value for the DLO objects that are referenced by the path object specifier.

Choose one of the following values:

- *GROUP—Uses the group value specified in the DMADDGRP (Add Group) or DMCHGGRP (Change Group) commands. By default, this value is used.
- *NONE—Does not poll DLO objects that are referenced by this path specifier.

8. Press Enter.

9. In the **Activation of new objects** field, choose the method by which replication will begin for the objects that come into replication scope when an object specifier is added.

Choose one of the following values:

- *NONE—Does not change the replication status of new, in-scope objects. This is the default value when the group is *INACTIVE.
- *CURRENT—Begins replication of journal entries for new, in-scope objects at the time the object specifier is added.
- *REFRESH—Begins replication of journal entries for new, in-scope objects for each object after an initial refresh of the objects.

Related information

[DMADDBSF \(Add Path Specifier to Group command\)](#)

[DMADDGRP \(Add Group command\)](#)

[DMCHGGRP \(Change Group command\)](#)

Removing a path object specifier

Use the DMRMVBSF (Remove Path Specifier from Group) command to remove a path object specifier.

1. Enter the following command:
DMRMVBSF
2. In the **Group** field, type the name of the group.
3. In the **Path specifier** field, specify the path that identifies the location of the BSF objects. Enclose the path in single quotes and start with a forward slash character. Use up to 5000 characters.

Related information

[DMRMVBSF \(Remove Path Specifier from Group command\)](#)

Changing path object specifier attributes

Use the DMCHGBSF (Change Path Specifier to Group) command to change the attributes of a path object specifier.

1. Enter the following command:
DMCHGBSF
2. Press Enter.
3. In the **Group** field, type the name of the group.
4. In the **Path specifier** field, specify the path that identifies the location of the BSF objects. The path must be enclosed in single quotes and start with a forward slash character. Use up to 5000 characters.
5. Press Enter.
6. In the **Include or exclude** field, specify whether the BSF objects referenced by the path object specifier will be replicated within the group.
Choose one of the following values:
 - *SAME—Keeps the current setting for this parameter.
 - *INCLUDE—Replicates referenced BSF objects within the group.
 - *EXCLUDE—Does not replicate referenced BSF objects within the group.
7. In the **Journal** field, specify how the BSF objects referenced by the path object specifier will be replicated within the group.
Choose one of the following values:
 - *SAME—Keeps the current setting for this parameter.
 - *NONE—Does not journal BSF objects matching this object specifier.
 - *GROUP—Journals BSF objects matching this object specifier using the journal specified at the group level.
8. In the **Object polling interval** field, specify the polling interval value for the DLO objects that are referenced by the path object specifier.
Choose one of the following values:
 - *SAME—Keeps the current setting for this parameter.
 - *GROUP—Uses the group value specified in the DMADDGRP (Add Group) or DMCHGGRP (Change Group) commands.
 - *NONE—Does not poll DLO objects that are referenced by this path specifier.

Related information

[DMCHGBSF \(Change Path Specifier to Group command\)](#)

Replicating database *FILE objects

Review and understand the considerations for replicating physical files.

Refreshing physical files

Replication uses a refresh-while-active method to refresh physical files to the backup node. In other words, *FILE objects can be refreshed as they are being updated. This is the only refresh option available for physical files during a regular group refresh or replication. Out-of-sync or suspended physical files can also be refreshed with a save-restore option on the activation command.

During a refresh of a database file, changes to the contents of the file may continue. Changes at the member level and file level (a rename or move operation, for example) may be delayed, or they might fail during a refresh. The length of time that is required to refresh a file depends on the number of members in the file, the amount of data in the file, and the communications bandwidth available to the job that performs the refresh.

Refreshing logical files

The refresh of logical files does not include saving and restoring the access paths that are associated with the logical files. The access paths are rebuilt on the backup node when the logical file is restored.

Dependent logical files are automatically refreshed when the associated physical file is refreshed. In this case, the logical files on the secondary system are deleted prior to the restore of the based on physical file, and restored after the data is refreshed to the physical file. Note that there might be a relatively long period of time between deleting the logical file and restoring the new version of the logical file. During this period, the logical file appears in the monitor in a pending state.

If a logical file has a unique key, updates to the based on physical file will be delayed until the access path is completely built after the logical file is restored. Updates to all files are applied in time order relative to the primary system, so the rebuild of a uniquely keyed access path may delay the updates to other files associated with the same journal.

Unique key update method

The use of a unique key update method allows you to reorganize physical files on the primary node without affecting the backup files.

If a physical file is a multimember file, the members of the unique index used for its replication must have the same names as the members in the physical file. Files that have a unique key update method must be journaled with *BOTH images.

Journaled objects

Journaled objects are database files, data areas, and data queues. BSF files can also be journaled, but are managed differently than other journaled objects.

Referential integrity (RI) constraints

When replicating RI constraints, the following replication rules apply:

- Create and delete operations are replicated.
- RI constraints are disabled on the backup node.
- When performing a switchover, RI constraints are enabled on the new primary node.

- When starting replication after a switchover, RI constraints are disabled on the new backup node.

Triggers

When replicating triggers, the following replication rules apply:

- All operations on triggers are replicated.
- Both native and SQL triggers are supported by iCluster.
- Triggers are disabled on the backup node.
- iCluster keeps track of the status of triggers in replication scope on the primary node in its metadata.
- When performing a switchover, triggers are enabled on the new primary node unless they were disabled on the previous primary node.
- When starting replication after a switchover, triggers are disabled on the new backup node.
- The maximum length of a SQL CREATE TRIGGER statement that can be replicated is 65535 bytes.

Check constraints

All operations for check constraints are replicated. If you add, remove, enable, or disable constraints on the primary system, these changes are replicated to the backup system.

Replicating system-period temporal tables and their associated history tables

Before starting replication of system-period temporal tables, you must run the **CHGFCNUSG** command on each node where the replicated system-period temporal tables are replicated from, and on each backup node where they are replicated to. This only has to be done once on each node. After the command has run, iCluster can replicate any system-period temporal tables and their associated history tables. To enable the replication of system-period temporal files, run the following command on each node in the cluster:

```
CHGFCNUSG FCNID(QIBM_DB_SECADM) USER(DMCLUSTER) USAGE(*ALLOWED)
```

To replicate a system-period temporal table, select both the table and its associated history table to the same group. If you use generic specifiers, make sure that the generic specifiers for the system-period temporal table and its associated history table are selected to the same group.

The system-period temporal table and its associated history table must be journaled to the same database journal. If the tables are not journaled when you begin replicating them, the iCluster product will ensure that they are journaled correctly for replication.

Note: System-period temporal tables are only supported on IBM i V7R3 and higher operating system releases.

Other considerations

- iCluster creates a journal on the backup node for physical files if one has not been created already. The journal on the backup node will have the same name and reside in the library with the same name as the corresponding journal on the primary node.
- iCluster creates a work library on the backup node for each apply job. These libraries are used to temporarily hold some replicated objects prior to restoring them. These libraries follow a naming convention of **Hxxxxxxxxx**, where **xxxxxxx** is digits.

The text that is associated with these libraries includes the name of the backup node, group name, journal

name, journal library, and the journal entry. Do not delete these libraries that are considered to be part of the staging store. Work libraries are also created on the primary node for each journal scrape job which follow the same naming convention. However, these libraries are temporary, and the libraries are deleted when they are no longer needed.

- Authority holders are replicated for *FILE objects only.
- The CHGPF command is not replicated when a source file is specified. The following attributes are replicated for the CHGPF and CHGLF commands: MAXMBRS, MAINT, RECOVER, FRCRATIO, and TEXT.
- When mirroring files that have user defined SQL data types (SQL DISTINCT TYPEs), ensure that the *SQLUDT object is also in mirroring scope. The file will become suspended unless the corresponding *SQLUDT object exists on the backup system.

Replicating database *FILE objects

Review and understand the considerations for replicating physical files.

Refreshing physical files

Replication uses a refresh-while-active method to refresh physical files to the backup node. In other words, *FILE objects can be refreshed as they are being updated. This is the only refresh option available for physical files during a regular group refresh or replication. Out-of-sync or suspended physical files can also be refreshed with a save-restore option on the activation command.

During a refresh of a database file, changes to the contents of the file may continue. Changes at the member level and file level (a rename or move operation, for example) may be delayed, or they might fail during a refresh. The length of time that is required to refresh a file depends on the number of members in the file, the amount of data in the file, and the communications bandwidth available to the job that performs the refresh.

Refreshing logical files

The refresh of logical files does not include saving and restoring the access paths that are associated with the logical files. The access paths are rebuilt on the backup node when the logical file is restored.

Dependent logical files are automatically refreshed when the associated physical file is refreshed. In this case, the logical files on the secondary system are deleted prior to the restore of the based on physical file, and restored after the data is refreshed to the physical file. Note that there might be a relatively long period of time between deleting the logical file and restoring the new version of the logical file. During this period, the logical file appears in the monitor in a pending state.

If a logical file has a unique key, updates to the based on physical file will be delayed until the access path is completely built after the logical file is restored. Updates to all files are applied in time order relative to the primary system, so the rebuild of a uniquely keyed access path may delay the updates to other files associated with the same journal.

Unique key update method

The use of a unique key update method allows you to reorganize physical files on the primary node without affecting the backup files.

If a physical file is a multimember file, the members of the unique index used for its replication must have the same names as the members in the physical file. Files that have a unique key update method must be journaled with *BOTH images.

Jounaled objects

Jounaled objects are database files, data areas, and data queues. BSF files can also be jounaled, but are managed differently than other jounaled objects.

Referential integrity (RI) constraints

When replicating RI constraints, the following replication rules apply:

- Create and delete operations are replicated.
- RI constraints are disabled on the backup node.
- When performing a switchover, RI constraints are enabled on the new primary node.
- When starting replication after a switchover, RI constraints are disabled on the new backup node.

Triggers

When replicating triggers, the following replication rules apply:

- All operations on triggers are replicated.
- Both native and SQL triggers are supported by iCluster.
- Triggers are disabled on the backup node.
- iCluster keeps track of the status of triggers in replication scope on the primary node in its metadata.
- When performing a switchover, triggers are enabled on the new primary node unless they were disabled on the previous primary node.
- When starting replication after a switchover, triggers are disabled on the new backup node.
- The maximum length of a SQL CREATE TRIGGER statement that can be replicated is 65535 bytes.

Check constraints

All operations for check constraints are replicated. If you add, remove, enable, or disable constraints on the primary system, these changes are replicated to the backup system.

Replicating system-period temporal tables and their associated history tables

Before starting replication of system-period temporal tables, you must run the **CHGFCNUSG** command on each node where the replicated system-period temporal tables are replicated from, and on each backup node where they are replicated to. This only has to be done once on each node. After the command has run, iCluster can replicate any system-period temporal tables and their associated history tables. To enable the replication of system-period temporal files, run the following command on each node in the cluster:

```
CHGFCNUSG FCNID(QIBM_DB_SECADM) USER(DMCLUSTER) USAGE(*ALLOWED)
```

To replicate a system-period temporal table, select both the table and its associated history table to the same group. If you use generic specifiers, make sure that the generic specifiers for the system-period temporal table and its associated history table are selected to the same group.

The system-period temporal table and its associated history table must be journaled to the same database journal. If the tables are not journaled when you begin replicating them, the iCluster product will ensure that they are journaled correctly for replication.

- !** **Note:** System-period temporal tables are only supported on IBM i V7R3 and higher operating system releases.

Other considerations

- iCluster creates a journal on the backup node for physical files if one has not been created already. The journal on the backup node will have the same name and reside in the library with the same name as the corresponding journal on the primary node.
- iCluster creates a work library on the backup node for each apply job. These libraries are used to temporarily hold some replicated objects prior to restoring them. These libraries follow a naming convention of **Hxxxxxxx**, where **xxxxxxx** is digits.

The text that is associated with these libraries includes the name of the backup node, group name, journal name, journal library, and the journal entry. Do not delete these libraries that are considered to be part of the staging store. Work libraries are also created on the primary node for each journal scrape job which follow the same naming convention. However, these libraries are temporary, and the libraries are deleted when they are no longer needed.

- Authority holders are replicated for *FILE objects only.
- The **CHGPF** command is not replicated when a source file is specified. The following attributes are replicated for the **CHGPF** and **CHGLF** commands: MAXMBRS, MAINT, RECOVER, FRCRATIO, and TEXT.
- When mirroring files that have user defined SQL data types (SQL DISTINCT TYPEs), ensure that the *SQLUDT object is also in mirroring scope. The file will become suspended unless the corresponding *SQLUDT object exists on the backup system.

Related information

[Supported object types for replication](#)

Replicating BSF objects

Byte Stream File (BSF) objects reside in the Integrated File System (IFS) and can be replicated by iCluster through the use of path object specifiers.

Note the following important issues and considerations concerning BSF objects.

Mirroring BSF Objects

iCluster supports two types of replication for BSF objects: object-level mirroring, and content-level mirroring. The type of mirroring used for BSF objects matching a particular object specifier is determined by the **JOURNAL** parameter of the object specifier.

The following points are worth considering when deciding on the type of replication that is most appropriate for your needs:

- iCluster supports content-level and object-level replication in the "root" ("/") and QOpenSys file systems. Content-level replication and object-level replication is also supported in the QDLS (DLO objects) file system.
- Content-level replication includes object-level replication support and also replication of changes to the contents of BSF objects. Content-level replication requires that the BSF objects are journaled and is, therefore, more resource intensive than object-level replication.
- BSF polling allows content-level replication of objects in the QDLS file system. No journaling is required.
- Object-level replication will capture changes to the object at the object level only including: creates, deletes,

moves, renames, restores, authority changes, ownership changes, and primary group changes.

- In object-level replication, changes to the contents of the object are not captured. Object-level replication is suitable for applications that create BSF objects but do not change the contents of those objects, or change the contents by creating a completely new version of the object.
- You can set the system value for content changes to non-journalized BSF to allow content changes for non-journalized BSF objects to be replicated by a full refresh of the changed file. However, this method is even more resource-intensive than journalized BSF replication.
- The overlap of journalized (*GROUP) and non-journalized (*NONE) path object specifiers is restricted in iCluster. For example, using both '/home/user/*' (non-journalized) and '/home/user/employees/*' (journalized) is not supported.
- The number of objects (BSF and/or native) that may be journalized to a particular journal is limited. Consider using object-level replication if you have a large number (100,000 or more) of BSF objects to be replicated.

Initial Synchronization of Path Objects

Initial synchronization of path objects between the primary and backup nodes involves the same considerations that are required for the initial synchronization of database files. Replication of journalized path objects involves sending to the backup node only the portion of the object that was changed, therefore the changed object must exist on the backup node prior to applying the change.

To start replication for existing path objects, use one of the following methods to synchronize the primary and backup nodes:

- Refresh the objects through iCluster. The easiest way to synchronize the primary and backup nodes is to initially refresh all objects before replication is started. Thereafter, when replication is started, only the capture of changes is required. The disadvantage of this approach is that the amount of data to refresh across the communications line might be large and the process might take a long time.
- Manually synchronize all objects. The second method to synchronize the primary and backup nodes is to manually save the objects on the primary node and restore them on the backup node. The starting journal position is established using DMMRKPOS (Mark Journal Positions) after saving the primary node objects and before changes are made to these objects.

BSF objects that are created on the primary node after the initial synchronization are automatically refreshed to the backup node if they are referenced by path object specifiers selected to groups.

Journaling Path Objects

Journaling is started for an object that is referenced by a path object specifier to the replication group's default BSF journal, if the object is refreshed and it is not already journalized. iCluster will also start journaling an object referenced by a path object specifier to the default BSF journal when you invoke the DMMRKPOS (Mark Journal Positions) command. After journaling has been started, most changes to that object can be mirrored without requiring access to the object on the primary node.



Note: Journaling changes to path objects requires resources on the primary node both to record (CPU) and store (disk) the changes. Journaling the changes to path object specifiers impacts the performance of applications that use the path object specifiers; just as journaling the database files has a performance impact on applications that use the database files.

Refresh (Before Mirroring) of BSF Objects

The following points should be considered when doing a refresh (before mirror) of BSF objects:

- Prior to the refresh, iCluster will attempt to delete all objects on the secondary system that match the object

specifiers.

- Refresh before mirroring will create base directories if they do not exist. For an object specifier of '/home/objects/journaled/*', iCluster will create the following base directories: '/home', '/home/objects', and '/home/objects/journaled'.

Full Cluster Status Monitor Support for BSF Objects

Suspended BSF objects will be shown in the Full Cluster Status Monitor and may be activated from the object status screen or activated automatically by iCluster.

iCluster Limitations for BSF Objects

iCluster does not provide support in the following areas:

- Hard links are not supported.
- BSF include/exclude specifiers cannot have symbolic links to directories as part of their paths.
- Replication must be to the same directory on the backup node. iCluster does not support directory redirection of BSF objects.
- It is recommended that you replicate all BSF objects within a group to the same journal if they are journaled. When using separate journals there is no guarantee that replication of file operations will occur in the same order in which they occurred on the primary node.
- The maximum length of path names is 5000 bytes (IBM i supports 16 MB path names).
- Two cluster system values, **Lock File on Backup Node** and **Maximum Refresh Size**, are not supported for path object specifiers.
- DLO extended object attributes are not replicated by iCluster.

Related information

[Generic object specifiers](#)

[Monitoring overview](#)

Replicating configuration objects

When configuration objects are replicated to a node, they can be created immediately after they are received or at a later time.

Configuration objects collectively refer to the following object types:

- Connection lists (*CNXL)
- Class-of-service descriptions (*COSD)
- Controller descriptions (*CTLD)
- Device descriptions (*DEVD)
- Internet package exchange descriptions (*IPXD)
- Line descriptions (*LIND)
- Mode descriptions (*MODD)
- NetBIOS descriptions (*NTBD)
- Network interface descriptions (*NWID)
- Network server descriptions (*NWSID)

Creating Configuration Objects

When configuration objects are replicated to a node, you can request iCluster to automatically create these objects immediately after they are received or create them at a later time. If a configuration object already exists on a node, the DMSETVAL (Set Cluster System Values), DMADDNODE (Add Node), and DMADDGRP (Add Group) commands have options that allow you to avoid creating a configuration object if it is currently in use. As a result, a set of source physical files (named CNNL, COSD, CTLD, DEVD, IPXD, LIND, MODD, NTBD, NWID, and NWSID) is provided that are used to hold the source code for generating configuration objects on each node in the cluster. These source physical files are all located in the product library. If, at a later time, you want to create these objects, issue the CRTCFGOBJ (Create Configuration Objects) command.

Identifying Configuration Objects

When defining object specifiers, no special treatment is required to identify configuration objects. The configuration object type can be specified in exactly the same way as all other supported types. However, you cannot use the generic type *ALL to identify configuration objects.

Replicating user profiles

Be aware of the recommendations and constraints of replicating user profiles.

When replicating user profiles with iCluster, note the following:

- When user profiles (*USRPRF) are replicated to backup nodes, iCluster does not replicate associated document passwords. The user must explicitly arrange for document passwords to be defined and updated on the backup node.
- It is recommended that all non-system user profiles (*USRPRF) in library QSYS are replicated to backup nodes. This will ensure that the owners of replicated objects are the same on both primary and backup nodes.
- When mirroring changes to user profiles (*USRPRF) that belong to a group, you need to set the parameter JOBMSGQFL to *WRAP for job description used for replication jobs (usually CSJOBID in the ICLUSTER library) on the backup node. Setting this option prevents the job message queue from filling up.
- Since one or more group profiles can be associated with a single user profile object (*USRPRF), it is important to note that all related group profiles are also sent to the backup node when a user profile is replicated by iCluster when the user profile replication level is *FULL.
- User profiles that are owned by the operating system, for example, QSYS, are not replicated by iCluster.
- The command defaults of the **MAXSTGLRG** and **MAXSTG** parameters of the **CRTUSRPRF** command should be set to *NOMAX.

Replicating LOBs

Use a Large Object (LOB) field to store a large amount of unformatted data in a database.

Previously, you might have excluded the files with LOB fields because they could have been suspended. You can include these files with LOB fields and they are replicated just as any other data file. To replicate LOBs, perform the same steps as for replicating other data in iCluster.

Be aware of the following considerations for LOBs:

- The staged LOB data is stored on the backup node in IFS in BSF objects. iCluster stores LOB data in the following location:

```
/home/DataMirror/HASUITE/LOB/00000000/<node name>/<group name>/<journal library>/<journal n
```

- The maximum store size on the backup node is not respected when it is creating the BSF objects for LOB staging. LOB data can be replicated as long as there is sufficient disk space to contain the LOB data.
- iCluster does not impose limitations in addition to those imposed by IBM i for LOB fields in a record format. There can be a total of 1023 LOB fields in a record format. The total size of the LOB fields cannot exceed two gigabytes.
- We recommend setting all journals to use the highest receiver size possible to ensure successful journaling of files containing LOB data. To do this, set the receiver size to *MAXOPT3.
- To obtain better performance, use local journals when commitment control is not required.

Related information

- [Replicating BSF objects](#)
[Large object \(LOB\) constraints](#)

Large object (LOB) constraints

Verify that the journals of Large Object Fields (LOBs) meet certain constraints.

The following constraints apply to LOBs:

- The default journal for these files must have the **RCVSIZOPT** parameter of the **CHGJRN** command set to at least *MAXOPT2, otherwise the file is suspended.
- It is recommended that all journals use the highest receiver size option possible.
- **MINENTDTA** (Minimize entry specific data) cannot be specified for journals that are journaling files with LOB fields. Only **MINENTDTA *NONE** is supported for files with LOB fields.

Replicating triggers

Replication is supported for system triggers and SQL triggers.

Consider the following guidelines, which apply to replicating triggers:

- The **ADDPFTRG** and **RMPFTRG** operating system commands control system triggers.
- SQL triggers are created and dropped through SQL statements on tables.
- All triggers on a physical file are disabled when the physical file is replicated to the backup node. When a switchover occurs, triggers that were enabled on the previous primary node are enabled on the new primary node.
- Up to 300 triggers are supported per file.
- Trigger names can be up to 128 characters.
- INSTEAD OF triggers on SQL views are replicated, but are not disabled for the logical files on the backup node.

Using the SAVFEXIT user exit program when replicating save files

iCluster supports the replication of save file (SAVF) objects and allows you to automatically perform actions on the backup node when the save file is replicated to the backup node. Objects that cannot be replicated directly with iCluster can be saved in a save file and then replicated by transferring the save file to the backup node.

When a save file is received on a backup node, iCluster automatically invokes a predefined program called SAVFEXIT. You can write this program to perform specific actions on the save file. The name and library of the replicated save file are passed as parameters to SAVFEXIT. The program can then use these parameters to determine what kind of actions should be performed on the save file. For example, if the replicated save file contains a document library object, SAVFEXIT can be modified to detect the save file and then restore the DLO by issuing the RSTDLO command.

The SAVFEXIT program has to be created after the product is installed. Issue the following command to create this program:

```
CRTCLPGM PGM (ICLUSTER/SAVFEXIT) SRCFILE(<iCluster product library>/QACLSRC)
```

The ICLUSTER library is where iCluster was installed. If you want iCluster to invoke the SAVFEXIT program automatically, it must be placed in the product library on the backup node before replication is started.

If you do not want to perform any actions on the save file, remove or delete the SAVFEXIT program after stopping all replication. To replace the current SAVFEXIT program with a new program, you must first stop replication to ensure that the new program is invoked when replication is restarted.

Sample source code is provided that can be used to replicate objects by using the savefile. These code segments are located in the file QACLSRC in your product library. The following table identifies and briefly describes each code segment.

Name	Description
SAVE_DLO	Saves document library objects (DLO) into a named save file. This program resides on the primary node. When the SAVE_DLO code is called, the specified DLOs are saved into the save file. These DLOs are then saved at specified intervals. If the save file is being mirrored to a backup node, all specified DLOs are replicated after they are saved in the save file.
SAVE_IFS	Saves Integrated File System objects (IFS) into a named save file. This program resides on the primary node. When the SAVE_IFS code is called, the specified directory is saved into the save file (including all directories and objects in the directory). Subsequent updates that are applied to these directories and objects are saved at specified intervals. If the save file is being mirrored to a backup node, any subsequent updates are replicated through the save file.
SAVFEXIT01	Restores document library objects from save files received on backup nodes. To invoke this routine when a save file is received on a backup node, rename this program to SAVFEXIT and compile it in the product library on the backup node.
SAVFEXIT02	Restores document library objects and Integrated File System objects from save files received on backup nodes. To invoke this routine when a save file is received on a backup node, rename this program to SAVFEXIT and compile it in the product library on the backup node.
SAVFEXIT03	Restores generic Integrated File System objects from save files received on backup nodes. To invoke this routine when a save file is received in a backup node, rename this program to SAVFEXIT and compile it into the product library on the backup node.

Replicating objects with save files and the SAVFEXIT program

Set up a save file for objects that cannot be replicated directly with iCluster, then replicate them by transferring the save file to the backup node.

1. Determine which objects on the primary node will be refreshed or mirrored.
These objects can be of any type that can be saved in a save file.
2. Create a save file that the objects will be saved to (use the IBM i command **CRTSAVF**).
3. Create a CL program that will save these objects into the created save file.

Note: A CL program can be written to use the autostart job or delay job capabilities to schedule when objects should be saved. Two sample CL programs (**SAVE_DLO** and **SAVE_IFS**) are provided that can be used to save document library objects and Integrated File System objects into the save file.

4. Write the **SAVFEXIT** program to apply user actions to replicated save files on backup nodes.
This program must reside in the iCluster product library on the backup node. Use the IBM i **CRTCLPGM** command to create the **SAVFEXIT** program.

The **SAVFEXIT01**, **SAVFEXIT02**, and **SAVFEXIT03** sample programs can be renamed to **SAVFEXIT**. These sample programs restore document library objects and Integrated File System objects on backup nodes.

5. Define the backup node that will be destination of the save files.
6. Create a replication group that includes the backup node.
7. Create an object specifier that references the save files and select the specifier to the group.
8. Write the **SAVFEXIT** program to apply user actions to replicated save files on backup nodes to make sure that the program is created on both nodes of your group.
9. Call the CL program (for example, **SAVE_DLO** or **SAVE_IFS**) that saves the objects into the save file you created.
10. Start the node and group operations.
When a save file is received on a backup node, **SAVFEXIT** is called. Typically, this program will restore the objects in the save file.

Replicating SQL objects

Replication is supported for SQL objects, including SQL functions (*SQLFUNC), SQL global variables (*SQLGVAR), SQL procedures (*SQLPROC), and SQL sequences (*SQLSEQ). iCluster supports SQL object names up to 128 characters in length for all SQL object commands.

To replicate SQL objects on an IASP, create the library of SQL objects on the IASP device first.

Note:

- The commands for native objects **DMSEL OBJ**, **DMDSEL OBJ**, **DMCHGOBJJSL**, and **DMACTOBJ** no longer support OBJTYPE for *SQLPROC and *SQLFUNC, nor do they support OBJTYPE *SQLGVAR and *SQLSEQ.
- The **DMWRKOBJ** command does not show SQL object specifiers of type *SQLFUNC and *SQLPROC in the list, nor does it show SQL objects of type *SQLGVAR and *SQLSEQ. The command to work with SQL objects is **DMWRKSQLO** (Work with SQL Object Specifiers).



- SQL object specifiers that were selected with **DMSELOBJ** on 7.1, 7.1 TR1, and 7.1 TR2 can be viewed with the **DMWRKSQLO** (Work with SQL Object Specifiers) command.

SQL functions (*SQLFUNC) include the user-defined functions created by CREATE FUNCTION SQL statements. SQL global variables (*SQLGVAR) include the global variables created by CREATE VARIABLE SQL statements. SQL procedures include the procedures created by CREATE PROCEDURE SQL statements. SQL sequences include the sequences created by CREATE SEQUENCE SQL statement.

SQL objects are not found by the Work with Objects (**WRKOBJ**) or the Work with Objects Using PDM (**WRKOBJPDM**) IBM i commands.

To find the SQL objects in a library, enter one of the following SQL statements:

SQL object type	SQL statement
SQL functions	SELECT * FROM QSYS2/SYSPFUNCS WHERE SPECIFIC_SCHEMA = '<library_name>'
SQL procedures	SELECT * FROM QSYS2/SYSPROCS WHERE SPECIFIC_SCHEMA = '<library_name>'
SQL global variables	SELECT * FROM QSYS2/SYSVARS WHERE SYSTEM_VAR_SCHEMA = '<library_name>'
SQL sequences	SELECT * FROM QSYS2/SYSSEQ WHERE SEQUENCE_SCHEMA = '<library_name>'

Specify **OBJTYPE** of *SQLFUNC, *SQLGVAR, *SQLPROC, or *SQLSEQ with the following commands:

- **DMSELSQLO** (Select SQL Objects)
- **DMDSELSQLO** (Deselect SQL Objects from Group)
- **DMCHGSQLSL** (Change SQL Object Selection)
- **DMACTSQLO** (Activate SQL Object)

The following requirements apply to replication of SQL functions, SQL global variables, SQL procedures, and SQL sequences:

- The system catalog table **QSYS2/SYSROUTINE** must be journaled to journal **QSYS2/QSQJRN** with **IMAGES(*BOTH)** for SQL function and SQL procedure replication. The system catalog table **QSYS2/SYSVARS** must be journaled to journal **QSYS2/QSQJRN** with **IMAGES(*BOTH)** for SQL global variable replication. The system catalog table **QSYS2/SYSSEQOBJ** must be journaled to journal **QSYS2/QSQJRN** with **IMAGES(*BOTH)** for SQL sequence replication.
 - When replicating SQL objects on non-IASP devices, first run this command on the primary node:

```
SETASPGRP *NONE
```

Then check the journal information for **QSYS2/SYSROUTINE**, **QSYS2/SYSVARS**, and **QSYS2/SYSSEQOBJ**.

- When replicating SQL objects on IASP devices, first run this command on the primary node:

```
SETASPGRP <group_primary_iasp>
```

Then check the journal information for **QSYS2/SYSROUTINE**, **QSYS2/SYSVARS**, and **QSYS2/SYSSEQOBJ**.

- The library on the backup node must be the same as the library on the primary node.
 - If you run a **DMSELSQLO** (Select SQL Objects) command to select SQL objects to a group, or **DMCHGSQLSL** (Change SQL Object Selection) command to change SQL object selection in a group, make

sure the group level **TGTLIB** parameter is either *PRIMARY, or the same as the Library in the **OBJ** parameter of the command.

- Replication is supported only for the following procedures and functions:
 - CREATE PROCEDURE
 - CREATE FUNCTION
 - CREATE SEQUENCE
 - CREATE VARIABLE
 - CREATE OR REPLACE PROCEDURE
 - CREATE OR REPLACE FUNCTION
 - CREATE OR REPLACE SEQUENCE
 - CREATE OR REPLACE VARIABLE
 - DROP PROCEDURE
 - DROP FUNCTION
 - DROP SEQUENCE
 - DROP VARIABLE
- You can manually refresh SQL objects with the DMACTSQLO (Activate SQL Object) command.

Related information

- [DMCHGSQLSL \(Change SQL Object Selection command\)](#)
- [DMDSELSQLO \(Deselect SQL Objects from Group command\)](#)
- [DMSELSQLO \(Select SQL Objects command\)](#)
- [DMACTSQLO \(Activate SQL Object command\)](#)

Replicating QDLS or DLO objects

Add QDLS objects (DLO objects) to an existing replication group to replicate them.

Including the QDLS folder and QDLS sub-folders in replication

To replicate the QDLS folder or any combination of its sub-folders, add each folder and sub-folder, and then exclude specific sub-folders.

To exclude QDLS folders from replication, replicate the entire QDLS folder and then specify which folders to exclude. Exclude only folders from replication and not the individual objects in the folders.

When you start the group, the QDLS file system folders are replicated, except the sub-folders that are specified for exclusion. The replication process creates the folders that you excluded on the backup node, but these folders will be empty.

1. To include QDLS sub-folders in replication, enter the following command:
`DMADDBSF GROUP(<group name>) PATH('/QDLS/*')`
2. To exclude sub-folders, enter the following command for each sub-folder:
`DMADDBSF GROUP(<group name>) PATH('/QDLS/<folder name>/*') INCFLG(*EXCLUDE)`

Related information

- [DMADDBSF \(Add Path Specifier to Group command\)](#)

Replicating WebSphere MQ objects

IBM WebSphere MQ is an application-to-application program that exchanges data contained in messages by way of queues. WebSphere MQ facilitates the exchange of information between applications that would not otherwise communicate with each other, such as to multiple and potentially remote systems in different geographical regions, and dissimilar systems. It provides assured, once-only delivery of messages.

iCluster and WebSphere MQ

iCluster has the ability to mirror WebSphere MQ objects, enabling WebSphere MQ to continue operating on a secondary system should a failover occur. This support allows interdependent applications to continuously communicate within enterprises and between enterprises. If you use it to integrate your applications, you can use iCluster to protect your applications from primary system outages since WebSphere MQ integration jobs are preserved.

Note: iCluster mirrors BSF objects that represent queues, processes, name lists, and so on. Remote journaling is used to guarantee that all WebSphere MQ transactions are replicated to the backup system and applied to the backup system queues.

iCluster supports WebSphere MQ Version 5 Release 2 and later.

Roleswitch is supported for WebSphere MQ groups with the DMCHGROLE (iCluster Change Primary Node) and DMSTRSWO (Switchover Group) commands.

DMSTRGRP (iCluster Start Group) starts replication for the queue manager.

Multiple queue managers are supported in iCluster. Use the DMADDGRP (Add Group) command to add a group for each of the queue managers to be mirrored.

The WebSphere MQ environment is automatically rebuilt on the new primary node during the roleswitch of an MQ Series group.

iCluster limitations

Authority changes made by using the IBM i Grant MQ Object Authority (**GRTMQMAUT**) and Revoke MQ Object Authority (**RVKMQMAUT**) commands are not journaled by WebSphere MQ. Therefore, iCluster does not mirror these changes.

WebSphere MQ support: Prerequisites

Before you can mirror WebSphere MQ objects with iCluster, perform the following steps:

- Verify that you have applied the prerequisite PTFs for the supported versions of WebSphere MQ.
- Create a local relational database directory entry for each node in the cluster.
- Add the QMQMADM administrative group to the DMCLUSTER user profile as either the group profile or as a supplemental group. This must be done on both the primary and backup nodes.
- Make sure that you have completed any prerequisite steps.
- Complete the steps described in "Enabling WebSphere MQ replication".

Additional prerequisites for WebSphere MQ support V5R2 and later

- The WebSphere MQ queue manager is active on the primary node and the WebSphere MQ queue manager on the backup node exists, but is inactive before starting replication. Use the **WRKMQM** command to check whether a

certain WebSphere MQ queue manager is active or not. An error message indicates when the manager is not active.

- Check the journal AMQAJRN on both the primary and backup machines (with the command WRKJRNA). Make sure the currently attached journal receiver has the highest number on the journal receiver chain.

Related information

- [DMSTRSWO \(Switchover Group command\)](#)
- [DMADDGRP \(Add Group command\)](#)
- [DMSTRGRP \(Start Group command\)](#)
- [DMCHGROLE \(Change Primary command\)](#)

Creating a local relational database directory entry

Each node in the cluster must contain a valid local relational database directory entry. Verify that these entries are unique across the cluster.

To determine if a node has a local database directory entry, issue the WRKRDBDIR IBM i command on each node in the cluster.

The local relational database directory entries are displayed as *LOCAL under the **Remote Location** field. When a *LOCAL value is not present, use the following command to create a local relational database directory entry for each node in the cluster:

```
ADDRDBDIR RDB(<chosen name>) RMTLOCNAME(*LOCAL)
```

where <chosen name> can be the system name that is typically used for identifying the machine.

 **Note:** If two machines have the same local relational database directory value, remove one of them and recreate it with a value that closely reflects the system name.

Enabling WebSphere MQ replication

Use the DMADDGRP (Add Group) and DMSTRGRP (iCluster Start Group) commands to enable WebSphere MQ replication.

1. Use the DMADDGRP (Add Group) command to set up a group for WebSphere MQ replication. The group type is *MQSERIES. The applicable object specifiers are added automatically and selected to this group by this command.
2. Use the DMSTRGRP (iCluster Start Group) command to start replicating WebSphere MQ objects.

Related information

- [DMADDGRP \(Add Group command\)](#)
- [DMSTRGRP \(Start Group command\)](#)

WebSphere MQ recommendations

Authority changes made through the GRTMQAUT and RVKMQAUT commands are not recorded by WebSphere MQ and are not mirrored by iCluster.

Use a CL program to apply the two commands to any WebSphere MQ objects on the primary node instead of issuing them at the command line. This means the operator might have to update and run the CL program periodically to

handle newly created WebSphere MQ objects.

iCluster can replicate this program to the backup node, where it can be run after a switchover. This ensures all authority changes on the primary node are properly applied to the backup node.

Supported object types for replication

Review the object types that are supported for replication.

Object Type	Object Type	Explanation
Alert table	*ALRTBL	
Authority holder	*AUTHLR	Authority holders are not referenced in an object specifier that uses the generic value of *ALL for the object type. Authority holders are replicated only for objects of type *FILE.
Block special file	*BLKSF	
Bind directory	*BNDDIR	
Chart format	*CHTFMT	
C locale description	*CLD	
Class	*CLS	
Command definition	*CMD	
Connection list	*CNNL	
Class-of-service description	*COSD	
Change request descriptor	*CRQD	
Communications side information	*CSI	
Cross-system product map	*CSPMAP	
Cross-system product table	*CSPTBL	
Controller description	*CTLD	
Device description	*DEVD	
Directory	*DIR	
Document Library Object	*DOC	
Data area	*DTAARA	
Data queue	*DTAQ	
Edit description	*EDTD	
Exit registration	*EXITRG	
Forms control table	*FCT	
First in - first out	*FIFO	

Object Type	Object Type	Explanation
File	*FILE	See the "Replicating Database *FILE Objects" topic for more information on considerations for replicating *FILE objects.
Folder	*FLR	
Font resource	*FNTRSC	
Forms definition	*FORMDF	
Font table resource	*FTR	
Graphics symbol set	*GSS	
Double-byte character set dictionary	*IGCDCT	
Double-byte character set font table	*IGCSRT	
Double-byte character set sort table	*IGCTBL	
Image catalog	*IMGCLG	
Internet packet exchange description	*IPXD	
Job description	*JOBD	
Job queue	*JOBQ	Job queues (*JOBQ) and message queues (*MSGQ) are replicated, but not changes to their contents.
Job scheduler	*JOBSCD	
Library	*LIB	Replicating library objects (*LIB) does not automatically replicate objects that are contained in the library. Only the library object is sent to the backup node. To replicate all objects in a library, use the predefined value *ALL to define an object specifier that addresses all objects and all object types in a specific library or a generic library.
Line description	*LIND	
Locale	*LOCALE	Locale object for localization.
Machine	*M36	
Advanced 36 machine configuration	*M36CFG	
Media definition	*MEDDFN	
Menu	*MENU	
Management collection	*MGTCOL	
Mode description	*MODD	
Module	*MODULE	
Message file	*MSGF	

Object Type	Object Type	Explanation
Message queue	*MSGQ	Job queues (*JOBQ) and message queues (*MSGQ) are replicated, but not changes to their contents.
Node group	*NODGRP	
Node list	*NODL	
NetBIOS description	*NTBD	
Network interface description	*NWID	
Network server description	*NWSD	
Output queue	*OUTQ	Spooled files in an *OUTQ can be replicated by setting the "Replicate *OUTQ contents" system value to *YES.
Overlay	*OVL	
Page definition	*PAGDFN	
Page segment	*PAGSEG	
PDF map	*PDFMAP	
Printer description group	*PDG	
Program	*PGM	
Panel group	*PNLGRP	
Product availability	*PRDAVL	
Print Services Facility Configuration	*PSFCFG	
Query form	*QMFORM	
Query manager query	*QMQRY	
Query definition	*QRYDFN	
Reference code translation table	*RCT	
System/36 machine description	*S36	
Subsystem description	*SBSD	
Search index	*SCHIDX	
Spelling help dictionary	*SPADCT	
SQL function	*SQLFUNC	SQL functions are created by the CREATE FUNCTION SQL statement.
SQL global variable	*SQLGVAR	SQL global variables are created by the CREATE VARIABLE SQL statement.
SQL package	*SQLPKG	
SQL procedure	*SQLPROC	SQL procedures are created by the CREATE PROCEDURE SQL statement, and include DB2 stored procedures.
SQL sequence	*SQLSEQ	SQL sequences are created by the CREATE SEQUENCE SQL

Object Type	Object Type	Explanation
		statement.
User defined SQL data type	*SQLUDT	
Service program	*SRVPGM	
Session description	*SSND	
Byte Stream File />	*STMF	
Symbolic links	*SYMLNK	
Table	*TBL	
Time zone	*TIMZON	Time zone object for localization
User index	*USRIDX	
User profile	*USRPRF	System-supplied user profiles cannot be replicated.
User queue	*USRQ	
User space	*USRSPC	<p>For user spaces, both object-level only change replication and object and content-level change replication are supported.</p> <p>Object-level only support will replicate creation, deletion, restore, and move/rename operations on user spaces. This is suitable for user spaces that are created and filled with data, but the data never changes.</p> <p>Content-level change support for user spaces is based on polling the objects at regular intervals to determine if they have changed. Note that since replication is based on polling, the Full Cluster Status Monitor cannot be relied upon to determine if the user spaces on the backup node are up-to-date.</p> <p>Since changes to user spaces (*USRSPC) are not journaled, iCluster uses polling to keep track of content changes for this type of object.</p>
Validation list	*VLDL	
Workstation customization	*WSCST	

User actions that can be applied to save file objects are replicated.

SQL functions, SQL procedures, SQL global variables, and SQL sequences are replicated

Replication is not supported for the following object types:

- System-provided authorization lists (*AUTL)
- IBM i operating system values
- Spooled files that contain Intelligent Printer Data Stream (IPDS) data
- *LIB, *USRPRF, and configuration objects in a local loopback configuration
- Journal objects (*JRN)

Related information

[Replicating SQL objects](#)

[Replicating database *FILE objects](#)

[Replicating user profiles](#)

[Using the SAVFEXIT user exit program when replicating save files](#)

Configuring high availability for external applications

Automatically configure replication groups to enable high availability in supported external application environments. Automatic configuration of high availability replication is supported for Rocket Lifecycle Manager for IBM i, Rocket LegaSuite for Application Modernization, and Help Systems products such as Robot/Schedule, Robot/Console, and others.

Configuring replication for external applications

Use the CFGHAXAPP (Configure HA for External Application) command to automatically configure replication groups to enable high availability replication in supported external application environments.

Planning and implementing high-availability replication for external applications typically requires knowledge about the external applications to determine the objects and libraries to replicate. The replication group configuration process is automated only for supported external applications.

Terminology

Review the following terms, which provide context for high availability as relevant to external applications:

- A node is a member of a cluster.
- Nodes are servers or logical partitions (LPAR) in the iCluster high availability solution.
- A cluster is a collection of nodes. A cluster for high availability must include two or more nodes: a primary node and at least one backup node.
- The server or LPAR where the external application is installed is the primary node.
- The server or LPAR where the external application is replicated is the backup node.

Related information

[Configuring replication for Rocket Lifecycle Manager](#)

[CFGHAXAPP \(Configure HA for External Application command\)](#)

Using iCluster with external applications

Create replication groups and enable high availability for the supported external applications.

Before you begin: Ensure that your environment meets the following requirements:

- The supported external application is installed on the same physical server or LPAR as iCluster.
 - The external application is installed and licensed on the primary node and the backup node.
 - The external application has the appropriate authorization keys activated on the backup node.
 - User profiles that are authorized for the external application must exist on the primary node and the backup nodes.
1. Log on to an active node as an authorized iCluster user with *ADMIN authority.
 2. Run the following commands to access iCluster:

```
CHGCLRLIB ICLUSTER  
GO DMCLUSTER
```

- Run the CFGHAXAPP (Configure HA for External Application) command:

```
CFGHAXAPP
```

- On the **Configure HA for Ext. Appl. (CFGHAXAPP)** screen, enter the values as appropriate for your environment and press Enter to run the command. Press F1 to view help on any field.
- Lifecycle Manager only: Follow the recommended procedure to end the external application subsystem that contains multiple jobs by running the following IBM i command:

```
ENDSBS ALDONDMI
```

 **Note:** There is no subsystem for Rocket LegaSuite.

- Run the following commands to perform initial synchronization between the nodes.

- If the external libraries or IFS directories that are managed by the external applications are small, run the DMSTRGRP (iCluster Start Group) command to perform initial synchronization of each new replication group:

```
DMSTRGRP GROUP(<group name>) REFRESH(*YES)
```

Where *<group name>* is a replication group that was added by the CFGHAXAPP (Configure HA for External Application) command.

- If the external libraries or IFS directories that are managed by the external application are large:
 - Run the DMMRKPOS (Mark Journal Positions) command:

```
DMMRKPOS GROUP(<group name>)
```

where *<group name>* is a replication group that was added by CFGHAXAPP (Configure HA for External Application) command.

- On the primary node, back up the libraries or directories that are managed by the external application to tape.
- Restore the tape to the backup node.
- Run the DMSTRGRP (iCluster Start Group) command to start replication:

```
DMSTRGRP GROUP(<group name>) USEMARKED(*YES)
```

where *<group name>* is a replication group that was added by CFGHAXAPP (Configure HA for External Application) command.

- From the backup node, run WRKHATMON (Work with Status Monitor on Backup Node) command and look for suspended or out-of-sync objects in the **Status** columns.
- Lifecycle Manager only: Restart the external application subsystem by running the following IBM i command on the primary node:

```
STRSBS ACMSCTL/ALDONDMI
```

- On the primary node, review the external application users that are defined as iCluster users by running the DMWRKUSR (Work with Users) command.

Note: The CFGHAXAPP (Configure HA for External Application) command adds external application users on the primary node only. Manually add the same users to the backup node.

- On the backup node, log on as an authorized iCluster user with *ADMIN authority.
- On the backup node, define the external application users as iCluster users by running the DMWRKUSR (Work with Users) command. Add the iCluster users that are shown on the **DMWRKUSR** screen in Step 9.

Related information

[Monitoring the replication environment](#)

[CFGHAXAPP \(Configure HA for External Application command\)](#)

Recreating groups after Lifecycle Manager upgrade from Version 7.5

Replication groups for Rocket Lifecycle Manager must be re-created after Lifecycle Manager upgrades from Version 7.5 to later versions.

When the Rocket Lifecycle Manager environment is upgraded from Version 7.5, remove and then recreate the replication groups. Upgrades from Version 7.6 or later do not require the replication groups for Lifecycle Manager to be recreated.

- Log on to an active node as an authorized iCluster user with *ADMIN authority.
- Run the following commands to access iCluster:

```
CHGCURLIB ICLUSTER  
GO DMCLUSTER
```

- To remove the Lifecycle Manager replication group or groups, run the RMVHAXAPP (Remove HA for External Application) command:

```
RMVHAXAPP
```

- To re-create the replication groups, run the CFGHAXAPP (Configure HA for External Application) command:

```
CFGHAXAPP
```

- On the **Configure HA for Ext. Appl. (CFGHAXAPP)** screen, enter the values as applicable for your environment and press Enter to run the command. Press F1 to view help on any field.
- Follow the recommended procedure to end the external application subsystem that contains multiple jobs by running the following IBM i command on the primary node:

```
ENDSBS ALDONDML
```

- Run the following commands to perform initial synchronization between the nodes.
 - If the external libraries that are managed by the external application are small, enter the DMSTRGRP (iCluster Start Group) command to perform initial synchronization of each new replication group:

```
DMSTRGRP GROUP(<group name>) REFRESH(*YES)
```

Where *<group name>* is a replication group that was added by the CFGHAXAPP (Configure HA for External Application) command.

- If the external libraries that are managed by the external application are large:
 - a. Run the DMMRKPOS (Mark Journal Positions) command:

```
DMMRKPOS GROUP(<group name>)
```

where *<group name>* is a replication group that was added by CFGHAXAPP (Configure HA for External Application) command.

- b. On the primary node, back up the libraries that are managed by the external application to tape.
- c. Restore the tape to the backup node.
- d. Run the DMSTRGRP (iCluster Start Group) command to start replication:

```
DMSTRGRP GROUP(<group name>) USEMARKED(*YES)
```

where *<group name>* is a replication group that was added by CFGHAXAPP (Configure HA for External Application) command.

8. From the backup node, run WRKHATMON (Work with Status Monitor on Backup Node) command and look for suspended or out-of-sync objects in the **Status** columns.
9. Restart the external application subsystem by running the following IBM i command on the primary node:

```
STRSBS ACMSCTL/ALDONDML
```

10. On the primary node, review the external application users that are defined as iCluster users by running the DMWRKUSR (Work with Users) command.

Note: The CFGHAXAPP (Configure HA for External Application) command adds external application users on the primary node only. You must manually add the same users to the backup node.

11. On the backup node, log on as an authorized iCluster user with *ADMIN authority.
12. On the backup node, define the external application users as iCluster users by running the DMWRKUSR (Work with Users) command. Add the iCluster users that are shown on the **DMWRKUSR** screen in Step 9.

Related information

[CFGHAXAPP \(Configure HA for External Application command\)](#)

[RMVHAXAPP \(Remove HA for External Application command\)](#)

[DMSTRGRP \(Start Group command\)](#)

[DMMRKPOS \(Mark Journal Positions command\)](#)

Configuring replication for Rocket Lifecycle Manager

After you configure replication groups for external applications, review the additional steps that are required for Lifecycle Manager.

Configuring replication of objects stored outside of Lifecycle Manager

User exit programs and reference libraries that are used by Lifecycle Manager might be stored in libraries outside of Lifecycle Manager and are not automatically included in replication groups.

Before you begin: To replicate objects that are stored outside of Lifecycle Manager, use the DMSELOBJ (Select Objects to Group) command to add libraries and object specifiers.

Run the DMSELOBJ (Select Objects to Group) command to include the objects in the replication groups.

For example, run the following command to select all objects of all types in the MYEXTLIB1 library and select the objects to the active ALDON group and begin replication of journal entries:

```
DMSELOBJ GROUP(ALDON) OBJ(MYEXTLIB1/*ALL) OBJTYPE(*ALL) NEWOBJACT(*REFRESH)
```

Related information

[DMSELOBJ \(Select Objects to Group command\)](#)

Staging objects

Staging is a storage mechanism that holds journal entries or transactions before they are applied to the objects on the backup node. This storage mechanism on the backup node allows the apply jobs to be ended while backup and other maintenance operations that require exclusive access to files and objects are performed. Cluster operations for the group are allowed to continue, but changes are not applied on the backup node until the apply jobs are started again.

Ending the apply process for a backup node does not affect processing on the node that is replicating objects until all available staging storage has been filled. The amount of space available for the staging store is set for each node with the staging store size parameter on the DMADDNODE (Add Node) and DMCHGNODE—Change Node commands.

If cluster operations for the group are active when the apply process is stopped, the backup node continues to receive journal entries which are placed into the staging store. If the staging store becomes full, then journal scraping stalls.

The apply process on a backup node can be stopped or started independently of any cluster operation. If the group is not currently active, the apply process ends after the staging store is drained.

Staging store and allocation

The staging store is a nonvolatile storage mechanism for backup nodes that is managed on a node-by-node basis. Set the size of the staging store for each backup node in the cluster. You can also set group-level staging stores. The maximum size of each group-level staging store is the same as the node-level staging store.

All information that is transferred to backup nodes is put into the staging store and, assuming the apply process is active, is applied. If the apply process is not active, the staging store stores the journal entries until the apply process is started.

You can increase the maximum size of the staging store at any time. The extra space that is allocated is used if needed while active. A decrease in the size of the store is made only if the total size of transactions in the store is less than the maximum size of the store and all iCluster replication is ended on the node.

If necessary, you can also choose to force drain the staging store when the apply process is started. Normally, the apply process merges entries from the audit and database channels and applies them in sequence. When one channel becomes empty, the apply process stops regardless of any entries remaining in the other channel. When the staging store is force drained, then the merge stops after the last entry of the channel with fewer entries is reached, and the apply process drains and applies the entries in the other channel until it is empty.

Considerations

Staging increases system resource requirements because journal data is placed into and extracted from the staging store. The system storage requirements might need to be increased to hold the staged journal entries. There is no impact on performance or resource requirements on primary nodes.

The scrape latency is not affected while the apply process is suspended as long as the staging store is large enough to hold all the journal information. Stalling might occur if the staging store is not large enough to hold the journal entries. For example, transferring large objects can cause communications with other nodes, or replication for other groups, to appear stalled as communication buffers back up.

LOB staging store

LOB data is stored in a directory in IFS which is separate from the regular staging store.

Calculating the minimum staging store size

The minimum size of the staging store is directly related to the number of database journals on a node. A typical staging store size is 10 GB.

To calculate the minimum disk space allocation for a staging store, use the following formula:

```
Minimum Size = (2 x <database journals> + 1) * 16 MB
```

In this calculation, *<database journals>* is the number of database journals on the node. For example, if six database journals are on a node, then the node requires a minimum of 208 MB of staging store space.

Related information

[Replicating LOBs](#)

[Changing node attributes](#)

[DMADDNODE \(Add Node command\)](#)

[DMCHGNODE \(Change Node command\)](#)

Clearing the staging store

In certain situations, the portion of the staging store reserved for a group is cleared.

The portion of the staging store that is reserved for a group is cleared, regardless of whether those entries have been applied on the backup node, in the following instances:

- When you start replication with the **Use marked journal positions** parameter set to *YES.
- When you start replication after issuing the DMSETPOS (Set Journal Start Position) command.
- When you start replication with a refresh before mirroring (**Refresh before mirror** parameter is set to *YES).

Related information

[DMSETPOS \(Set Journal Start Position command\)](#)

Changing the size of a staging store

Use the DMCHGNODE (Change Node) command to change the size of a staging store.

1. Enter the following command:
DMCHGNODE
2. In the **Node** field, type the name of the node.
3. In the **Staging store size (in MB)** field, type the size of the staging store.
Specify a value from 512 - 1048576.

Related information

[DMCHGNODE \(Change Node command\)](#)

Changing the name of a staging store library

Use the DMENDGRP (iCluster End Group) and DMCHGGRP (Change Group) commands to change the name of a staging store library for a group.

Before you begin: End the group before you change the name of the staging store library for a group.

1. Enter the following command:
DMENDGRP
2. Enter the following command:
DMCHGGRP
3. Specify the staging store library in the **Staging store library** field:
 - *NODE—Set the staging store library for the group to the same library that is set for the node.
 - <name>—Set the staging store library to a unique staging store library name. It is recommended to specify a library that does not already exist on your server. The library is created if it does not already exist. The maximum size of the staging store is determined by the staging store size that is set for the node.
4. Use one of the following options to restart the group:
 - Refresh.
 - Set **Use marked positions** *YES option after DMMRKPOS.
 - Run the DMSETPOS *LASTAPY command for all journals for the group and then restart the group.

Related information

- [DMENDGRP \(End Group command\)](#)
[DMCHGGRP \(Change Group command\)](#)
[DMSETPOS \(Set Journal Start Position command\)](#)

Starting the replication apply process

Use the DMSTRAPY (Start Replication Apply Process) command to start the replication apply process for all groups or a specific group.

1. Enter the following command:
DMSTRAPY
2. In the **Group or resilient application** field, type one of the following values:
 - *ALL—Starts the replication apply process for all groups.
 - <name>—Specify a group name.
3. In the **Override current stop time** field, specify whether to override a previously set stop date/time for apply processes, which is specified through the DMENDAPY command.
Choose one of the following values:
 - *YES—Starts the apply process and overrides the previous stop date/time value. By default, this value is used.
 - *NO—Starts the apply process, but preserves the previous stop date/time value.
4. In the **Drain staging store** field, specify if the staging store is drained.
Choose one of the following values:
 - *YES—Drains the staging store.
 - *NO—Does not drain the staging store. By default, this value is used.

Related information

- [DMSTRAPY \(Start Replication Apply Process command\)](#)
[DMENDAPY \(End Replication Apply Process command\)](#)

Starting the replication apply process for groups on the backup node

Use the DMSTRAPY (Start Replication Apply Process) command to start the apply processes on a backup node.

1. Enter the following command:
DMSTRAPY
2. In the **Group or resilient application** field, enter the following command:
*BACKUP
3. In the **Backup node** field, specify the name of the backup node.
Choose one of the following values:
 - <name>—Specifies the backup node name.
 - *ALL—Specifies all groups on all backup nodes in the cluster. By default, this value is used.
4. In the **Override current stop time** field, specify if a current set stop date/time for apply processes is overridden. The stop date/time is specified through the DMENDAPY command.
Choose one of the following values:
 - *YES—Starts the apply process and overrides the previous stop date/time value. By default, this value is used.
 - *NO—Starts the apply process, but preserves the previous stop date/time value.
5. In the **Drain staging store** field, specify if the staging store drains.
Choose one of the following values:
 - *YES—Drains the staging store.
 - *NO—Does not drain the staging store. By default, this value is used.

Related information

[DMSTRAPY \(Start Replication Apply Process command\)](#)

[DMENDAPY \(End Replication Apply Process command\)](#)

Ending the replication apply process

Use the DMENDAPY (End Replication Apply Process) command to end the replication apply process for all groups or for a specific group.

1. Enter the following command: DMENDAPY
2. In the **Group or resilient application** field, type one of the following values:
 - *ALL—Includes all groups in the cluster.
 - <name>—Specify a group name.
3. In the **End option** field, specify how to stop the apply processes.
Choose one of the following values:
 - *CTRLD—Specifies a controlled end of the processes, allowing them to complete their current processing.
 - *INVLID—Invalidates a previously-entered pending end date/time for the apply processes; applies will continue.
 - *DATETIME—Specifies the date and time when the processes will end
 - *IMMED—Stops the processes immediately.
4. If *DATETIME is selected, set the date value in the **ENDDATE** field.
5. If *DATETIME is selected, set the time value in the **ENDTIME** field.

Related information

[DMENDAPY \(End Replication Apply Process command\)](#)

Ending the replication apply process on a backup node

Use the DMENDAPY (End Replication Apply Process) command to end the apply processes on a backup node.

1. Enter the following command:
DMENDAPY
2. In the **Group or resilient application** field, enter the following: .
*BACKUP
3. In the **Backup node** field, specify the name of the backup node.
Choose one of the following values:
 - <name>—Specifies the backup node name.
 - *ALL—Specifies all backup nodes in the cluster. By default, this value is used.
4. Specify to stop the apply processes in the **End option** field.
Choose one of the following values:
 - *CTRLD—Specifies a controlled end of the processes, allowing them to complete their current processing.
 - *INVLID—Invalidates a previously-entered pending end date/time for the apply processes; applies will continue.
 - *DATETIME—Specifies the date and time when the processes will end
 - *IMMED—Stops the processes immediately.
5. If *DATETIME is selected, set the date value in the **ENDDATE** field.
6. If *DATETIME is selected, set the time value in the **ENDTIME** field.

Related information

[DMENDAPY \(End Replication Apply Process command\)](#)

Locked objects and locked object contention

Operations sometimes lock objects to ensure that only one process at a time can affect the object. To configure object processing use the DMSETSVAL (Set Cluster System Values) command or the DMCHGGRP (Change Group) command.

Locked object checks

If the replication apply processes on the backup node cannot apply a journal entry to a file because it is exclusively locked, configure the apply processes on the backup node to wait a certain amount of time and retry the apply process a specified number of times. Configure locked object checks with the following system or group values:

- **Number of locked object checks**
- **Delay between lock checks**

Avoiding locked object contention

Specify a delay for replication processes to wait on the primary node before processing create, rename, move, or activate journal entries for journaled objects. Use the **Delay for object processing** system and group value to specify the required delay. A delay of this type might be required in environments where applications briefly retain a lock on

objects that they create, rename, or move. Such a lock can interfere with replication processing and cause replication errors.

This delay value is also used when the product processes a T ZC (change) journal entry for a user space object (*USRSPC) before refreshing the user space.

Setting retry limits for saving objects

Specify the number of attempts to save a locked object by using the **Number of save operation retries** system or group value.

Related information

[DMSETVAL \(Set Cluster System Values\) command](#)

[DMCHGGRP \(Change Group command\)](#)

Commitment control

Commitment control offers staging for database transactions before being applied.

Commitment control stages database transactions so that they are assembled before being applied, or opened in a commitment control environment to ensure that only complete transactions are applied. It also makes sure that the changes are applied in the correct sequence. Commitment control staging is performed on backup nodes. Specify commitment control at the replication group and system levels.

The following commitment control levels are supported:

- *NONE—No commitment control. The update processes on the backup node do not perform commitment control staging, do not open the files under commitment control, and do not apply updates under commitment control.
- *LEVEL1—All updates that comprise a transaction are assembled before they are applied on the backup node. *LEVEL1 does not require that the backup files are journaled, it does not open the files under commitment control, and does not apply the updates under commitment control. If the backup node ends abnormally in the middle of applying a transaction, the transaction is partially applied and the transaction is completed on the next startup. Use this option when you do not want to journal the backup node files for performance reasons while keeping transaction consistency.
- *LEVEL2—All updates in a transaction are applied in a commitment control environment to ensure that only complete transactions are applied. *LEVEL2 requires that backup files are journaled with both images. Files are opened under commitment control and updates are applied under commitment control. If the backup node ends abnormally in the middle of applying a transaction, the system automatically rolls back the transaction, ensuring that the database is still consistent. This option provides true commitment control.



Note: A suspended file cannot be involved in transactions under commitment control. Updates to suspended files are not applied as part of a committed transaction.

Related information

[Specifying the commitment control level for a group](#)

Specifying the commitment control level for a group

Use the DMCHGGRP (Change Group) command to specify the level of commitment control to use for replication of *FILE objects.

-
1. Enter the following command:
DMCHGGRP
 2. In the **Group** field, type the name of the group.
 3. Press **PgDn**.
 4. In the **Commitment control level** field, specify the level of commitment control to use for *FILE object replication.
Choose one of the following values:
 - *SAME—Keeps the current setting for this parameter.
 - CLUSTER—Uses the system value that is assigned in the DMSETSVAL (Set Cluster System Values) command.
 - *NONE—Does not perform commitment control staging during the update process on the backup node.
 - *LEVEL1—Assembles all updates that comprise a transaction before they are applied on the backup node.
 - *LEVEL2—Applies all updates in a commitment control environment to ensure backup database consistency.

Related information

[DMCHGGRP \(Change Group command\)](#)

Cluster operations

Start and end cluster operations for groups, the current node, or a specified node.

To start and end iCluster processes for the current node, and for the groups that are configured to autostart, use the DMSTRCST (Start iCluster Processes) and DMENDCST (End iCluster Processes) commands.

To start cluster operations at a specified node, use the DMSTRNODE (iCluster Start Node) command.

To start cluster operations for a group, use the DMSTRGRP (iCluster Start Group) command.

Related information

- [DMSTRCST \(Start iCluster Processes command\)](#)
- [DMENDCST \(End iCluster Processes command\)](#)
- [DMSTRNODE \(Start Node command\)](#)
- [DMSTRGRP \(Start Group command\)](#)

Starting cluster operations at a node

When cluster operations are started at a node, the status of the node is set to *ACTIVE.

If the node is the backup node of a group that has an active status, cluster operations for the group are also started when the failover mechanism is Cluster Resource Services.

1. From the Main Menu, type **11** at the command line.
2. Press Enter.
3. In the **Node** field, type the name of the node where cluster operations start.
4. Press Enter.

Starting cluster operations for groups

When cluster operations are started for a group, the group status is set to *ACTIVE.

The group replication status is displayed in two columns of the Work with Groups screen, which is invoked with the DMWRKGRP command or option 2 from the main iCluster menu.

The first column that displays group replication status is titled "Repl. status". This column shows the combined status of a group's journal scrapers or refresh processes on the primary node and the receive processes on the backup node.

The second column that displays group replication status is titled "Apply status". This column shows the combined status of a group's apply processes on the backup node, that is, the processes that are applying the changes required to the replicated objects on the backup node. The apply processes are independent of the journal scrape and receive processes and can be in *INACTIVE status when the replication status for a group is *ACTIVE, and vice versa, when needed.

1. From the Main Menu, type **13** at the command line.
2. Press Enter.
3. In the **Group** field, specify how cluster operations start for a group or all groups.
Choose one of the following values:

- *ALL—Starts cluster operations for all groups on all nodes.
 - *PRIMNODE—Starts cluster operations for all groups with the primary node specified.
 - <name>—Starts cluster operations for the specified group.
4. Press Enter.
 5. If *PRIMNODE was selected, type the name of the node in the **Primary node** field.
 6. In the **Start replication apply jobs** field, specify whether the apply process for the group is started on the backup node when replication begins.
Choose one of the following values:
 - *NOCHG—Preserves the last operational status of the apply jobs on the backup node. By default, this value is used.
 - *YES—Starts the apply jobs on the backup node for the group.
 - *NO—Does not start the apply jobs on the backup node for the group.
 7. Press Enter.
 8. In the **Refresh selected objects** field, specify whether to perform an initial refresh of the selected objects in the group before replication is started.
Choose one of the following values:
 - *YES—Performs an initial refresh of selected objects in the group.
 - *NO—Does not perform an initial refresh of selected objects in the group. By default, this value is used.
 - *MQRUNSTR—Refreshes objects on the backup node for WebSphere MQ groups and is recommended for environments where WMQ transactions may be started while the group is inactive.
 9. To start replication at marked journal positions, select a value in the **Use marked journal positions** field.
Choose one of the following values:
 - *YES—Starts replication at marked journal positions.
 - *NO—Does not start replication at marked journal positions. By default, this value is used.
 10. To start the scrape process for a remote journal replication group when the group's primary node has failed, select a value in the **Force remote journal scrape** field.
Choose one of the following values:
 - *YES—Starts the scrape jobs on the backup node for the remote journal group.
 - *NO—Does not start the scrape jobs on the backup node for the remote journal group.
 11. Press Enter.

Ending cluster operations for groups with a specific primary node

End cluster operations for groups with a specific primary node immediately or in a controlled manner. A controlled stop completes tasks to ensure that replication ends gracefully. However, the completion of these tasks can take some time.

1. From the Main Menu, type **14** at the command line.
2. Press Enter.
3. In the **Group** field., type ***PRIMNODE**.
4. Press Enter.
5. In the **Primary node** field, type the name of the primary node.
6. In the **End option** field, enter one of the following values:
 - *IMMED—Ends replication immediately. Operations might not stop as expected.
 - CNTRLD—Ends replication in a controlled manner. Controlled stops are recommended when possible.
7. Press Enter.

Restarting cluster node operations with current machine rejoining the cluster

Use the DMREJOIN (iCluster Rejoin Cluster) command to specify the rejoin attempt parameters. You can also specify whether to start a new cluster on the current machine if attempts to rejoin an existing cluster fail.

1. Enter the following command:

DMREJOIN

2. In the **Number of rejoin attempts** field, specify the number of attempts to rejoin an existing cluster. The default value is 1.
3. In the **Delay secs. between attempts** fields, specify the time interval in seconds between attempts to restart cluster operations at the current node. The default value is 10.
4. In the **Start new cluster** field, specify whether to start a new cluster on the current machine if attempts to rejoin an existing cluster fail. Choose one of the following values:
 - *YES—Starts a new cluster on the current machine if attempts to rejoin an existing cluster fail.
 - *NO—Does not start a new cluster on the current machine if attempts to rejoin an existing cluster fail. By default, this value is used.

Related information

[DMREJOIN \(Rejoin Cluster command\)](#)

Ending cluster operations at a node

When cluster operations are ended at a node, the status of the node is set to *INACTIVE.

1. From the Main Menu, type **12** at the command line.
2. Press Enter.
3. In the **Node** field, type the name of the node where cluster operations end.
4. Press Enter.

Ending cluster operations for all groups

End cluster operations for all groups immediately or in a controlled manner. A controlled stop completes tasks to ensure that replication ends gracefully. However, the completion of these tasks can take some time.

1. From the Main Menu, type **14** at the command line.
2. Press Enter.
3. In the **Group** field, type ***ALL**.
4. Press Enter.
5. In the **End option** field, type one of the following values:
 - *IMMED—Ends replication immediately. Operations might not stop as expected.
 - *CNTRLRD—Ends replication in a controlled manner. Controlled stops are recommended when possible.
6. Press Enter.

Restarting iCluster after restarting a node

To automatically restart iCluster for a node, include the commands in the autostart job files for each node in the

cluster.

1. To initialize TCP/IP processing, enter the **STRTCP** command.
2. You must minimally start the *INETD server. To start the TCP/IP server, enter the following **STRTCPSVR** command:
STRTCPSVR SERVER(*INETD)
3. To start the XDMCLUSTER subsystem, enter the following command:
STRSBS SBSD(XDMCLUSTER)
4. To start the TCP/IP listener job on the local node, enter the **STRHATCP** (Start TCP/IP Listener) command.
5. To start the node, enter the **DMSTRNODE** (iCluster Start Node) command.

Related information

- [**STRHATCP \(Start TCP/IP Listener command\)**](#)
[**DMSTRNODE \(Start Node command\)**](#)

Upgrading the version of a cluster

Ensure that you upgrade the cluster version when Cluster Resource Services is the failover mechanism. You must upgrade the cluster to the highest available version to reduce the likelihood of version problems after future node upgrades.

Every node in a cluster has a potential node version, which is determined by the operating system version installed on the node.

Ensure that you upgrade the cluster version when Cluster Resource Services is the failover mechanism.

1. To view the version of the cluster, enter the following command: **DSPCLUINF**
DSPCLUINF
2. Upgrade the operating system for nodes in the cluster. The potential node version for each node must be greater than or equal to the version you want the cluster to have.
3. Enter the following command:
CHGCLUVER
4. In the **Cluster** field, type the name of the cluster that you want to change.

Deleting a cluster

To ensure that all cluster operations are stopped and all cluster definitions are deleted, issue the **DMDLTCLSTR** (Delete Cluster) command on each node in the cluster.

When this command is invoked on an inactive node, cluster operations are stopped and cluster definitions are deleted only on the node where the command is invoked.

1. Enter the following command:
DMDLTCLSTR
2. Specify the cluster to delete with the **CLSTR(<cluster name>)** parameter.
The default cluster name is *CURRENT.

Related information

- [**DMDLTCLSTR \(Delete Cluster command\)**](#)

Suspended objects

During normal replication, changes to replicated objects are applied on the backup node. However, objects might become suspended if one or more changes cannot be processed.

Potentially, the primary and backup nodes could become unsynchronized.

If a suspended object is part of a transaction under commitment control, the transaction is not applied on the backup system in its entirety.

An object can become suspended by one or more of the following possible causes:

- If the object cannot be refreshed or if a file or member level operation fails.
- If a file is larger than the cluster system value for the maximum size that can be refreshed through the network. Files that exceed the maximum size are not refreshed are marked as suspended.
- If you select a certain object to be suspended. User-suspended objects are displayed in the Full Cluster Status Monitor with the SBU (Suspended by User) reason code.
- If the object is locked by an application. Locked objects cannot be restored to the backup node.

In most cases, suspended objects can be automatically reactivated. To manually reactivate objects use the DMACTOBJ (Activate Object) command for native objects and the DMACTBSF (Activate BSF Object) command for IFS objects. All objects suspended with EJF (journaling ended for the object) or SBU (object suspended by user) can be activated for a group with the DMACTSUS (Activate Suspended Objects) command.

Related information

[DMACTOBJ \(Activate Object command\)](#)

[DMACTBSF \(Activate BSF Object command\)](#)

[DMACTSUS \(Activate Suspended Objects command\)](#)

Suspending an object

Suspend an object to get the object back in sync, or when the record format of a physical or logical file changes.

1. Enter the following command:
DMSUSOBJ
2. In the **Group** field., type the name of the replication group
3. In the **Object** field, type the object name.
4. In the **Library** field, type the name of the library where the object resides.
5. In the **Object type** field, specify the type of object in the **Object type** field.
6. To suspend a specific member of a physical file if you set *FILE as the object type, choose a value from the **Member** field.
Choose one of the following values:
 - <name>—Specifies a member name.
 - *ALL—Specifies all members of the physical file. By default, this value is used.

Related information

[DMSUSOBJ \(Suspend Object command\)](#)

Suspending a BSF object

Suspend a Byte Stream Files (BSF) object to get the object back in sync.

1. Enter the following command:
DMSUSBSF
2. In the **Group** field, type the name of the replication group.
3. In the **Path specifier** field, specify the complete path to identify the location of the BSF object to activate. Enclose the path name in single quotation marks, and start with the forward slash (/) character.

Related information

[DMSUSBSF \(Suspend BSF Object command\)](#)

Activating objects

Replication of suspended objects begins when you activate the objects. Objects can be refreshed as a part of the activation, or replication can begin upon activation.

Activate one or more suspended objects through the DMACTOBJ (Activate Object) command for native objects, DMACTBSF (Activate BSF Object) command for BSF objects, or from the Full Cluster Status Monitor. You cannot specify a particular time or journal entry to start replication for the file being activated.

If you do not refresh the objects when you activate them, then you must ensure that logical files that are associated with a suspended physical file are created on the backup node before you activate the physical file, which starts replication. If you do not refresh the object when you activate it, you must also ensure that no changes are made to the object between the time that the object was suspended and the time the object is activated. Changes that are made since the time of the suspension are not applied to the object on the backup node if the object is activated without a refresh.

Related information

[Automatic object reactivation](#)

[DMACTOBJ \(Activate Object command\)](#)

[DMACTBSF \(Activate BSF Object command\)](#)

Activating a suspended object

If objects become suspended during replication, activate these objects.

1. Enter the following command:
DMACTOBJ
2. In the **Group** field, type the name of the replication group.
3. In the **Object** field, specify the object name component.
Choose one of the following values:
 - <name>—Specifies the object name.
 - <generic*>—Specifies a generic object name to identify multiple objects in a library.
 - *ALL—Specifies all objects in the library.
4. In the **Library** field, type the name of the library where the object resides.
5. In the **Object type** field, specify an object type.
Choose one of the following values:
 - *ALL—Specifies all objects in the library that have the same object name component that is set in the

Object field but different types.

- <name>—Specifies the object type.

6. To activate a specific member of a physical file if you set *FILE as the object type, select a value from the **Member** field.

Choose one of the following values:

- *ALL—Specifies all members of the physical file. By default, this value is used.
- <name>—Specifies the member name.

7. In the **Refresh object** field, specify whether to refresh the object to the backup node when it is activated.

Choose one of the following values:

- *YES—Refreshes the object on the backup node when the object is activated. By default, this value is used.
- *NO—Does not refresh the object on the backup node when the object is activated. You are responsible for refreshing the object and ensuring that it is synchronized at the time the activation is performed.
- *FIXREC—Refreshes only physical file records that failed to replicate because of record-level errors that occurred during the backup apply process. Only files that are suspended with the RLE reason code and whose **Maximum record level errors** parameter evaluates to *FIX are activated.

8. In the **Refresh method** field, specify the physical file refresh method.

Choose one of the following values:

- *RBR—Record-by-record refresh method. By default, this value is used.
- *SAVRST—Save and restore refresh method. Saves the files with the data and restores the file on the backup node.

Note:

On IBM i 7.4, to enable refresh with the *RBR method for physical files with ALWDLT(*NO) and ALWUPD(*NO), do one of the following:

- Ensure that necessary PTFs for the supported versions are installed
- Run the following command on each node in the cluster:

```
CHGFCNUSG FCNID(QIBM_DB_SECADM) USER(DMCLUSTER) USAGE(*ALLOWED)
```

9. Press Enter.

Related information

[Supported object types for replication](#)

[DMACTOBJ \(Activate Object command\)](#)

Activating BSF objects

If Byte Stream Files (BSF) objects become suspended during replication, activate these objects.

1. Enter the following command:

DMACTBSF

2. In the **Group** field, type the name of the replication group.

3. In the **Path specifier** field, specify the complete path to identify the location of the BSF object to activate. Enclose the path name in single quotation marks, and start with a forward slash (/).

4. In the **Refresh object** field, specify whether to refresh the object to the backup node when the object is activated.

Choose one of the following values:

- *YES—Refreshes the object on the backup node when the object is activated. By default, this value is used.
- *NO—Does not refresh the object on the backup node when the object is activated. You are responsible for refreshing the object and ensuring that it is synchronized at the time the activation is performed.

Related information

[DMACTBSF \(Activate BSF Object command\)](#)

Automatic object reactivation

Automatic reactivation attempts to resynchronize suspended objects without user intervention.

If an error occurs during object replication, the object becomes unsynchronized and is suspended. After an object is suspended, subsequent updates to that object are not applied until the object is automatically reactivated or manually activated.

Automatic reactivation allows the automatic refresh of objects that become unsynchronized on the primary node or the backup node. Automatic reactivation provides the following benefits:

- Reduces the effort that is required to keep the primary and backup nodes synchronized.
- Decreases the time that nodes could be unsynchronized when an object is suspended.
- Reduces the time that is required to monitor replication status and activate suspended objects.

Automatic reactivation is configurable. You can enable automatic reactivation, or specify the number of reactivation attempts for an object and the maximum size of an object to refresh for reactivation.

To set the maximum size of object allowed to be auto-reactivated, use the **Maximum Reactivation Size** parameter in the DMSETVAL (Set Cluster System Values) command. This parameters is also configurable at the group and object specifier levels with the DMADDGRP (Add Group), DMCHGGRP (Change Group), DMSELOBJ (Select Objects to Group), and DMCHGOBJ (Change Object Specifier) commands.

Objects that are suspended with the following reason codes cannot be automatically reactivated. All of the other reason codes are eligible for automatic reactivation.

Native object reason codes:

EJF

The object was suspended on the primary node because journaling ended on the primary node for the object.

JPF

A logical file was suspended on the primary node because the associated physical file could not be journaled.

MRR

The file was suspended on the primary node. The file should be refreshed manually but it has yet to be activated.

NGP

A logical file was suspended on the primary node because the associated physical file was not replicated in the same group as the logical file.

RBC

The file was part of a canceled rollback operation on the primary node.

RGF

The file was suspended because a member could not be reorganized or cleared on the backup node.

SBU

An object was explicitly suspended on the primary node as a result of issuing the **DMSUSOBJ** command.

SIZ

An object was suspended on the primary node. A refresh of the object was required, but the size of the object was greater than the value specified in the **Maximum Refresh Size** system value.

SPF

A logical file was suspended on the primary node because the associated physical file was suspended.

IFS object reason codes:

EJF

The object was suspended on the primary node because journaling ended on the primary node for the object.

LNK

The BSF object is a hard link. Replication of hard links is not supported by iCluster.

SBU

An object was explicitly suspended on the primary node as a result of issuing the **DMSUSOBJ** command.

SIZ

An object was suspended on the primary node. A refresh of the object was required, but the size of the object was greater than the value specified in the **Maximum Refresh Size** system value.

Related information

[Activating objects](#)

[Work with Object Status views](#)

[DMSETSVAL \(Set Cluster System Values\) command](#)

[DMADDGRP \(Add Group command\)](#)

[DMCHGGRP \(Change Group command\)](#)

[DMSELOBJ \(Select Objects to Group command\)](#)

[DMCHGOBJSL \(Change Object Selection command\)](#)

Removing user profiles from suspension

Remove user profiles from suspension when the UID or GID values are not the same on the primary and backup nodes, and the **UID/GID mismatch suspension** parameter for the **DMSETSVAL** command is set to *YES.

1. Set the **UID/GID mismatch suspension** parameter for the **DMSETSVAL** command to *NO.
2. Modify all user profiles on the backup node to ensure no differences or conflicts occur with the UID and GID values of the profiles on the primary node.
3. Activate the suspended user profiles with the **DMACTOBJ** command, or use one of the activation options from the Full Cluster Status Monitor.

Related information

[DMSETSVAL \(Set Cluster System Values\) command](#)

[DMACTOBJ \(Activate Object command\)](#)

Journaling considerations

Understand the differences between traditional local journaling and remote journaling. Local journaling is used by default and is available only in asynchronous mode.

Each journal can have up to one terabyte of uncommitted transactions.

Local journaling

Applications on the primary system generate database changes with local journaling. These changes generate journal entries that are written to a local journal receiver. These changes are then transmitted by the product to the backup system, where the same changes are applied. In iCluster, local journaling is used by default.

For local journaling, the following implementation considerations apply:

- You can be more selective in what you journal. For example, choose to avoid journaling BSFs with non-journalized BSF support.
- Asynchronous data updates have minor impact to applications on the primary system.
- Local journaling is available only in asynchronous mode. Asynchronous mode results in very less latency, because control returns to the applications after the journal entries are written to the journal.

Remote journaling

Remote journaling is an IBM i feature that replicates journal entries for an object on the primary node (local journal) to a journal on the backup node (remote journal). The local and remote journals contain the same journal entries.

When a group that is using remote journals is started, remote journals are automatically created if the journals do not exist. Remote journaling occurs independently from iCluster operations and provides an alternative method for transporting data from the primary node to a backup node.

Transactions from a remote journal on the backup node are applied to the replicated objects on the backup node. Using a remote journal in a replication group can improve system performance on the primary node because the backup node does more of the replication processing. Configure the local journal to update the remote journal synchronously or asynchronously.

For remote journaling, the following implementation considerations apply:

- Remote journaling does not let you select what journal entries are transmitted to the backup system. All entries in the primary system journal are sent to the remote journal.
- A synchronous data update can have a performance impact on applications and journaling throughput on your primary system.
- Remote journaling is configured at the journal level.

Uncommitted transactions

Each journal can have up to one terabyte of uncommitted transactions.

Related information

[Generic object specifiers](#)

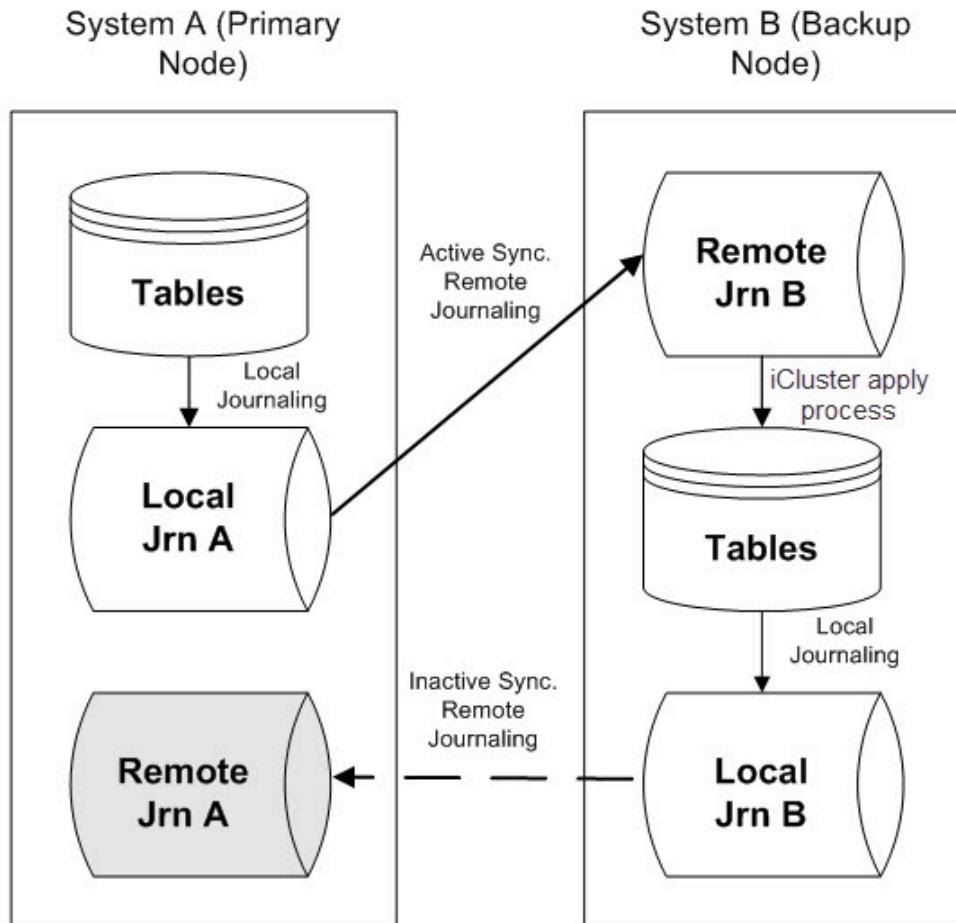
Uncommitted transactions

Each journal can have up to one terabyte of uncommitted transactions.

Remote journaling

Review the recommended configuration for remote journaling between two systems.

For example, the following diagram demonstrates the recommended configuration for remote journaling between System A (primary node) and System B (backup node).



As shown, a remote journal is configured on both System A and System B. The purpose of these remote journals is to receive journal entries from the other system when it acts as a primary node. Whenever a system acts as a primary node (System A), the remote journal that resides on that system (Remote Jrn A) must be inactive. The inactive status prevents the unnecessary transmission of journal entries from the backup system (System B) to the remote journal on the primary node (System A).

When a role switch is performed, the remote journal on the new backup node is automatically activated from the new primary node.

Note the following:

- Use the After Role Switch User Exit program to activate or deactivate the remote journals on the applicable nodes.
- The sample program, **RMTJRNNSWO**, is in the **QACLSRC** file in the product library. The name of the member is **RMTJRNNSWO**.

Related information

[Configuring remote journaling](#)

[Starting a role switch](#)

Configuring remote journaling

When a group that uses remote journals starts, remote journals are automatically created when the journals do not exist. However, there are scenarios where you must manually create a remote journal.

Remote journals are created according to the following rules:

- The remote journals are automatically created in the **DMRMTJRN** library on the backup node. If there is already a journal with the same name in the **DMRMTJRN** library, the remote journal will be created in the **DMRMTJRNDR** library. For the journals in the **AAAJRNLIB** library that are created with the **DMAUTOCFG** (Auto-configure iCluster) command, the remote journals are created in the **AAARMTJRN** library on the backup node.
- If a name conflict occurs with an existing journal in both the **DMRMTJRN** library and the **DMRMTJRNDR** library, the remote journal creation fails. You must manually create the remote journal on the backup node.
- For DDM secured connections with password requirements, the **DMCLUSTER** user profile must have a server authentication entry for this relational database directory entry.
- If a name conflict occurs with an existing journal in the **AAARMTJRN** library, the remote journal creation fails. You must manually create the remote journal.

1. On the primary node, to add a relational database entry for the database on the backup node, run the Add Relational Database Directory Entry (**ADDRDBDIRE**) IBM i command.
2. On the primary node, to create a journal receiver, run the Create Journal Receiver (**CRTJRNRCV**) IBM i command.
3. On the primary node, to create a journal that uses the journal receiver, run the Create Journal (**CRTJRN**) IBM i command.
You must configure the journal to have the system manage the changes to the journal receiver.
4. On the primary node, to add a remote journal, run the Add Remote Journal (**ADDRMTJRN**) IBM i command with the following actions:
 - a. Set the relational database to the database directory entry that you created in Step 1.
 - b. Set the names of the source journal and target journal to the journal that you created in Step 3.
The target journal name must be the same value as the source journal.
 - c. Set the source journal library to the library of the journal that you created in Step 3.
 - d. Set the target journal library to a library that is different from the source journal library.
The source and target journal libraries must be different libraries.
 - e. Set the remote journal type.
5. On the primary node, to activate the remote journal, run the Change Remote Journal (**CHGRMTJRN**) IBM i command with the **JRNSTATE(*ACTIVE)** parameter.
To configure the journal to perform synchronous updates, use the **DELIVERY(*SYNC)** parameter.

Note: When a remote journal group starts, the remote journal is automatically activated (if it is not already activated) with the default value for the **DELIVERY** parameter in the **CHGRMTJRN** command.

6. To add the group on any node, run the **DMADDGRP** (Add Group) command.
As an alternative, use the **DMCHGGRP** (Change Group) command to modify existing groups. When running this command, specify the following values to use remote journaling:
 - Set the default database journal to use the journal that you created in Step 3.

- Set the journal location to *REMOTE.
7. Perform the normal configuration and administration tasks for your replication group, including selecting the objects to be replicated, setting or marking journal positions, and starting the applications and groups.

Related information

[Server authentication for secure DDM connections](#)

Role switching with remote journals

To perform failover or switchover operations on a group that uses remote journaling, complete the required configuration tasks.

For example, consider a group that replicates data from NODE A to NODE B. In this group, NODE A is the primary node and NODE B is the backup node. Since the group uses remote journaling, NODE A has a remote journal on NODE B.

Since a switchover or failover reverses the roles of the nodes in a group, NODE B must be able to become the primary node in the group. This switch requires that NODE B also has a remote journal on NODE A. The remote journals are activated and deactivated as required for role switching. The remote journal is created on the new backup node if it does not exist, without a naming conflict with another journal.

Switchovers and failovers with synchronous remote journaling

Remote journal groups that use the synchronous delivery mode for the remote journal or journals might have suspended journaled objects after a failover or switchover, if the group includes only after journal images. To prevent these suspensions, configure your group to include both before and after journal images.

After a switchover or a failover, database changes on the previous primary node are undone when the changes were not synchronized with the remote journal before the primary node failed or switched over. To undo such changes, the before and after journal images are required. If the objects with unsynchronized changes are not journaled with both journal images, the objects are suspended with the UBI (missing before image) suspension reason when mirroring is started from the new primary node.

To include both before and after journal images, perform one of the following tasks:

- To change this property for one group, run the **DMCHGGRP–Change Group** command to set the **JRNBA** property to *BOTH. New objects that are added to this group use this setting, but existing objects in the group must be manually changed to include both before and after images.

To change this setting for individual objects, run the **CHGJRNOBJ** command with the IMAGES(*BOTH) parameter.

- To change this property globally for all groups, set the journal images attribute of the PF property to *BOTH by running the **DMSETSVAL–Set Cluster System Values** command. New groups use this setting by default, but groups that were explicitly configured to include only after images need to have their JRNBA property set to either *BOTH or *CLUSTER by running the **DMCHGGRP–Change Group** command.

Related information

[DMCHGGRP \(Change Group command\)](#)

Setting journal positions

Understand the differences between the DMSETPOS (Set Journal Start Position) command and the DMMRKPOS (Mark Journal Positions) command, which impact how replication is started after setting or marking journal positions. The

starting entry for each journal that is used by the group for replication is set on the primary node of the group.

The **DMSETPOS** command requires you to specify a journal position for all journals in a group. The specific journal entry can be entered directly with a special value such as *CURSRCPOS, which represents the current last entry in the journal, or determined by the command if a date and time is entered.

The **DMSETPOS** command also allows you to specify that the last applied journal position is used as the starting point for replication when the group is restarted. This command is typically used for recovery purposes to resume replication from a journal position back in time.

The **DMMRKPOS** command does not require you to specify a journal position for all journals in a group. This command automatically determines the journal positions by using the current sequence numbers in all of the relevant journals for the group and is typically used for initial synchronization. For example, use this command after a full tape save and restore to the backup node when replication is started from the current point in time.

Starting replication after setting journal positions

To start replication with the **DMSTRGRP** (iCluster Start Group) command after you set or marked journal positions with the **DMSETPOS** command or the **DMMRKPOS** command, note the following considerations:

- If you issue the **DMMRKPOS** command for a group and then start replication, you must specify *YES for the **USEMARKED** parameter in the **DMSTRGRP** command. Replication starts at the journal positions that were marked with the **DMMRKPOS** command.
- If you issue the **DMSETPOS** command for a group and then start replication, you must specify *NO for the **USEMARKED** parameter in the **DMSTRGRP** command. Replication starts at the journal positions that you set with the **DMSETPOS** command.

Related information

- [DMSTRGRP \(Start Group command\)](#)
[DMMRKPOS \(Mark Journal Positions command\)](#)
[DMSETPOS \(Set Journal Start Position command\)](#)

Setting a journal start position

Set the starting entry in the journal on the primary node in the recovery domain for the group or application.

1. Enter the following command:
DMSETPOS
2. Type the name of the replication group in the **Group or resilient application** field.
3. Specify which database or system audit journals are affected in the **Journal** field.
Choose one of the following values:
 - <name>—Specifies the journal name.
 - *ALL—Specifies all journals that are being scraped by the group that is specified in the **Group** field, including the system audit journal.
4. Specify the name of the database or system audit journal receiver in the **Starting journal receiver** field.
Choose one of the following values:
 - *CURRENT—Specifies the current journal receiver for the journal. By default, this value is used.
 - *CURCHAIN—Specifies the journal receiver that is determined from the timestamp information of the starting entry.
 - <name>—Specifies the journal receiver name.
5. If *CURCHAIN or *CURRENT was not chosen, specify the name of the library where the journal receiver is

located in the **Library** field.

Choose one of the following values:

- <name>—Specifies the library name.
- *LIBL—Specifies the set of libraries in your library list.

6. To specify the starting point for processing when replication resumes, choose one of the following methods:

- Specify the sequence number of the entry in the journal that iCluster starts processing when replication resumes in the **Starting large jrn position** field. This parameter cannot be used with **STRDATE**, or the **STRTIME** parameter if the **JRNRCV** parameter value is ***CURCHAIN**. Choose one of the following values:
 - <number>—Specifies a sequence number of up to 20 digits.
 - *LASTAPY—Specifies the last journal entry in the source journal that was applied to the backup node.
 - *CURSRCPOS—Specifies the last journal entry in the source journal.
- In the **Start date** field, specify the date of the entry from which iCluster should start processing from when replication is restarted .
- In the **Start time** field, specify the time of the entry from which iCluster should start processing from when replication is restarted.

Marking the current journal positions for replication

During initial synchronization or when a group needs to be resynchronized, mark the current journal positions so that replication starts from the marked position.

1. Enter the following command:

DMMRKPOS

2. In the **Group or resilient application** field, type the name of the replication group.

The possible values are:

- *ALL— Marks positions for all groups regardless of their current recovery domain.
- *PRIMNODE—Marks position for groups with a primary node of the value specified for the Primary node field.
- <name>—The name of the group whose positions are to be marked.

3. In the **Primary** node field, specify the name of the primary node for which groups are being marked.

The possible values are:

- *CURRENT— Mark journal positions for groups using the current node as a primary. By default, this value is used.
- <name>—The name of the primary node for groups being marked. This value is only used when the GROUP parameter is set to *PRIMNODE.

4. In the **Change object auditing** field, specify whether iCluster should automatically change the object auditing value of all objects in replication scope on the primary node. iCluster requires that this value be ***CHANGE** or ***ALL**.

The possible values are:

- *YES—Changes the object auditing value of all objects in replication scope to *CHANGE. By default, this value is used.
- *NO—Does not change the object auditing value. If you are certain all objects being replicated are already set to *CHANGE or *ALL, using this value will make DMMRKPOS run faster.

5. In the **Save objects** field, specify whether to save objects in save files.

The possible values are:

- *NO—Does not save the objects in the replication scope. By default, this value is used.
- *YES—Saves the objects in the replication scope.

6. In the **Save file library** field, type the library name on the primary node.

The save files are saved in this library when the **SAVOBJ** parameter is *YES on the primary node. The possible values are:

- *NONE—Specifies no library. By default, this value is used.
- <name>—The library name where the save files are saved. Required when **SAVOBJ** *YES.

7. In the **Save target release** field, type the target release of the save files when the **SAVOBJ** parameter is *YES on the primary node.

The possible values are:

- *CURRENT—Specifies the target release of the save files is the same as the operation system release on the primary node. By default, this value is used.
- <charval>—The operating system release in format VxRxMx, where:
 - Vx is the version
 - Rx is the release
 - Mx is the modification

Related information

[**DMMRKPOS \(Mark Journal Positions command\)**](#)

Changing journal receivers

Journal receivers are automatically detached and new receivers are created as changes occur. Delete any unnecessary receivers to keep disk space at an acceptable level.

1. Enter the following command:
`CHGHAJRN`
2. In the **Journal** field, type the name of the journal on the primary node for which you are generating a journal receiver.
3. In the **Library** field, type the name of the library where the journal resides.
4. In the **Delete processed receivers** field, specify if fully processed receivers associated with the named journal are deleted.
Choose one of the following values:
 - *YES—Deletes fully processed journal receivers and remote receivers that are associated with the

named journal.

- *NO—Does not delete fully processed journal receivers and remote receivers that are associated with the named journal. By default, this value is used.

5. In the **Min age of receivers to delete** field, specify the number of days that fully processed journal receivers must be older than before they are deleted.

Choose one of the following values:

- <days>—Specifies a value from 1–9999 days.
- *NONE—Does not consider the age of the receivers when deleting fully processed journal receivers. By default, this value is used.

Related information

[CHGHAJRN \(Change Journal Receiver command\)](#)

Displaying replication status of journal receivers

Determine the earliest journal position and its receiver, that is fully processed across all of the replication groups that use a specific journal.

Knowing the earliest journal position allows you to delete the receivers in the receiver chain after they are fully processed.

1. Enter the following command:
DSPHAPOS
2. In the **Journal** field, type the name of the journal.
3. In the **Library** field, type the name of the library where the journal resides.
4. Use the RTVHAPOS (Retrieve iCluster Journal Position) command in a CL program to retrieve the information about the earliest fully processed journal position and its receiver into CL variables.

Related information

[DSPHAPOS \(Display Journal Information command\)](#)

Verifying the audit journal

Verify that the audit journal exists in the QSYS library and that the audit-related operating system values are set correctly.

1. Enter the following command:
VFYHAJRN
2. In the **Audit objects in QTEMP** field, specify whether to audit objects in the QTEMP library.
Choose one of the following values:
 - *YES—Audits objects in the QTEMP library.
 - *NO—Does not audit objects in the QTEMP library. By default, this value is used.
3. In the **Audit Spooled files** field, specify whether to audit spooled file functions.
Choose one of the following values:
 - *YES—Audits spooled file functions.
 - *NO—Does not audit spooled file functions. By default, this value is used.

Related information

[VFYHAJRN \(Verify Audit Journal command\)](#)

Synchronization points

Entries can be placed in journals to define synchronization points for the journal scrape processes on the primary node and the journal entry apply process on the backup node.

Define a user exit program to be invoked when replication jobs arrive at the journal entries.

A replication job is any iCluster replication job that is associated with group activities. This list of jobs includes, but is not limited to: OMGROUPJOB, HADDJS (database journal scrape), HADSFP (save file process), and HADTUP (target update process).

Use the sync point journal entries when it is necessary to synchronize operations on primary and backup nodes. After defining a synchronization journal entry on the primary node, synchronization is achieved when the synchronization point journal entry is reached on the backup node. Define a user exit program to perform an operation with the user-defined data that is passed to the program when the synchronization point is reached.

A replication job on a backup node processes journal entries until it reaches the synchronization journal entry that is generated by this command. At this point, the job waits for the group's other replication jobs to reach this synchronization point journal entry. Synchronization is achieved when all active replication jobs for the group reach the sync point journal entry. This means that one or more jobs will wait at the sync point journal entry until all replication jobs for the group reach the sync point.

Specify the same user exit program to be invoked at different sync points (journal entry scrape, receive, and apply). If it is invoked when journal entries are being scraped, the program must reside on the primary node. If it is invoked at journal receive or apply times, the user exit program must reside in the same location on the backup node. To specify different user exit programs at each synchronization point, use this command multiple times (one time for each sync point) so that you can specify a different user exit program.

If the user exit program cannot be located, an error message is generated and mirroring continues.

For defining sync points, note the following:

- Normal mirroring operations are not active during the time that the user exit program is running. Mirroring resumes after the user exit program runs to completion, unless a special value is returned through one of the user exit program arguments.
- You can minimize the execution time of the user exit program so that mirroring can resume as soon as possible.

Setting a synchronization point

Run programs on the backup node at a set synchronization time. For example, perform a backup on the backup node when the interactive processing completes. Set a synchronization point so that both systems can synchronize to the same time.

After you set a sync point, you cannot delete or change the sync point journal entry. Exercise caution before you use this command.

1. Enter the following command:
`DMSETSYNC`
2. In the **Group or resilient application** field, type the name of the replication group.
3. In the **Sync point exit program** field, specify the name of the user exit program to invoke when all active replication jobs reach the checkpoint journal entry.
Choose one of the following values:
 - <name>—Specifies the user exit program.
 - *NONE—Specifies that no user exit program is called.

4. In the **Library** field, specify the name of the library where the user exit program resides. Choose one of the following values:
 - <name>—Specifies the library name.
 - *PRODLIB—Specifies the iCluster installation library.
5. In the **Synchronize at scrape** field, specify whether to synchronize group replication jobs when journal entries are scraped on the primary node. You cannot set a scrape synchronization point for groups whose journal location is *REMOTE. Choose one of the following values:
 - *YES—Synchronizes group replication jobs at the checkpoint journal entry and invokes the user exit program.
 - *NO—Does not synchronize group replication jobs at the checkpoint journal entry and does not invoke the user program. By default, this value is used.
6. In the **Synchronize at receive** field, specify whether to call the user exit program when all the receive jobs have passed through the sync point. Choose one of the following values:
 - *YES—Invokes the sync point user exit program.
 - *NO—Does not call the user exit program when all the receive jobs have passed the sync point. By default, this value is used.
7. In the **Synchronize at apply** field, specify whether to synchronize replication jobs when journal entries are applied on the backup node. Choose one of the following values:
 - *YES—Synchronizes group replication jobs at the checkpoint journal entry and invokes the user exit program. By default, this value is used.
 - *NO—Does not synchronize group replication jobs at the checkpoint journal entry and does not invoke the user program.
8. In the **Exit program data** field, identify up to 400 bytes of data of user-defined data to pass to the user exit program, which is specified through the **USREXIT** parameter.

Related information

[Passing arguments to sync point user exit programs](#)

[DMSETSYNC \(Set Sync Point command\)](#)

Passing arguments to sync point user exit programs

If user exit programs are called at a sync point that is set with the DMSETSYNC (Set Sync Point) command, specific arguments are passed to the program.

Table: Arguments passed to sync point user exit programs

Argument	Type	Len.	Description
User Exit Point	Character	1	<p>The point where the user exit program was called. Exit point is one of the following values:</p> <ul style="list-style-type: none"> • S: At journal entry scrape on the primary node • R: At journal entry receive on the backup node • A: At journal entry apply on the backup node <p>To stop replication after completion of the user exit program, return the value '9'. Replication can be stopped for scrape and apply sync points, but not for receive sync points.</p>

Argument	Type	Len.	Description
Backup Node Name	Character	10	The name of the backup node in the replication group.
Replication Group Name	Character	10	The group that had its jobs synchronized at the checkpoint journal entry.
User Data	Character	400	The user-defined data that is specified by the DMSETSYNC–Set Sync Point command.

Related information

[DMSETSYNC \(Set Sync Point command\)](#)

Recovery checkpoints

Retrieve the recovery checkpoint to find the approximate backup node journal position that corresponds to a given primary node journal position. This journal position can be used as the starting journal position in a recovery situation.

Retrieving a recovery checkpoint from the command line

Use the RTVRCVPT (Retrieve Recovery Checkpoint) command to retrieve the recovery checkpoint position from the journal on the backup node given a journal key, journal, receiver, and position from the primary node.

The RTVRCVPT command is used only from the command line. To use values retrieved by the command in a CL program, use the RTVRCVPTR (Retrieve Recovery Checkpoint (CL Program)) instead of this command.

- From a command line, enter the following command:

RTVRCVPT

- To specify the key of a recovery checkpoint, use the **JRNKEY(<journal key>)** parameter.

Note: This value was specified when the DMADDGRP or DMCHGGRP commands were issued for the group.

- To specify the name of the journal on the primary node, use the **JRN(<library>/<journal>)** parameter.
- To specify the name of the journal receiver on the primary node, use the **OLDRCV(<library>/<journal receiver>)** parameter.
- To specify the journal position (up to 20 digits) on the primary node, use the **OLDPOSLRG(<journal position>)** parameter.
- Press Enter.

Related information

[RTVRCVPT \(Retrieve Recovery Checkpoint command\)](#)

[RTVRCVPTR \(Retrieve Recovery Checkpoint \(CL Program\) command\)](#)

Performance of replication groups

To gauge the performance of specific groups in the cluster, use the DMLOGENT (Log Journal Entry) command to record the times that data is scraped and applied.

Logging a journal entry

Log a journal entry to create the **DMLOG** database file in the ICLUSTER library on the backup node, if it does not already exist.

The DMLOGENT (Log Journal Entry) command inserts one journal entry into each of the journals that are scraped by a group on the primary node. When the entries for each journal are processed by the group's apply processes on the backup node, a record for each entry is placed into the **DMLOG** database file on the backup node in the ICLUSTER library. Each record in the file contains the time that the journal entry was scraped and the time that it was applied, as well as the name of the group, the journal, and the journal receiver that contains the entry.

1. Enter the following command:
DMLOGENT
2. In the **Group or resilient application** field, type the name of the replication group for which logging information is recorded.
3. In the **Entry type** field, specify the type of logging information to create.
Choose one of the following values:
 - *INFO—Specifies an informational entry. By default, this value is used.
 - *STRRUN—Specifies the beginning of replication.
 - *ENDRUN—Specifies the end of replication.
4. **Optional:** In the **User entry data** field, include a user defined field of up to 400 characters.
5. In the **Entry data area** field, specify the name of the data area for the log file.
Choose one of the following values:
 - <name>—Specifies the data area.
 - *NONE—Specifies that no data area is set. By default, this value is used.
6. In the **Library** field, specify the library where the data area resides.
Choose one of the following values:
 - <name>—Specifies the data library name.
 - *PRODLIB—Specifies your iCluster installation library.
7. Press Enter.

Related information

[DMLOGENT \(Log Journal Entry command\)](#)

Sync check operations

During replication, changes to replicated objects are applied on the backup node. A sync check verifies whether the objects in a replication group are equivalent on the primary and backup nodes.

A sync check generates the attributes of the selected objects on the primary node and compares them with the attributes of the selected objects on the backup node.

Sync checks other than continuous sync checks should be run at a time when you expect little update activity to the objects being checked. If an object is exclusively locked by an end user application, it might not be able to be checked and will be reported as **Locked** in the **Out of Sync or In Use Object List** portion of the sync check report.

Perform regular sync checks and review the sync check reports to ensure that the replicated objects on the primary and backup systems are equivalent. The following commands verify that the primary and backup nodes are synchronized:

- DMSTRSC (Start group sync check) or STRHASC (Start group sync check) checks objects for a replication group.
- DMSTRSCUSR (Start user-specified sync check) or STRHASCUSR (Start user-specified sync check) checks a subset of the objects that are replicated in a group.
- STRCNSC (Start continuous sync check) continuously checks objects for a replication group.

Related information

- [DMSTRSC \(Start Group Sync Check command\)](#)
- [DMSTRSCUSR \(Start User-specified Sync Check command\)](#)
- [STRHASCUSR \(Start User-Specified Sync Check command\)](#)
- [STRCNSC \(Start Continuous Sync Check command\)](#)

Setting attributes for sync checks

Select specific attributes or use the default attributes for sync checks.

Specify which attributes to verify for the following types of sync checks:

- File-attribute (*FILEATTR)
- Object-attribute (*OBJATTR)
- Full (*FULL) sync checks

Select file, member, record, field, and object attributes.

1. Enter the following command:
SELSCATTR
2. To specify the sync check attributes, enter one of the following values in the **Use default selections** field:
 - *YES: Use the default attribute selection for the sync check.
 - *NO: Use the selected attributes. Press PgDn to view the list of attributes. Type ***YES** for the attributes to include in the sync check.

Related information

- [SELSCATTR \(Select Sync Check Attributes command\)](#)

Selecting sync check attributes for a group

Select specific attributes or use the default attributes for sync checks of a replication group.

Select the object-level attributes to check in the following types of sync checks for a group:

- Object attribute (*OBJATTR)
- Full (*FULL) sync checks

Select file, member, record, and field attributes.

1. Enter the following command:

DMSCATTR

2. In the **Group name** field, type the name of the replication group.
3. To specify the sync check attributes, enter one of the following values in the **Use default selections** field:
 - *YES—Use the default attribute selection for the sync check.
 - *NO—Use the selected attributes. Press PgDn to view the list of attributes. Type ***YES** for the attributes to include in the sync check.

Starting a sync check for a group

Set sync check options and start a sync check to check all objects in the replication group.

1. Enter the following command:

DMSTRSC

2. In the **Group name** field, type the name of the replication group.
3. In the **Sync check type** field, specify the type of sync check you want to perform.

Choose one of the following values:

- *FULL—Performs an *OBJATTR sync check on all objects that are replicated by the group and a *FILEATTR sync check on database file objects. If the selected group is not active, only an *OBJATTR sync check is run and a message is issued. By default, this value is used.

You can optionally run a checksum sync check as part of the continuous *FULL sync check by specifying **YES** for the Check file contents element of the **FULLSCOPTS** (*FULL sync check options) parameter of the **DMSETSVAL** command.

Similarly, you can optionally run an obsolete object sync check by specifying **YES** for the Check obsolete objects element of the **FULLSCOPTS** (*FULL sync check options) parameter of the **DMSETSVAL** command.

- *FILEATTR—Checks whether the *FILE objects on the primary node exist on the backup node. It also checks that the attributes of the primary node *FILE objects match the attributes of the backup node *FILE objects.
- *OBJATTR—Checks whether objects exist on the backup node. Also checks their authorities and journaling attributes (for journaled objects). BSF and native objects are checked. User profile attributes are checked for replicated user profiles.
- *CHECKSUM—Checks database *FILE objects, IFS *STMF and DLO (Document Library Objects) that match the group specifiers. Compares attributes (excluding record counts) and results of MD5 checksum calculations on the records of the files.
- *CHKOBSLETE—Checks for objects that exist on the backup node but do not exist on the primary node.

4. In the **Output** field, specify the type of report to be generated.

Choose one of the following values:

- *MISMATCH—Displays those objects that are not synchronized when writing the sync check report.
 - *ALL—Displays the objects that are not synchronized and the details of the out-of-sync attributes.
 - *NONE—Does not generate a spooled file report for the sync check. The results of the sync check are recorded in a set of database files on the backup node and can be retrieved with the DMSCRPT (Sync Check Report) command. By default, this value is used.
5. In the **Lock files** field, specify whether to lock the files during a file attribute (*FILEATTR) sync check. Choose one of the following values:
- *YES—Locks files during the sync check. If a file cannot be locked, then it will not be involved in the sync check. It will be listed in the sync check report under the **Out of Sync** or **In Use Object** list with reason code LCK and a message will be placed in the event log and job log.
 - *NO—Does not lock files during the sync check. Ensure that the files are not modified during the course of the sync check. By default, this value is used.
6. In the **Submit sync check job** field, specify whether to submit a job to the XDMCLUSTER subsystem to run the sync check. This parameter applies only when the DMSTRSC command is invoked on the primary node. Choose one of the following values:
- *YES—Runs the sync check by a submitted job in the XDMCLUSTER subsystem. By default, this value is used.
 - *NO—Runs the sync check in the job where the DMSTRSC command is invoked.
7. In the **Delete obsolete objects** field, specify whether to delete obsolete objects on the backup node as part of full (*FULL) and check obsolete (*CHKOBSELETE) sync checks. The *YES option on the DLTOBSOBJ parameter is only available for the *CHKOBSELETE sync check type, and the *FULL sync check type when the RUNCHKOBSL parameter is set to *YES.

Note: Obsolete objects are objects that exist on the backup node, but not on the primary node. Objects might be reported as obsolete by the sync check in certain situations. For example, objects that were created during the sync check run in which they were reported as obsolete. Wait for a second complete continuous sync check, or end the continuous sync check, and then run this command without this option to see if the obsolete objects are removed.

- Choose one of the following values:
- *NO—Does not delete obsolete objects as part of full (*FULL) and check obsolete (*CHKOBSELETE) sync checks. By default, this value is used.
 - *YES—Deletes obsolete objects as part of full (*FULL) and check obsolete (*CHKOBSELETE) sync checks. The *YES option on the DLTOBSOBJ parameter is only available for the *CHKOBSELETE sync check type, and the *FULL sync check type when the RUNCHKOBSL parameter is set to *YES.
8. In the **Activate all out of sync objects** field, specify whether to activate the out-of-sync objects for groups with the refresh type specified after a sync check, unless the objects are suspended with a reason code that prevents auto-reactivation of the object. For example, the DMACTOOS (Activate All OOS Objects) command is set to *YES. Choose one of the following values:
- *YES Runs the DMACTOOS command when the sync check processing completes.
 - *NO: Does not run the DMACTOOS command at the end of sync check processing. By default, this value is used.
9. In the **Send output to** field, specify the node or nodes to send the spooled file results of the sync check. This requires that the **OUTPUT** parameter is *ALL or *MISMATCHED. Choose one of the following values:
- *ALL—Sends the spooled file results of the sync check from the backup node to all active nodes in the cluster. By default, this value is used.
 - *PRIMARY—Sends the spooled file results of the sync check from the backup node to the primary node, as long as the primary node is active.
 - *BACKUP—The spooled file results are retained only on the backup node.

-
10. In the **Check file contents with *FULL** field, specify whether to check the contents of physical database files, IFS *STMF objects, and DLO (Document Library Objects) that are replicated by the group as part of the full (*FULL) sync check.

This parameter applies only to *FULL sync checks. Choose one of the following values:

- *CLUSTER - uses the cluster system value assigned for this option in the DMSETSVAL command. By default, this value is used.
- *NO—Does not check the contents of physical database files, IFS *STMF objects, and DLO (Document Library Objects).
- *YES—Compares the contents of the physical database files, IFS *STMF objects, and DLO (Document Library Objects) using the same method that is used for *CHECKSUM sync checks.

11. In the **Check obsolete obj with *FULL** field, specify whether to check for obsolete objects as part of the *FULL sync check.

This parameter applies only to full (*FULL) sync checks. Choose one of the following values:

- *CLUSTER - uses the cluster system value assigned for this option in the DMSETSVAL command. By default, this value is used.
- *NO—Does not check for obsolete objects.
- *YES—Checks for obsolete objects during a *FULL sync check.

12. In the **Auto-repair** field, specify whether to automatically repair mismatched objects.

Automatic repair occurs only when the replication group is active during sync check. Choose one of the following values:

- *YES—Repairs mismatched *FILE objects, IFS *STMF objects, and DLO (Document Library Objects) after a *CHECKSUM sync check.
- *NO—Does not repair mismatched objects. By default, this value is used.
- *AUTH—Refreshes private authorities when a *FULL or *OBJATTR sync check discovers that authorities (including ownership and primary group) are found to be mismatched.
- *JRN—Automatically corrects journaled objects whose journaling is found to be mismatched by a *FULL, *FILEATTR, or *OBJATTR sync check. The repaired object on the backup node is journaled to the same journal as the object on the primary node.
- *CBU—Indicates that database *FILE objects whose "Created by user" attribute is found to be mismatched by a *FULL or *OBJATTR sync check, and will be refreshed as part of the sync check processing.
- *CRD—Refreshes library (native) objects when their "Creation date and time" attribute is found to be mismatched by a *FULL or *OBJATTR sync check.

13. In the **Checksum on suspended objects** field, specify whether to perform a *CHECKSUM sync check on suspended objects to verify that database *FILE objects, IFS *STMF objects, and DLO (Document Library Objects) match the group specifiers.

The *CHECKSUM sync check compares attributes (excluding record counts) and results of MD5 checksum calculations on the object contents. Choose one of the following values:

- *YES—Checks suspended *FILE objects, IFS *STMF objects, and DLO (Document Library Objects) during a *CHECKSUM sync check.
- *NO—Does not check suspended *FILE objects, IFS *STMF objects, and DLO (Document Library Objects). By default, this value is used.

Related information

[DMSCATTR \(Select Sync Check Attributes for a Group command\)](#)

[DMSTRSC \(Start Group Sync Check command\)](#)

[DMSCRPT \(Sync Check Report command\)](#)

Starting a user-specified sync check

Set sync check options and start a user-specified sync check on a subset of the objects that are replicated in the group.

1. Enter one of the following commands:
 - On the primary node, enter **STRHASCUSR**.
 - On any node in the cluster, enter **DMSTRSCUSR**.
2. For the **STRHASCUSR** command on the primary node, type the name of the backup node in the **Backup node name** field.
3. In the **Group name** field, type the name of the replication group.
4. In the **Sync check type** field, specify the type of sync check to perform.
Choose one of the following values:
 - *FULL—Performs an *OBJATTR sync check on all objects that match the object filter and a *FILEATTR sync check on database file objects. If the selected group is not active, only an *OBJATTR sync check is run and a message is issued. By default, this value is used.
 - *FILEATTR—Checks whether the *FILE objects on the primary node exist on the backup node. It also checks that the attributes of the primary node *FILE objects match the attributes of the backup node *FILE objects.

*FILEATTR sync check requires that the files being checked are journaled, and that all of the group's required database journal scrape and apply processes are active. Source physical files can be sync checked with *FILEATTR sync check if either of the following two requirements are met:

 - If the file is journaled and there is a scrape and apply process for the file's journal active for the group that is being sync checked.
 - If the file is not journaled but there is a scrape and apply process active for the default database journal for the group that is being sync checked.

*FILEATTR sync check checks objects of attribute type PF-SRC only if the group the object belongs to replicates at least one object that is being journaled. For example, at least one PF-DTA object in the group ensures *FILEATTR sync checks include the PF-SRC objects.

 - *OBJATTR—Checks whether objects exist on the backup node. Also checks their authorities and journaling attributes (for journaled objects), and user profile attributes for replicated user profiles. Checks BSF and native objects.
 - *CHECKSUM—Checks database *FILE objects, IFS *STMF objects, and DLO (Document Library Objects) that match the object filter and are replicated by the group. Compares attributes (excluding record counts) and results of MD5 checksum calculations on the object contents.
5. In the **Lock files** field, specify whether to lock the files involved in the sync check during a file attribute sync check.
Locking ensures that an accurate synchronization occurs between the primary and backup nodes, and increases the time to complete the operation. Choose one of the following values:
 - *YES—Locks files during the sync check. If a file cannot be locked, then it is not involved in the sync check, and a message is logged in a spooled file.
 - *NO—Does not lock files during the sync check. Make sure that the files are not modified during the course of the sync check. By default, this value is used.
6. To specify whether to check native objects or BSF objects, enter the following filters for the sync check.
 - Native objects: In the following **Object** fields, specify the object specifiers:
 - **Object**
 - **Library**
 - **Object type**
 - **Object attribute**

- BSF objects: In the **Path specifier** field, identify the location of the BSF objects. Enclose the path in single quotation marks and start with a forward slash character.

7. **Optional:** For the STRHASCUSR command on the primary node, specify the date to start the sync check in the **Schedule date** field.

Choose one of the following values:

- <date>—Specifies a particular date in *MMDDYY* format, where *MM* is the month, *DD* is the day, and *YY* is the year.
- *CURRENT—Performs the sync check when the command is issued. By default, this value is used.

8. In the **Library** field, type the name of the library where the object specifiers reside.

9. Specify whether to check native objects or BSF objects.

- To check native objects, in the **Object attribute** field, specify how the object attributes are filtered.

Choose one of the following values:

- *ALL—Specifies all object attributes. By default, this value is used.
- <name>—Specifies the object attribute name.
- To check BSF objects, specify the location of the BSF objects to include in the sync check in the **Path specifier** field.

Enclose the path in single quotation marks (') and start with a forward slash (/); for example: '/Dir3/Dir4/file'. The path can be 2 - 5000 characters. Generic path names of the form /mydir* are supported, where the generic indicator * is the final character of the path name. Generic path names include all sub-directories recursively.

10. **Optional:** For the STRHASCUSR command on the primary node, specify the time to start the sync check in the **Schedule time** field.

Choose one of the following values:

- <time>—Specifies a particular time in *HHMMSS* format, where *HH* is hours, *MM* is minutes, and *SS* is seconds.
- *CURRENT—Performs the sync check when you issue this command. By default, this value is used.

11. In the **Output** field, specify the type of output for the sync check.

Choose one of the following values:

- *MISMATCH—Displays those objects that are not synchronized when the sync check report is written.
- *ALL—Displays the objects that are not synchronized and the details of the out-of-sync attributes.
- *NONE—Does not generate a spooled file report for the sync check. The results of the sync check are recorded in a set of database files on the backup node. By default, this value is used.

12. In the **Lock files** field, specify whether to lock the files during a file attribute sync check.

Choose one of the following values:

- *YES—Locks files during the sync check. If a file cannot be locked, then it is not involved in the sync check, and a message is logged in a spooled file.
- *NO—Does not lock files during the sync check. Ensure that the files are not modified during the sync check. By default, this value is used.

13. In the **Submit sync check job** field, specify whether to submit a job to the **XDMCLUSTER** subsystem to run the sync check. This parameter applies only when the command is invoked on the primary node for the group.

Choose one of the following values:

- *YES—Runs the sync check by a submitted job the **XDMCLUSTER** subsystem. By default, this value is used.
- *NO—Runs the sync check in the job where the command is invoked.

14. In the **Send output to** field, specify the node or nodes to send the spooled file results of the sync check.

This requires that the **OUTPUT** parameter is *ALL or *MISMATCHED. Choose one of the following values:

- *ALL—Sends the spooled file results of the sync check from the backup node to all active nodes in the

- cluster. By default, this value is used.
- *PRIMARY—Sends the spooled file results of the sync check from the backup node to the primary node, when the primary node is active.
 - *BACKUP—The spooled file results are retained only on the backup node.
15. In the **Auto-repair** field, specify whether to automatically repair mismatched objects. Automatic repair occurs only when the replication group is active during sync check. Choose one of the following values:
- *YES—Repairs mismatched *FILE objects, IFS *STMF objects, and DLO (Document Library Objects) after a *CHECKSUM sync check.
 - *NO—Does not repair mismatched objects. By default, this value is used.
 - *AUTH—Refreshes private authorities when a *FULL or *OBJATTR sync check discovers that authorities (including ownership and primary group) are found to be mismatched.
 - *JRN—Automatically corrects journaled objects whose journaling is found to be mismatched by a *FULL, *FILEATTR, or *OBJATTR sync check. The repaired object on the backup node is journaled to the same journal as the object on the primary node.
 - *CBU—Indicates that database *FILE objects whose "Created by user" attribute is found to be mismatched by a *FULL or *OBJATTR sync check will be refreshed as part of the sync check processing.
 - *CRD—Refreshes library (native) objects when their "Creation date and time" attribute is found to be mismatched by a *FULL or *OBJATTR sync check.
16. In the **Checksum on suspended objects** field, specify whether to perform a *CHECKSUM sync check on suspended objects to verify that database *FILE objects, IFS *STMF objects, and DLO (Document Library Objects) match the group specifiers. The *CHECKSUM sync check compares attributes (excluding record counts) and results of MD5 checksum calculations on the object contents. Choose one of the following values:
- *YES—Checks suspended *FILE objects, IFS *STMF objects, and DLO (Document Library Objects) during a *CHECKSUM sync check.
 - *NO—Does not check suspended *FILE objects, IFS *STMF objects, and DLO (Document Library Objects). By default, this value is used.
17. In the **Checksum record selection** field, specify whether to check only a subset of records or a percentage of records of a single database file during a *CHECKSUM sync check. Depending on the record length, a checksum sync check can take a long time to run. To reduce the time to run the sync check, adjust the parameters to run the sync check on a smaller subset of the files. Choose one of the following values:
- *RRNRNG—Checks a range of records by relative record number. Specify the range of records with the **STRRRN** and the **ENDRRN** parameters. By default, this value is used.
 - *PERCENT—Checks a percentage of the records of the file. Specify the percentage with the **PERCENT** parameter.
18. In the **Starting RRN for checksum** field, specify the starting relative record number for a *CHECKSUM sync check when the **SELREC** parameter value is *RRNRNG. Choose one of the following values:
- *FIRST—Indicates the first record of the file (relative record number 1). By default, this value is used.
 - <number>—Specifies the relative record number in the file.
19. In the **Ending RRN for checksum** field, specify the ending relative record number for a *CHECKSUM sync check when the **SELREC** parameter value is *RRNRNG. Choose one of the following values:
- *LAST—Indicates the last record of the file. By default, this value is used.
 - <number>—Specifies the relative record number in the file.
20. For **Percentage for checksum** in the **From** field, specify a starting point for the sync check when the **SELREC** parameter value is *PERCENT. Choose one of the following values:
- *FIRST—Starts the percentage that is specified in the **Percentage for checksum** field at the beginning of

the file. By default, this value is used.

- *LAST—Starts the percentage that is specified in the **Percentage for checksum** field at the end of the file.

21. For **Percentage for checksum** in the **Percentage** field, specify a percentage of a file to check during a *CHECKSUM sync check when the **SELREC** parameter value is *PERCENT.

Choose one of the following values:

- <number>—A number 1 - 100 that specifies the percentage of the records of a file. The default value is 100.

Related information

[STRHASCUSR \(Start User-Specified Sync Check command\)](#)

Starting a continuous sync check for a group

Specify sync check options and start a continuous sync check for a currently active replication group.

1. Enter the following command on the primary node:

STRCNSC

2. In the **Backup node name** field, type the name of the backup node.

3. In the **Group name** field, type the name of the replication group.

4. In the **Sync check type** field, specify the type of sync check.

Choose one of the following values:

- *FULL—Performs an *OBJATTR sync check on all objects that are replicated by the group and a *FILEATTR sync check on database file objects.

You can optionally run a checksum sync check as part of the continuous *FULL sync check. This can be done by specifying *YES for the "Check file contents" element of the FULLSCOPTS (*FULL sync check options) parameter of the DMSETVAL command.

Similarly, you can optionally run an obsolete object sync check by specifying *YES for the "Check obsolete objects" element of the FULLSCOPTS (*FULL sync check options) parameter of the DMSETVAL command. By default, this value is used.

- *FILEATTR—Checks whether the *FILE objects on the primary node exist on the backup node. It also checks that the attributes of the primary node *FILE objects match the attributes of the backup node *FILE objects.

- *OBJATTR—Checks whether objects exist on the backup node. Checks their authorities and journaling attributes (for journaled objects), and user profile attributes for replicated user profiles. Checks BSF and native objects.

5. In the **Delay between objects (msec)** field, specify the delay time between object checks.

Choose one of the following values:

- *SAME—Keeps the current setting for this parameter.
- *GROUP—Uses the value that is assigned to the group in the DMADDGRP (Add Group) command.
- <value>—Specifies a value between 0 - 10,000 milliseconds.

6. In the **Auto-repair** field, specify whether to automatically repair mismatched objects.

Choose one of the following values:

- *NO—Does not repair mismatched objects. By default, this value is used.
- *AUTH—Refreshes private authorities when a *FULL or *OBJATTR sync check discovers that authorities (including ownership and primary group) are found to be mismatched.
- *JRN—Automatically corrects journaled objects whose journaling is found to be mismatched by a *FULL, *FILEATTR, or *OBJATTR sync check. The repaired object on the backup node is journaled to the same

journal as the object on the primary node.

- *CBU—Indicates that database *FILE objects whose "Created by user" attribute is found to be mismatched by a *FULL or *OBJATTR sync check, are refreshed as part of the sync check processing.
- *CRD—Refreshes library (native) objects when their "Creation date and time" attribute is found to be mismatched by a *FULL or *OBJATTR sync check.

Related information

[STRCNSC \(Start Continuous Sync Check command\)](#)

Ending a sync check

End a sync check for a replication group.

1. Enter the following command on the primary node:
ENDHASC
2. In the **Group name** field, type the name of the replication group.
3. Press Enter.

Related information

[ENDHASC \(End Sync Check command\)](#)

Viewing sync check results

Use the iCluster Sync Check Report to view information about the sync check, including the list of completed sync checks, out-of-sync objects, and attribute difference details.

1. Enter the following command:
DSPHASC
2. Press Enter.
3. Select the spooled file.
4. To display the iCluster Sync Check Report, type 5 in the field beside the name of the selected spooled file.
5. Press Enter.

Related information

[DSPHASC \(Display Sync Check Results command\)](#)

Generating sync check reports

Generate sync check reports for objects on the backup node, which is based on current information in the sync check database files.

Specify sync check reports to perform the following actions:

- Send a report that contains sync check results to a spooled file.
- List groups and objects that are out-of-sync, as of the last sync checks that were run for the groups.

1. Enter the following command on the backup node:
DMSCRPT
2. In the **Group name** field, specify the name of the replication group.

-
- Choose one of the following values:
- <name>—Specifies the group name.
 - *LOCAL—Specifies all groups that have the current node as their backup node. By default, this value is used.
3. In the **Sort by** field, specify whether to sort the output (alphabetically) of the sync check report by group name or object name.
- Choose one of the following values:
- *GROUP—Sorts the results by group name. By default, this value is used.
 - *OBJECT—Sorts the results by object name.
4. In the **Output** field, specify the output type of the report.
- Choose one of the following values:
- *—Displays the sync check report on your screen. By default, this value is used.
 - *PRINT—Places the sync check report in a spooled file.
5. In the **Display mismatch details** field, specify whether to display the details of those objects that are not synchronized when writing the sync check report.
- Choose one of the following values:
- *YES—Includes details regarding out of sync objects in the report.
 - *NO—Does not include out of sync object details in the report. By default, this value is used.
6. Press Enter.

Related information

[DMSCRPT \(Sync Check Report command\)](#)

Purging sync check results

Purge obsolete records in the sync check database files (for example, records for groups that no longer exist and objects that are no longer replicated). These records can accumulate when a group is removed or an object specifier is deselected from a group when a backup node is not available.

1. On the backup node where the sync check database files are located, enter the following command:
PRGHASC
 2. In the **Purge option** field, specify the records to remove.
- Choose one of the following values:
- *ALL—Purges all records from the sync check database files.
 - *OBSOLETE—Purges only obsolete records from the sync check database files. By default, this value is used.
3. Press Enter.

Related information

[PRGHASC \(Purge Sync Check Results command\)](#)

Event log

Use the event log to view messages that are generated during replication, communication, and cluster activities. This log is maintained on each node.

Refer to the following information:

- **Display the event log:** To view all events, display the event log on a backup node. All communication and replication events for a group are sent to the backup node from the primary node.

For more information, refer to: *Displaying the event log*.

- **Clear the event log:** After you review and take action on the messages in the event log, you can remove or clear event messages from the log.

For more information, refer to: *Clearing the event log*.

Related information

[Displaying the event log](#)

[Clearing the event log](#)

Displaying the event log

Display the event log, which is maintained on each node.

To view events for a specific node, display the event log on that node.

To view all events, display the event log on a backup node. All communication and replication events for a group are sent to the backup node from the primary node.

1. From the iCluster main menu, enter 4 at the command line.
2. To specify the type of event for which to display messages, enter one of the following values in the **Event type** field:
 - *ALL—All events. By default, this value is used.
 - *REPL—Replication events.
 - *COMM—Communication events.
 - *CLUSTER—Cluster and node events.
3. To specify the replication groups to display messages for, enter one of the following values in the **Group name** field:
 - *ALL—All groups.
 - <group name>—The name of a specific replication group.
4. To filter events to display in the log, type a value in one of the following fields:
 - **Start date**—Enter a value before a specific date.
 - **Start time**—Enter a value before a specific time.
 - **Ending date**—Enter a value after a specific date.
 - **Ending time**—Enter a value after a specific time.
5. To filter events by message ID, type one of the following values in the **Message ID** field.
 - <value>—Specify from 1 to 20 message IDs.
 - *ALL—Specifies all message IDs. By default, this value is used.

- *LATNCY—Specifies only messages with ID OMI0308 that are generated when a latency threshold is exceeded.
 - *SUSPND—Specifies only messages that relate to objects being suspended.
6. In the **Output** field, specify the method of output.
Choose one of the following values:
- *—Displays output on-screen
 - *PRINT—Directs output to a spooled file.
7. In the **Detail** field, specify the level of detail in the message display.
Choose one of the following values:
- *BASIC—Displays first level message text.
 - *FULL—Displays both first-level and second-level text for each message.
8. Messages with a severity level of 40 are always displayed. In the **Display message levels** field, specify the additional message severity levels to include.
Choose up to four of the following values:
- *ALL—Specifies all message IDs. By default, this value is used.
 - 00—Displays information messages.
 - 10—Displays non-critical status messages.
 - 20—Displays stop/start messages.
 - 30—Displays messages that report recoverable errors.
9. Press Enter.

Related information

[iCluster Event Log Report \(DMDSPLOG command\)](#)

[DMDSPLOG \(Display Cluster Event Log command\)](#)

Clearing the event log

After you review and take action on the messages in the event log, remove event messages from the log.

You can also specify options to filter which messages to clear from the event log.

1. From the iCluster main menu, enter 5 at the command line.
2. To specify the type of event for which to clear messages, enter one of the following values in the **Event type** field:
 - *ALL—All events. By default, this value is used.
 - *REPL—Replication events.
 - *COMM—Communication events.
 - *CLUSTER—Cluster events.
3. When the **Event type** is *ALL or *REPL, specify the replication events to clear messages for. Enter one of the following values in the **Clear replication events** field:
 - *ALL—All groups.
 - <group name>—The name of a specific replication group.
4. When the **Event type** is *ALL or *COMM, specify the communication events for which to clear messages. Enter one of the following values in the **Clear communications events** field:
 - *ALL—All nodes.
 - *LOCAL—The local node.
 - <node name>—The name of a specific node
5. When the **Event type** is *ALL or *CLUSTER, specify *LOCAL in the **Clear communications events** field to clear

cluster events for the local node.
*LOCAL is the only supported value.

6. Press Enter.

Related information

- [**DMSETSVAL \(Set Cluster System Values\) command**](#)
- [**DMMONHA \(Monitor iCluster Replication command\)**](#)
- [**DMCLRLOG \(Clear Cluster Event Log command\)**](#)

Failover and switchover mechanisms

A switchover is a role swap procedure that is planned by the user. A failover is a role swap procedure that occurs as a result of an unplanned system failure.

A failover occurs when the primary (or production) node in a group unexpectedly becomes unavailable. To prepare for a failover, configure the cluster to either automatically move client applications and users to the backup node as a high availability solution, or configure the cluster to wait for an administrator to take action. Power failures or network failures are examples of unexpected events that can cause a node to become unavailable.

A switchover occurs when you manually swap the role of the primary node to backup node. This manual role swap causes the original backup node to replicate data to the original primary node. After a switchover, client applications and users use the new primary node (formerly the backup node). Perform a switchover for maintenance or upgrade reasons. For example, a common switchover scenario is upgrading a computer that is part of a cluster. The switchover moves users and client applications off the computer before removing the computer from the network. An important reason to perform a switchover is to verify that your cluster and applications are ready in the event of a real failure of the primary node.

Configure the backup node to automatically take over the role of the primary node during a failure. This is known as an automatic failover. In an automatic failover, the backup node assumes the role of the primary node. If the primary node later becomes active, then the objects are replicated from the former backup node to the former primary node.

A manual failover situation occurs when the primary node of a group becomes unavailable and the group is configured to wait for a response instead of performing an automatic failover. This wait time enables you to try to fix the problem instead of changing the characteristics of the group and the cluster.

Failover and switchover is not supported for the following types of groups:

- Groups that are refresh-only, if the **RSWRFSHGRP** parameter of the **DMSETSVAL** command is set to *NO.
- Groups that are local loopback.
- Groups that have a value in the **Target library** field that is not *PRIMARY.
- Groups that have an object specifier whose value in the **Target library** field that is not *PRIMARY or *GROUP.

Related information

[DMSETSVAL \(Set Cluster System Values\) command](#)

What happens during a switchover or failover

For both switchover and failover, use a set of common steps to prepare the backup node to become the primary node.

The following commands are used to perform a switchover:

- DMSTRSWO** (Switchover Group) (for active groups)
- DMCHGROLE** (iCluster Change Primary Node) (for inactive groups).

The same steps are automatically performed when a failover occurs, and the group **ROLESWITCH** parameter is set to *YES.

The following operations are automatically performed:

- Replication is ended if the group is active.

- The user-specified before role switch user exit for the group is called.
- The staging store on the backup node is drained by applying all staged entries. For remote journal groups, unprocessed entries in the remote journals are scraped and applied on the backup node.
- Journaling on backup files and objects is started, as required for replication.
- Database triggers and referential integrity constraints on backup files are enabled if they were enabled on the primary node when the role switch command was invoked.
- The restart value of identity columns in database files on the backup node is increased by 100 increments.
- The role switch is performed for the group.
- Starting journal positions are established on the new primary node for the group.
- The user-specified after role switch user exit for the group is called.
- Replication is started from the new primary to the new backup if the group was active before the switchover, and the new backup node is active.

Related information

[DMSTRSWO \(Switchover Group command\)](#)
[DMCHGROLE \(Change Primary command\)](#)

Failover mechanisms

iCluster uses two failover mechanisms to detect connectivity problems and manage node status.

The following failover mechanisms are available:

SwitchOver System

SwitchOver System simplifies node and replication group management and support while removing some of the limitations of Cluster Resource Services. SwitchOver System provides a comparable set of functionality to Cluster Resource Services, with the exception of support for switched disks.

SwitchOver System has each active node in the cluster test its connection with every other active node regularly. If a node is unavailable, then the node is declared as failed. When this happens, and the failed node is the primary node of an active group, you can also have the product notify an administrator and run a user exit.

For new installations, SwitchOver System is the default failover mechanism.

Cluster Resource Services

Cluster Resource Services is the IBM i cluster management framework that maintains the status of each node in the cluster. Cluster Resource Services handles outages by either partitioning the cluster or failing the node, depending on how the outage occurred. The partition and failed states require administrators to develop multiple recovery strategies.

Key differences in failover mechanisms

Consider the key differences in SwitchOver System and Cluster Resource Services failover mechanisms to determine which system meets the needs of your environment.

Table: Key differences in failover mechanisms

SwitchOver System	Cluster Resource Services
Supports consecutive and non-adjacent IBM i releases	Supports only consecutive IBM i releases in the same

SwitchOver System	Cluster Resource Services
in the same cluster. For example, you can have nodes running on IBM i V5R4 and V7R1 in the same cluster.	cluster. For example, you can have nodes running on IBM i V5R4 and V6R1 in the same cluster but not V5R4 and V7R1.
Uses TCP/IP to test connectivity between nodes.	Uses ICMP (ping) to test connectivity between nodes.
Handles communication failures through a single *FAILED state. This simplifies recovery from failures.	Uses *FAILED and *PARTITION states to handle communication failures.
Does not support switchable disk storage devices on the primary and backup nodes.	Supports switchable disk storage devices.
<ul style="list-style-type: none"> Both SwitchOver System and Cluster Resource Services support calling user exit programs before and after switchover operations. 	

The use of SwitchOver System is recommended, but if your environment requires support for switchable disk storage devices, you must use Cluster Resource Services.

Related information

[Cluster Resource Services failover mechanism](#)

[SwitchOver System](#)

[Changing an existing cluster to use SwitchOver System](#)

SwitchOver System

The SwitchOver System provides a comparable set of functionality to Cluster Resource Services, with the exception of support for switched disks.

Organizations can use SwitchOver System to support operational switching between two nodes. SwitchOver System provides high availability when a production system is considered to be no longer operational. When a production server is not operational, use SwitchOver System to switch operations to a backup server so that normal activities can resume as soon as possible.

SwitchOver System automatically detects when the communication link has failed and invokes the corrective measures that you specify. Determine how frequently the communication links should be polled to verify that they are working properly. To define a switch condition, specify the maximum waiting time to receive a response from a polled communication link and how many consecutive communication failures are tolerated. It also provides the necessary support for user exits and messaging.

Related information

[Cluster Resource Services failover mechanism](#)

User actions

User actions consist of generating messages into a message queue and optionally invoking a user exit program each time SwitchOver System produces a message. This process provides a way to configure the cluster to instantly notify administrators after detecting node failures. You can also use user actions to perform cleanup and other maintenance tasks before and after a role switch occurs.

Message Queues

After declaring a switch condition, SwitchOver System can periodically generate messages and place them in a message queue. Each backup node can have its own message queue. iCluster generates the message **HAS0220**, which contains a single, ten character parameter that identifies the backup node that detected the node failure. Each time a message is produced, it is placed in the queue.

When you configure the backup node, specify how many messages to generate, how long to wait between messages, and in which queue to place the messages.

User Exits

User exits offer the flexibility of invoking user-written programs to perform a specific task. Therefore, each time a message is generated, you can invoke a user exit program to accommodate requirements that are different for each working environment. Configure nodes and groups to identify the name of the user exit program that you want to invoke. The delay period between consecutive messages also indicates the period of time between consecutive user exit calls.

Related information

[Defining user exit programs](#)

[Assigning a failover message queue to a node](#)

[Specifying a failover user exit on a node](#)

Detecting node failures

Clusters that use SwitchOver System detect node failures by having each active node in the cluster poll the other active nodes at a configurable interval. After a node determines that it cannot communicate with another node, even though the status of that node is *ACTIVE, it changes the status of the node to *FAILED.

Each node has its own set of timeout and threshold values for declaring a failover. The basic properties that you can configure to detect a failure are:

- How frequently the nodes are polled.
- How long a node waits for a response from another node.
- How many consecutive polls must fail for a node to be declared *FAILED.

After determining that a primary node is unavailable using these criteria, iCluster can automatically switch the group so the backup node becomes the primary node. This property is configured for each group and two groups with the same backup node can have different values. Configure the cluster to do the following when detecting node failures or performing switchovers:

- Place messages generated by the failover into a message queue.
- Run user exit programs before and after switching the group. These user exits are configured at the group level and apply to both failovers and switchovers.

If the TCP/IP network experiences a short-term outage that does not exceed the total time interval that is associated with failure declaration (poll frequency times maximum number of poll retries), an attempt is made to restore normal communication between the nodes involved and restart the affected replication groups. The messages that describe the steps in this automatic recovery process are logged in the event log.

Related information

[Node failure examples](#)

[Performing a manual failover](#)

[Checking for communications failure on a node](#)

Defining user exit programs

Define user exit programs to execute a set of tasks from the backup node at various stages of node failure detection and role switching.

The sample user exits located in **ICLUSTER/QACLSRC** demonstrate how to access the parameters that are passed to user exit programs.

Note: User exit programs must be compiled with the **Use adopted authority option** set to *NO. Otherwise, the user exit programs cannot be executed.

Configure the cluster to run user exits in conjunction with the following switchover events:

- Before the backup node performs its failover processing. This user exit is run immediately before the role switch occurs. Each group can call a different user exit when this happens. The user exit that you configure with this parameter will run on all of the group's active nodes.

For example, if you are performing a switchover, then this user exit runs on both the primary and backup nodes. For an automatic failover, this user exit will only run on the backup node because the primary node will have a status of *FAILED.

- After the backup node performs its failover processing. This user exit runs immediately after the role switch. Each group can call a different user exit when this happens. The user exit that you configure with this parameter will run on all of the group's active nodes.
- When the backup node places a message in its queue to track repeated communication failures. Each node in the cluster can call a different user exit when communication failures occur.

Role switch user exit parameters

When the backup node calls a user exit before or after performing a role switch, the following parameters are passed to the user exit.

Table: Role switch user exit parameters

Position	Description	Length and Type
1	The name of the group that the switch applies to.	10 characters
2	The state of the role switch. Possible values are *BEFORE and *AFTER, depending on whether the role switch has occurred.	10 characters
3	The node that the exit is running on. Possible values are *PRIMARY and *BACKUP for the primary and backup nodes.	10 characters
4	User-specified data from the EXITDATA property for the group.	256 characters

Message action user exit parameters

When a node calls a user exit while queuing node failure messages, the following parameters are passed to the user exit:

Table: Message action user exit parameters

Position	Description	Length and type
1	The name of the node reporting the node failure.	10 characters
2	The role of the node initiating the failure declaration. The value of this parameter is always *BACKUP for SwitchOver System.	10 characters
3	An event code that indicates why the user exit was called. The value of this parameter is always *MSG.	10 characters
4	The number of messages that have been placed in the queue for this node failure.	2 characters

Related information

[Node failure examples](#)

Detecting problems in processing a role switch

iCluster checks for message ID CSI9953 during group role switch processing. The roleswitch is terminated when message **CSI9953** is sent by a role switch user exit.

In role switch user exit program, use the following command:

```
SNDPGMMMSG MSGID(CSI9953) MSGF(*LIBL/HAMSGF) TOPGMQ(*SAME)
```

After problems are detected during role switch processing, iCluster stops processing a role switch and declares that the role switch did not complete successfully. The following message is placed into the *CLUSTER portion of the event log: **CSI1357 (role switch for group &1 could not be completed)**.

Related information

[Node failure examples](#)

Node failure examples

Review the actions that iCluster performs when a roleswitch is performed or a node failure is detected is determined by the group and node parameters.

The following table includes the group level settings and the resulting actions:

Table: Group level settings

Setting	Keyword and value	Meaning
Do roleswitch at failover	Manual failover example: ROLESWITCH(*NO)	After the primary node status is changed to *FAILED, replication is suspended and the group waits for an administrator to either repair the problem or initiate a manual failover.
	Automatic failover example: ROLESWITCH(*YES)	After the primary node status is changed to *FAILED, the backup node automatically takes over as the group's primary node.
Check journaling at roleswitch	CHKJRN(*NO)	<p>It is assumed that each object that requires journaling is already journaled on the original backup node when the role switch takes place.</p> <p>Using this option is significantly faster than CHKJRN(*YES). However, all objects that require journaling must be journaled on the backup node prior to any node failures in order to use this option successfully.</p>
Failover message queue name	MSGQUEUE(QUEUES/ROLECHG)	Messages generated by the failover are placed in the ROLECHG message queue on the node that is declaring the failure.
User exit before roleswitch	BSWUSREXIT(APPS/PREEXIT)	The PREEXIT program runs on the current primary and backup nodes (if active) before the role switch is done.
User exit after roleswitch	ASWUSREXIT(APPS/POSTEXIT)	The POSTEXIT program runs on the current primary and backup nodes (if active) after the role switch is done.

The following table includes the node level settings and the resulting actions:

Table: Node Level Settings

Setting	Keyword and value	Meaning
Check primary link	CHKPRIMLNK(*YES)	The other nodes poll the primary IP address of the node.
Link check frequency (seconds)	LNKCHKFRQ(120)	The other nodes poll the node every 2 minutes (120 seconds).
Link check reply timeout (seconds)	LNKCHKRTO(60)	The other nodes wait for one minute (60 seconds) before considering a node poll to

Setting	Keyword and value	Meaning
		be a failure.
Maximum link check retries	LNKCHKTRY(3)	The other nodes allow three consecutive polling failures before changing the status of the node to *FAILED.
Message action user exit	MSGUSREXIT(APPS/MSGEXIT)	After failing another node, the node initiating the failure declaration calls the MSGEXIT user exit.
Message queue name	MSGQUEUE(QUEUES/NODEFAIL)	After failing another node, the node initiating the failure declaration places a message in the NODEFAIL message queue.
Wait time after action (minutes)	MSGACTWAIT(1)	The node initiating the failure declaration places subsequent messages in the NODEFAIL message queue every minute. The node also calls MSGEXIT once for every queued message.
Number of message actions	NUMMSGACT(5)	Five messages are placed in the NODEFAIL message queue, if a node fails. The node initiating the failure declaration calls MSGEXIT once for every queued message.

Related information

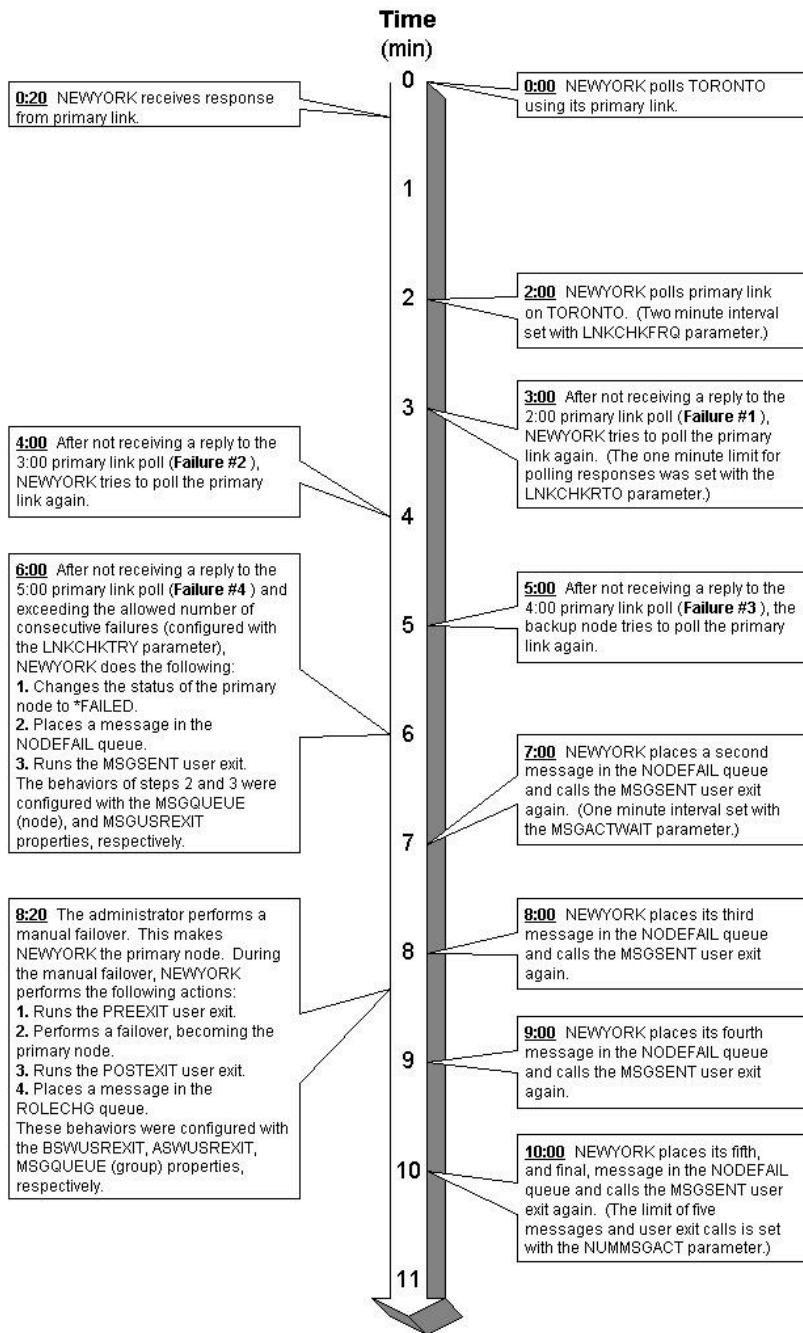
[Detecting node failures](#)

[Performing a manual failover](#)

Manual failover example

When iCluster is configured for manual failover, replication is suspended and the group waits for an administrator to either repair the problem or initiate a manual failover after the primary node status is changed to *FAILED.

The following figure shows a timeline for a group that replicates data from a node named TORONTO (primary) to a node named NEWYORK (backup), where the primary node becomes unavailable in a group configured for manual failover. The timeline shows the sequence of events from the backup node's perspective. The backup node performs all operations in the event of a failover.

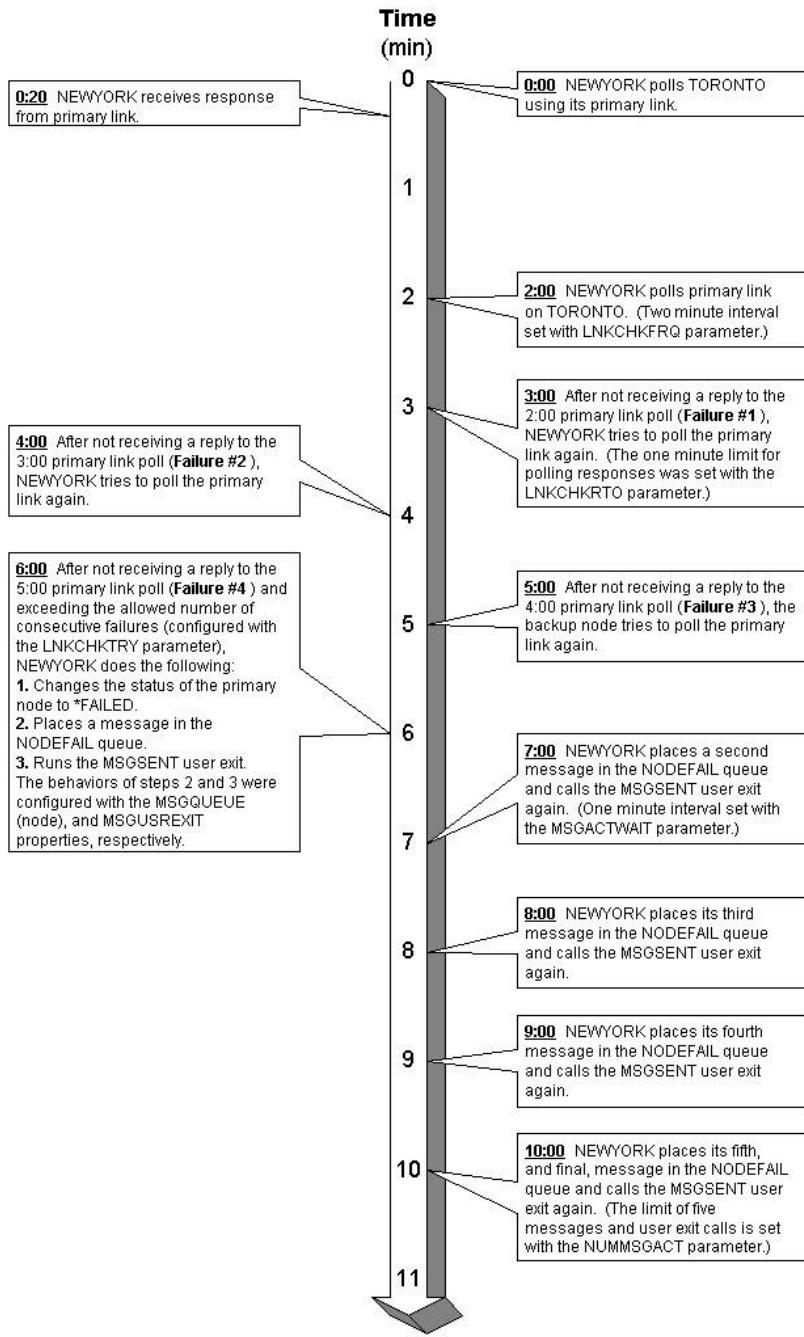


In this example, the administrator has chosen to perform a manual failover at 8:20. Alternatively, the administrator could have tried to repair TORONTO and restart the group.

Automatic failover example

When iCluster is configured for automatic failover, the backup node automatically takes over as the group's primary node after the primary node status is changed to *FAILED.

The figure below shows a timeline for an automatic failover. The configuration in this example is the same as the one described for a manual failover, except that the **ROLESWITCH** parameter is set to *YES. This setting specifies that the backup node must automatically failover to the primary node after changing its status to *FAILED.



Performing a manual failover

When a group is configured to not perform an automatic failover (the group's **ROLESWITCH** parameter is *NO), perform a manual failover after another node changes the state of the primary node to *FAILED and suspends replication.

At this point, you have the following options:

- Repair the problem on the primary node, and then resume replication for the group.
- Perform a manual failover: switch the roles of the primary and backup nodes for the group.

Performing a manual failover after the primary node fails

Perform a manual roleswitch for a group whose **ROLESWITCH** parameter is *NO and whose primary node has failed.

1. To exchange the roles of the primary and backup nodes, run the following command on the backup node:
`DMCHGROLE GROUP(<group name>)`
2. Repair the failed primary node.
3. To start the failed primary node, enter the following command:
`DMSTRNODE NODE(<node name>)`
4. To restart replication, enter the following command:
`DMSTRGRP GROUP(<group name>) USEMARKED(*YES)`

Note: Replication from the backup node to the primary node starts because you changed the role of the group's nodes in step 1. You must specify the USEMARKED(*YES) property to ensure replication includes all changes that occurred since the DMCHGROLE (iCluster Change Primary Node) command was issued.

Related information

[Recovering from failovers](#)

[DMCHGROLE \(Change Primary command\)](#)

[DMSTRNODE \(Start Node command\)](#)

[DMSTRGRP \(Start Group command\)](#)

Recovering from failovers

Depending on the value of the group's **ROLESWITCH** parameter, use different options for recovery.

If you have configured a group to perform an automatic failover (the group's **ROLESWITCH** parameter is *YES), perform a switchover to restore the group to resume replication in the original direction. Before performing a switchover, both nodes must have a state of *ACTIVE.

If you have configured a group to not perform an automatic failover (the group's **ROLESWITCH** parameter is *NO), perform a manual failover after the backup node changes the state of the primary node to *FAILED and suspends replication.

You can either repair the problem on the primary node and then resume the group, or perform a manual switchover and switch the roles of the primary and backup nodes for the group.

Related information

[Performing a manual failover](#)

Restoring a group whose primary node has failed

Restore a group whose **ROLESWITCH** parameter is *NO so that the group can resume replication from the current primary node to the current backup node after the primary node has failed.

1. Repair the primary node.
2. To start the failed primary node, enter the following command on the current primary node:
`DMSTRNODE NODE(<node name>)`
3. To restart replication, enter the following command:

```
DMSTRGRP GROUP(<group name>)
```

- Note:** You must specify the USEMARKED(*NO) property to ensure that replication restarts from the last position replicated before the primary node failed.

Testing switchover for a single group

Perform regular switchover verification testing to ensure that operational procedures are correctly defined and that the backup system is prepared to assume the role of the production system. The group is ready for a switchover when there is no latency, and no suspended or out-of-sync objects.

1. To verify that the group is ready for a switchover, complete one of the following actions:
 - In the Full Cluster Status Monitor, verify that the roleswitch readiness flag in the **R** column is **Y**.
 - Run the WRKCSMON (Work with Full Cluster Monitor) command and check for excessive latency, suspended objects, and out of sync objects. If none of the values exceed your requirements for switchover, the group is ready for a switchover.
 - Run the DMSWTCHRDY (iCluster Switchover Ready) command.
 - Call the RTVSWCHRDY (Retrieve Switchover Ready Status) command from a CL program.

If you determine that the group is ready for switchover, proceed to step 2 and the following steps. Otherwise, repeat step 1 until the group is ready for switchover.

2. On the primary node, end all external application activity.
3. To end the group, run the following command:

```
DMENDGRP GROUP(<group name>)
```
4. After the group is fully ended, initiate the switchover by entering the following command:

```
DMCHGROLE GROUP(<group name>)
```
5. To restart replication after the switchover processing is complete, run the DMSTRGRP (iCluster Start Group) command and specify ***YES** for the **Start replication apply jobs** parameter. For example:

```
DMSTRGRP GROUP(<group_name>) USEMARKED(*YES) STRAPY(*YES)
```

6. Run the WRKCSMON (Work with Full Cluster Monitor) command:

```
WRKCSMON
```

The status **AA** in the **PB** columns on the **Real Time Overall Latency** screen confirms that the switch has completed.
7. To verify that the roles of the primary and backup nodes have switched, run the DMWRKGRP (Work with Groups) command on either node and use Option 5.
8. On the backup node, view the event log to verify that the primary and backup nodes of the group were switched.
For example, a successful switchover of the PAYROLL group generates the following message: **OMI3190 20 DMMRKPOS completed successfully for group PAYROLL.**
9. Sign on to the new primary node.
10. Make changes to the data on the new primary node.
11. Sign on to the backup node to verify that the changes exist on both nodes.
12. Confirm that your environments, including external application environments, are identical to environments that existed on the original primary node.

Related information

- [**WRKCSMON \(Work with Full Cluster Monitor command\)**](#)
- [**DMSWTCHRDY \(Switchover Ready command\)**](#)
- [**RTVSWCHRDY \(Retrieve Switchover Ready Status command\)**](#)
- [**DMENDGRP \(End Group command\)**](#)
- [**DMCHGROLE \(Change Primary command\)**](#)
- [**DMSTRGRP \(Start Group command\)**](#)

Changing an existing cluster to use SwitchOver System

Change the failover mechanism for an existing cluster from Cluster Resource Services to SwitchOver System.

1. Enter the following command on the node that currently owns the iCluster metadata: DMWRKNODE
 2. Press Enter.
 3. On the **Work with Nodes** screen, ensure that all nodes have a status of *ACTIVE.
 4. Press Shift + F10.
 5. Enter the following command: DMWRKGRP
 6. Press Enter.
 7. On the **Work with iCluster Groups** screen, ensure that all groups have a replication status of *INACTIVE. If necessary, stop any active groups.
 8. Press Shift + F10.
 9. Type 30 at the command line. Press Enter.
 10. Press PgDn repeatedly to locate the **Use Cluster Resource Services** field.
 11. Change the **Use Cluster Resource Services** parameter to *NO. Press Enter.
 12. Change the default switchover settings for each node using the DMCHGNODE (Change Node) command to ensure they meet your recovery needs.
 13. On the **Work with Cluster Nodes** screen, restart all of the nodes in the cluster.
 14. On the **Work with iCluster Groups** screen, restart all of the groups.
- After you perform these tasks, you can no longer use DMSETPRIM (iCluster Prepare Primary Node) to perform a switchover. Instead, use the DMCHGROLE (iCluster Change Primary Node) command.

Related information

- [**DMWRKNODE \(Work with Nodes command\)**](#)
- [**DMWRKGRP \(Work with Groups command\)**](#)
- [**DMCHGNODE \(Change Node command\)**](#)
- [**DMCHGROLE \(Change Primary command\)**](#)

Cluster limits for SwitchOver System

If your cluster uses the SwitchOver System failover mechanism, then each node cannot have more than 200 links.

A link is created by having a node be either the primary or backup node in a group. Local loopback groups count as two links for a node.

When performing role switching with triggers enabled, the **QAQQINI** file (which contains system triggers) is skipped.

Cluster Resource Services failover mechanism

iCluster installations that do not use SwitchOver System use Cluster Resource Services for switchover and failover mechanisms.

Cluster limits for Cluster Resource Services

If your cluster uses Cluster Resource Services as its failover mechanism, then you cannot have more than 128 nodes in a cluster.

Related information

[SwitchOver System](#)

Clustering overview and concepts

A cluster consists of a networked collection of systems (or nodes) that work together to provide seamless IBM i operations. The underlying framework of iCluster is provided in the IBM i operating system.

iCluster builds on Cluster Resource Services for high availability and switchover capabilities.

The following terms describe clustering, high availability, and disaster recovery concepts.

Cluster

A cluster consists of a set of nodes that are configured to work together in order to provide continuous, uninterrupted operations. A cluster of nodes is capable of recovering from anticipated or unexpected disruptions in the normal flow of operations by moving processing activities from one node in the cluster to another. Therefore, communications must exist between all nodes in the cluster.

Cluster partition

The detection of a heartbeat timeout with no detection of a distress message.

Cluster Resource Services

A set of IBM i operating system functions that allow for cluster operations and implementations.

Cluster Resource Group (CRG)

A CRG specifies a recovery domain that defines a group of nodes and the role of each node in the group. There are three kinds of CRGs: data, applications, and devices.

Distress message

A message that is broadcast to all known nodes in the cluster subnet. The distress message indicates that a failure situation has been detected by a cluster node. The node ends clustering locally.

Failover

When a primary node automatically switches over to a backup node because of a system failure. A failover is generally unplanned.

Regardless of whether a failover or switchover occurs, the Cluster Resource Services failover mechanism is used to switch operations to a backup node when the primary node is unable to provide business services.

Heartbeat

The mechanism within the cluster that verifies that every active node in the cluster is responding to signals across one or two networks.

Node

A node is any server that is a member of a cluster. The iCluster node name is associated with one or more IP

addresses that represent a server. iCluster communications makes use of the TCP/IP protocol suite to provide the communications path between each node in the cluster.

Node failure

The detection of a distress message that is received from a failing cluster node before a heartbeat or message timeout is detected.

Recovery domain

The ordered list of nodes in a CRG from the primary node to the first backup node, to the second backup node, and so on.

Resilient devices

A resilient device is a physical resource that is represented by a configuration object, such as a device description, that is accessible from more than one node in a cluster.

Switchover

When a user manually switches the role of a primary node over to the role of a backup node. A switchover is a planned procedure.

Takeover IP addresses

A takeover IP address allows multiple nodes in the recovery domain to use the same IP address at different times. It facilitates a transparent switchover in a TCP/IP environment.

Related information

[Alertable messages for clustering](#)

iCluster and Cluster Resource Services

The cluster definition is maintained by both iCluster and Cluster Resource Services.

There are two parts to the cluster definition:

- Low-level definition maintained by Cluster Resource Services. The Cluster Resource Services definition includes the status of each node of the cluster and status of each cluster group.
- High-level definition maintained by iCluster. The iCluster portion of the definition includes the object specifiers and the replication status of replication groups.

iCluster groups and role switching

When Cluster Resource Services determines that a node in the cluster has failed, it puts the node into *FAILED status on the other nodes of the cluster. If a group in *ACTIVE status has the failed node as its primary node, the group might be switched over at the operating system level. The previous backup node becomes the primary node and the previous primary node becomes the backup node in the view of Cluster Resource Services.

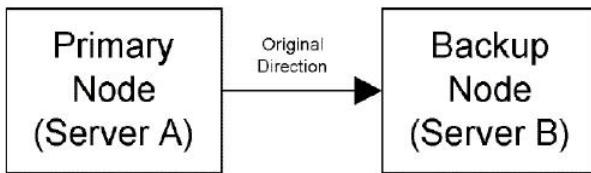
The value of the **ROLESWITCH** parameter of the DMADDGRP (Add Group) and DMCHGGRP (Change Group) commands determines whether the group switches over at the operating system level.



Note: Inactive groups are not switched over even when their primary node is the one that failed.

In the following diagram, the terms "server A" and "server B" refer to physical machines.

Figure: Typical Node Configuration



The terms "primary node" and "backup node" refer to roles that a physical machine can adopt in a Cluster Resource Group (CRG). Server A can be either the primary node or the backup node of a group. In this example, server A is the primary node and server B is the backup node before the failover happens. After the failover happens, server B is the primary node and server A is the backup node in the view of Cluster Resource Services.

Related information

[ROLESWITCH parameter](#)

[DMCHGROLE \(Change Primary command\)](#)

[DMADDGRP \(Add Group command\)](#)

[DMCHGGRP \(Change Group command\)](#)

ROLESWITCH parameter

Understand the **ROLESWITCH** parameter, which controls the behavior of the cluster in a primary node failure and allows you to decide whether to switch roles for a node.

Each replication group has a **ROLESWITCH** parameter which can be set to *YES or *NO.

ROLESWITCH *NO

This setting lets you decide if a role switch is necessary after a failover, and whether you proceed with a role switch or continue with your current configuration. When a failure of the group's primary node (for example, server A) is declared by Cluster Resource Services, iCluster will not prepare the backup node (server B) to take over as the primary node. Replication will not start up automatically when server A is active again in the cluster.

Note: After the failure, server A will still be the primary node and server B still the backup node, which is the same as prior to the failure.

*NO is the default setting for the **ROLESWITCH** parameter. With this setting, the failed primary node retains its role and resumes its functions as the primary node when the failed node becomes active in the cluster. To resume the replication in the same direction as before the primary node fails, use the DMSTRGRP (iCluster Start Group) command to start the group.

To do a role switch of the primary node and backup node, use the DMCHGROLE (iCluster Change Primary Node) command to change the primary node to be the backup and the backup node to be the primary, before you start the group with the DMSTRGP command.

For minor system failures with a minimum amount of down time, this is probably the optimal situation.

ROLESWITCH *YES

This setting automatically changes the role of the backup node in the replication group to the primary node when a failover occurs. When a failure of the group's primary node (for example, server A) is declared by Cluster Resource Services, the user exit programs that are specified through the **BSWUSREXIT** and **ASWUSREXIT** parameters in the DMADDGRP (Add Group) are called automatically and the backup node (server B) is prepared to take over as the primary node. After server A is again active in the cluster, replication starts from server B, which is now the primary node, to server A, which is now the backup node.

The **ROLESWITCH** parameter is defined at the group level. User exits are not required or advised, for each group. Set the **ROLESWITCH** parameter to *YES for a thoroughly defined and tested switch plan to prepare the

backup node to take over as the primary node. This often involves, but is not limited to, enabling user profiles, varying on devices, and changing TCP/IP configurations. These steps would have to be reversed and could be time-consuming should a true failure not have occurred.

iCluster does not allow the **ROLESWITCH** parameter to be used for groups replicating from *SYSBAS on the primary node to an iASP on the backup node, or from an iASP on the primary node to *SYSBAS on the backup node.

Related information

- [DMSTRGRP \(Start Group command\)](#)
- [DMCHGROLE \(Change Primary command\)](#)

Preparing the cluster to handle switchovers and failovers

Update the value of the **ROLESWITCH** parameter in the DMADDGRP (Add Group) or DMCHGGRP (Change Group) commands to control the behavior of the system on a failover.

Specify a user exit program for a group with the DMADDGRP (Add Group) or DMCHGGRP (Change Group) commands. The user exit program is called when the role of the node is changed within the group.

User exits for a group can be called before or after the role change. If a failover occurs and the **ROLESWITCH** parameter for the group is *YES, the user exits are called automatically when the failure is detected.

Related information

- [Clustering overview and concepts](#)
- [Passing arguments to role switch user exit programs](#)
- [DMADDGRP \(Add Group command\)](#)
- [DMCHGGRP \(Change Group command\)](#)

Journal entry processing

Change your journal cleanup processes and schedule in the before role switch user exit program.

For example, you might have configured your target receivers to quickly clean up on the backup node. If your group is configured for remote journaling, you need the receivers created as a result of the failover switch to resynchronize your original primary node. Change the schedule for journal receiver cleanup in the before role switch user exit program.

Failover message queue

In iCluster, specify a failover message queue to receive messages that are generated when a failover occurs. You can only specify a failover message queue for replication groups.

Related information

- [Failover and switchover mechanisms](#)
- [Cluster Resource Services failover mechanism](#)

Line controller heartbeat

Understand the communication line controller, which detects and records line errors. Within the line description are two parameters that determine when the system operator is notified of line errors.

The default settings specify that if an error count limit of two is reached within the last five-minute time interval, then the system operator is notified with a message that requires a response as to whether to continue retrying the line (G or R) or to no longer attempt retries (C).

```
WRKLIND <Name of line description for TCP/IP line>
```

Table: Recovery Limits

Count Limit	2 (default)
Time interval	5 (default)

For example, a LAN cable pull results in one line error. Every 10–15 seconds a retry is attempted, which results in additional errors. Therefore, with the default settings an operator message is received within 20–30 seconds. This manual intervention should be eliminated so that the cluster heartbeat timeouts can take control.

You can adjust these parameters to allow for a longer interval before the system message is generated. This creates a larger window in which Cluster Resource Services can recover from line outages. Because increasing these parameters would mean that the system overall would detect line errors more slowly, the values that you specify depend on what is acceptable for your system.

Passing arguments to role switch user exit programs

When the backup node calls a user exit program, before or after performing a role switch, specific parameters are passed to the user exit.

If you define a user exit program that is called when a failover or switchover occurs, the following arguments are passed to the program:

Table: Role switch user exit parameters

Position	Description	Length and type
1	The name of the group that the switch applies to.	10 characters
2	The state of the role switch. The possible values are *BEFORE and *AFTER, depending on whether the role switch has occurred.	10 characters
3	The node that the exit is running on. The possible values are *PRIMARY and *BACKUP for the primary and backup nodes.	10 characters
4	User-specified data from the EXITDATA property for the group.	256 characters

! Notes:

- The order of the arguments passed to the user exit program is very important, and they are passed in the order as they appear.
- User exit programs must be compiled with the **Use adopted authority** option set to *NO. Otherwise, the user exit programs cannot be executed.

FAILED state overview

Understand the FAILED state, how to detect it, and how to specify the method of recovery depending on certain conditions.

If a node fails and sends a distress signal, then Cluster Resource Services reports the node failure to the other nodes in the cluster and initiates a failover.

FAILED state scenarios

The following scenarios result in a detected node failure if Cluster Resource Services detects a distress message from the failing cluster node.

All of the active resources that are managed by Cluster Resource Services on that node go through the failover process. This scenario includes all application Cluster Resource Groups (CRGs), data CRGs, and device CRGs.

Loss of utility power scenarios

- The internal Battery Backup Unit (BBU) is installed and the internal battery timer times out.
- The internal BBU is installed and an internal battery low signal is detected.
- UPS is installed, a UPS Utility Failure signal is received by SPCN, and a timer set by the QUPSDLYTIM system value times out.
- UPS is installed and both UPS Utility Failure and UPS Battery Low signals are received by SPCN.

Controlled shutdown scenarios

- The Initial Program Load (white) button is pushed to power down the system.
- **PWRDWNSYS *IMMED** command is issued.
- **PWRDWNSYS *CNTRLD** command is issued and the controlled function expires.
- **ENDSBS *ALL** command with:
 - *IMMED or *CNTRLD where a time limit on the controlled function expires.
- **ENDSYS** command on controlling subsystem with:
 - *IMMED or *CNTRLD where a time limit on the controlled function expires.
- **ENDSBS QSYSWRK** command with:
 - *IMMED or *CNTRLD where a time limit on the controlled function expires.Where Cluster Resource Services and TCP/IP jobs reside.
- **ENDTCP** command with:
 - *IMMED or *CNTRLD where a time limit on the controlled function expires.
- **ENDJOB QCSTCTL** command with:
 - *IMMED or *CNTRLD where a time limit on the controlled function expires.
- **ENDJOB QCSTCRGM** command with:
 - *IMMED or *CNTRLD where a time limit on the controlled function expires.

FAILED state scenarios

The following scenarios result in a detected node failure if Cluster Resource Services detects a distress message from the failing cluster node.

All of the active resources that are managed by Cluster Resource Services on that node go through the failover process. This scenario includes all application Cluster Resource Groups (CRGs), data CRGs, and device CRGs.

Loss of utility power scenarios

- Internal Battery Backup Unit (BBU) installed and the internal battery timer times out.
- Internal BBU installed and internal battery low signal detected.
- UPS installed, UPS Utility Failure signal received by SPCN, and a timer set by system value QUPSDLYTIM system value times out.
- UPS installed and both UPS Utility Failure and UPS Battery Low signals received by SPCN.

Controlled shutdown scenarios

- Initial Program Load (white) button pushed to power down the system.
- PWRDWNSYS *IMMED command issued.
- PWRDWNSYS *CNTRLD command issued and controlled function expires.
- ENDSBS *ALL:
 - *IMMED or *CNTRLD where a time limit on the controlled function expires.
- ENDSYS on controlling subsystem:
 - *IMMED or *CNTRLD where a time limit on the controlled function expires.
- ENDSBS QSYSWRK:
 - *IMMED or *CNTRLD where a time limit on the controlled function expires.
- Where Cluster Resource Services and TCP/IP jobs reside.
- ENDTCP:
 - *IMMED or *CNTRLD where a time limit on the controlled function expires.
- ENDJOB QCSTCTL is issued:
 - *IMMED or *CNTRLD where a time limit on the controlled function expires.
- ENDJOB QCSTCRGM is issued:
 - *IMMED or *CNTRLD where a time limit on the controlled function expires.

Detecting a FAILED state

Detect whether a node is in a FAILED state from the DMWRKNODE (Work with Nodes) command.

From the iCluster command line, issue the **DMWRKNODE** command. The status of the node is displayed on the screen. If it displays as *FAILED, then the node is in a FAILED state.

After a node is in a FAILED state, you must recover the node.

Related information

[Recovery paths for a FAILED state](#)

[DMWRKNODE \(Work with Nodes command\)](#)

Group failover

Understand the failover processes for an active replication group whose primary node status is *FAILED.

An active replication group whose primary node goes into *FAILED state undergoes failover in one of two ways, depending on the group **ROLESWITCH** parameter. Inactive groups do not undergo failover.

- If the group **ROLESWITCH** parameter is set to *YES, the group backup node become the group primary node. The group new primary node is prepared so that the group is ready to begin replication from the new primary node (the previous backup node) when a backup node for the group becomes available.
- If the group **ROLESWITCH** parameter is set to *NO (the default), the group primary node and backup node remain the same after the failure of the primary node.

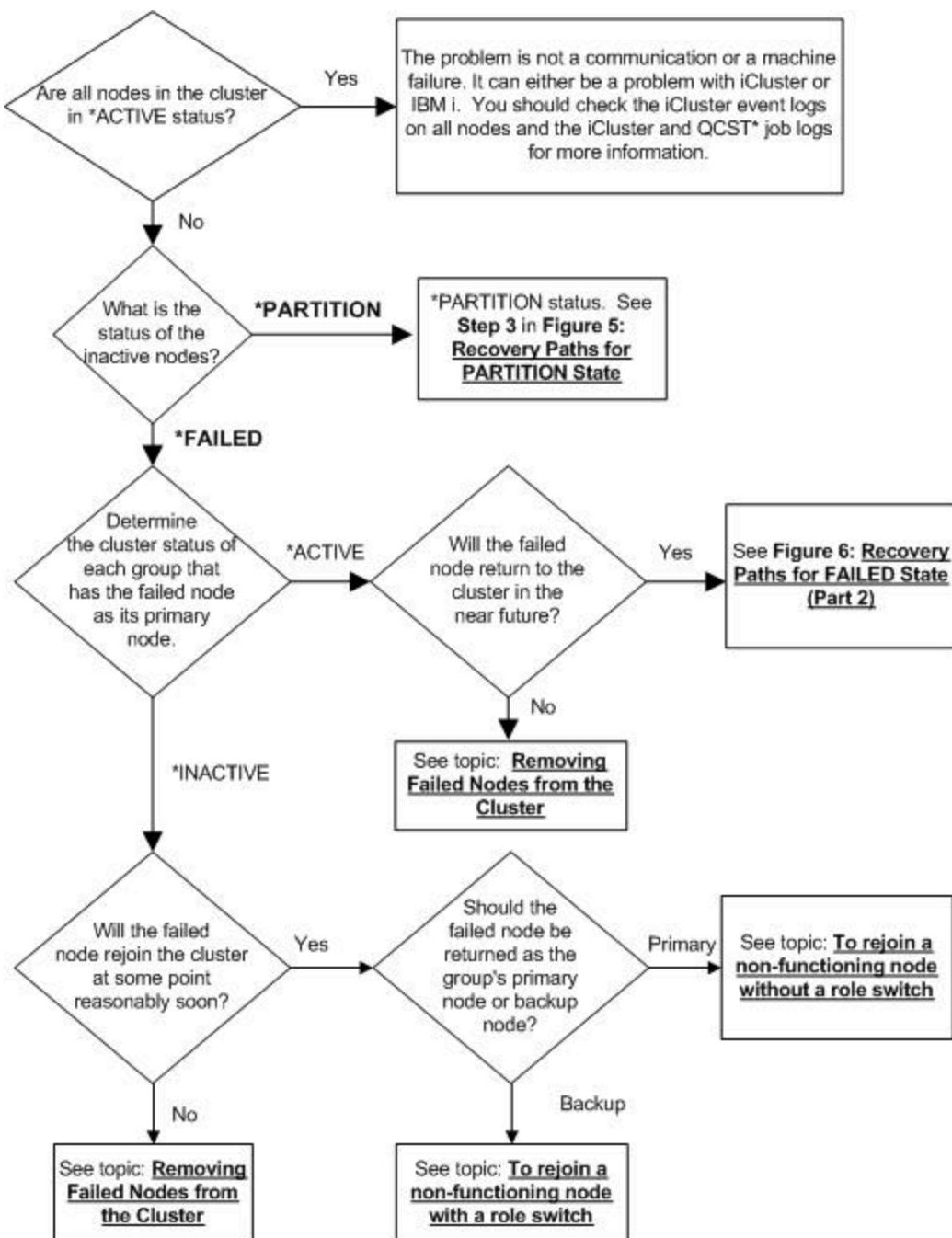
Set the **ROLESWITCH** parameter to *NO for the option to continue with the current configuration. The failed primary node retains its role and resume its functions as the primary node after it recovers and rejoins the cluster.

Recovery paths for a FAILED state

Based on a number of conditions, different recovery paths are available.

The following diagram displays the different options:

Figure: Recovery Paths for Failed State (Part 1)



The Cluster Resource Services jobs that run in the QSYSWRK subsystem on an active cluster node are listed in the following table:

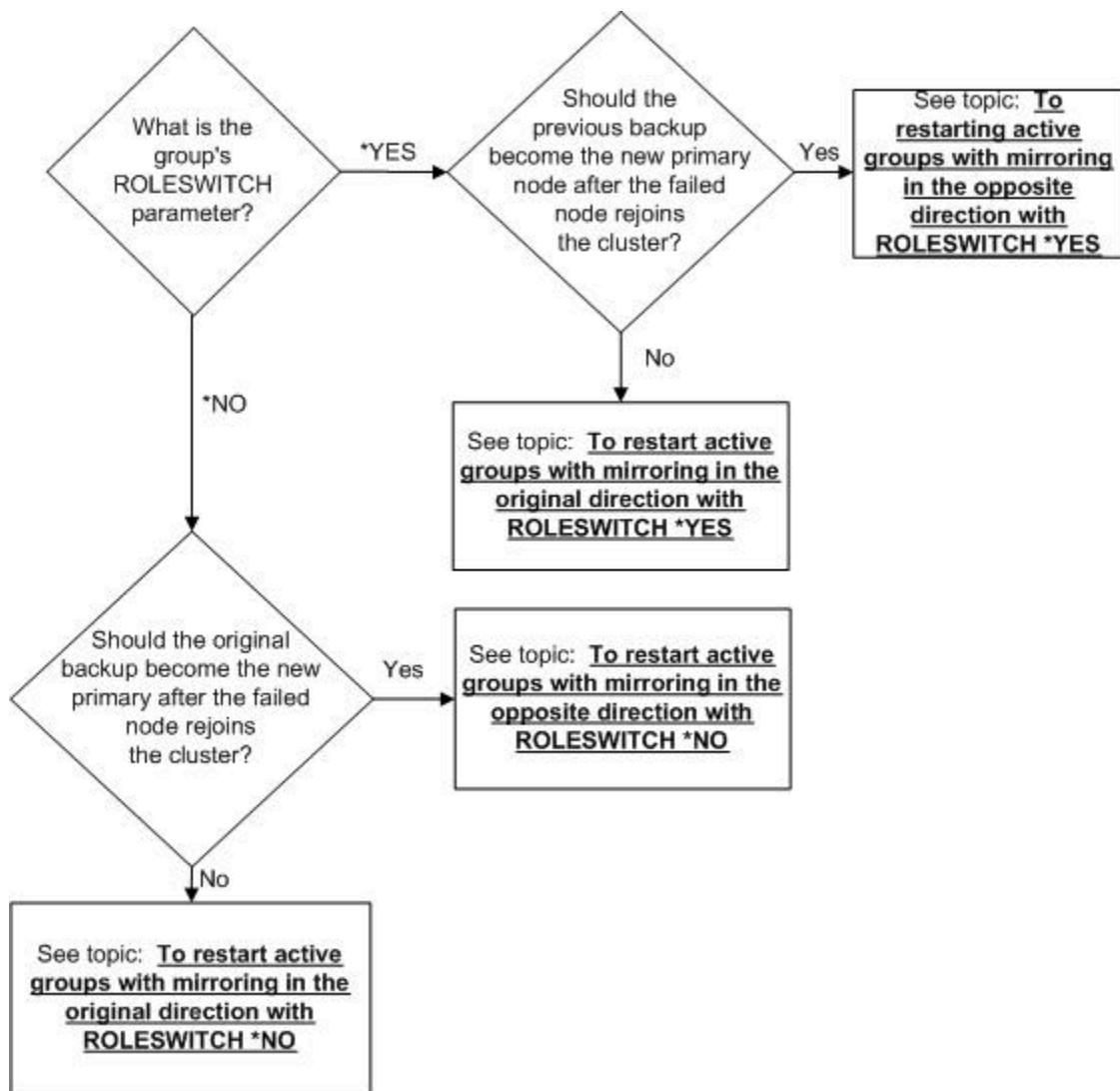
Table: Cluster Resource Services Jobs in the QSYSWRK Subsystem

Job Name	Function
QCSTCRGM	PGM-QCSTCRGMA
QCSTCTL	PGM-QCSTCCJOB
QCSTINETD	Not

Job Name	Function
	available.
DM_INTGRP	PGM-QCSTCRGJOB
All other jobs whose function name is PGM-QCSTCRGJOB. The job names are the names of *CRG objects (cluster resource groups) in the cluster.	

To view the job logs if they end, issue the IBM i command WRKSPLF QSYS and search for either the job name or the job function name.

Figure: Recovery Paths for FAILED State (Part 2)



Any underlined text in the figures indicates a topic in this document that you should read for information about how to proceed.

Related information

[Restarting active groups with mirroring in the opposite direction with ROLESWITCH *NO](#)

[Restarting active groups with mirroring in the opposite direction with ROLESWITCH *YES](#)

[Restarting active groups with mirroring in the original direction with ROLESWITCH *NO](#)

[Restarting active groups with mirroring in the original direction with ROLESWITCH *YES](#)

[Rejoin a non-functioning node with a role switch](#)

[Rejoin a non-functioning node without a role switch](#)

PARTITION state overview

Understand the PARTITION state, how to detect it, and how to specify the method of recovery depending on certain conditions.

Cluster Resource Services checks every three seconds for communication among the nodes in a cluster. If communication between two nodes is lost, then Cluster Resource Services attempts to reestablish communications. At this point, the node is in a PARTITION state.

In contrast, a FAILED state occurs when the primary node will soon no longer be operational, but can inform other nodes of this situation by issuing a distress signal. If the primary node of an active group enters a FAILED state, then the Cluster Resource Services initiates a failover for the group.

False and True PARTITIONS

If a node is in a PARTITION state, it can either be true or false.

True PARTITION

A true PARTITION state occurs when one node (node A) cannot communicate with another node (node B).

However, all nodes are functioning; only the communication among nodes is lost. In a PARTITION state, it is assumed that communication will return soon, and that the node will become active again.

Essentially, a PARTITION state is a waiting state, under the assumption that communication between nodes is down only temporarily. While the node is in a PARTITION state, replication does not take place. When communication between the partitioned nodes is restored, replication automatically resumes.

False PARTITION

A false PARTITION state occurs when communication among nodes is lost, but one of the nodes is no longer functioning. It is considered a false PARTITION because the node should really be in a FAILED state. However, due to the lack of communication, node A does not know that node B is no longer functioning, and node A is aware only that it cannot communicate with node B

PARTITION state scenarios

The Cluster Resource Services detects a PARTITION state when a node repeatedly fails to respond to the heartbeat. By default, the PARTITION state is detected within 42 seconds.

The following scenarios will result in a detected cluster partition if Cluster Resource Services is active on the node detecting the cluster partition. The nodes may or may not be operational but the cluster will stop all further operations and inform the cluster operator that a partition has been detected by an alertable message in the QSYSOPR message queue.

False PARTITION scenarios

CEC hardware failures:

- SRC B6xx Terminate Immediate
- SRC B9xx XPF originated Terminate Immediate

Loss of Utility power scenarios:

- Neither Battery Backup Unit nor UPS installed
- Remote EPO signal active

Power down from control panel (blue button):

- Option 3 restart
- Option 8

Operating System software hard machine check:

- Jobs are not ended and a mainstore dump is initiated.

True PARTITION scenarios

Communications hardware failure:

- On all IP interface addresses defined for a node
- Comm adapter, line, hub or router failures
- **ENDTCPIFC** command issued on all IP interfaces addresses defined for cluster heartbeat

False and True PARTITIONS

If a node is in a PARTITION state, it can either be true or false.

True PARTITION

A true PARTITION (simply referred to as PARTITION) occurs when one node (node A) cannot communicate with another node (node B). However, all nodes are functioning; only the communication among nodes is lost. In a PARTITION state, it is assumed that communication will return soon, and that the node will become active again.

Essentially, a PARTITION state is a waiting state, under the assumption that communication between nodes is down only temporarily. While the node is in a PARTITION state, replication does not take place. When communication between the partitioned nodes is restored, replication automatically resumes.

False PARTITION

A false PARTITION state occurs when communication among nodes is lost, but one of the nodes is no longer functioning. It is considered a false PARTITION because the node should really be in a FAILED state, but due to the lack of communication, node A does not know that node B is no longer functioning; node A is aware only that it cannot communicate with node B.

Related information

- [**Recovery paths for a PARTITION state**](#)
- [**Alertable messages for clustering**](#)
- [**HAPNGTCP \(Ping Using TCP command\)**](#)

PARTITION state scenarios

The Cluster Resource Services detects a PARTITION state when a node repeatedly fails to respond to the heartbeat. By default, the PARTITION state is detected within 42 seconds.

The following scenarios will result in a detected cluster partition if Cluster Resource Services is active on the node detecting the cluster partition. The nodes may or may not be operational but the cluster will stop all further

operations and inform the cluster operator that a partition has been detected by an alertable message in the QSYSOPR message queue.

False PARTITION scenarios

CEC hardware failures:

- SRC B6xx Terminate Immediate.
- SRC B9xx XPF originated Terminate Immediate.

Loss of Utility power scenarios:

- Neither Battery Backup Unit nor UPS installed.
- Remote EPO signal active.

Power down from control panel (blue button):

- Option 3 restart.
- Option 8.

Operating System software hard machine check:

- Jobs are not ended and a mainstore dump is initiated.

True PARTITION scenarios

Communications hardware failure:

- On all IP interface addresses defined for a node.
- Comm adapter, line, hub or router failures.

ENDTCPIFC command issued:

- On all IP interfaces addresses defined for cluster heartbeat.

Detecting a PARTITION state

Detect whether a cluster is in a **PARTITION** state from the DMWRKNODE (Work with Nodes) command line.

From the iCluster command line, issue the **DMWRKNODE** command. The status of the node is displayed on the screen. If the status displays as ***PARTITION**, then the cluster is in a **PARTITION** state.

Related information

[**DMWRKNODE \(Work with Nodes command\)**](#)

Verifying a PARTITION state

To verify that your cluster is in a true ***PARTITION** state, issue the HAPNGTCP (Ping Using TCP) command from all active nodes in the cluster to the nodes that are listed in a ***PARTITION** state.

Enter the following parameters when issuing the **HAPNGTCP** command:

-
- IP address of the node
 - Port number of the node
 - iCluster product library on the node
 - Default replication job description of the node

The correct values for these parameters are the ones listed for the node by the **Display node** option (option 5 on the **Work with Nodes** screen).

If the **HAPNGTCP** command succeeds, your cluster is not in a ***PARTITION** state and the nodes should return to a ***ACTIVE** state within 15 minutes.

If the **HAPNGTCP** command fails, there are two possible explanations:

- If all the nodes in the cluster list the status as ***ACTIVE** with one or more of the other nodes in the cluster having a ***PARTITION** state, you have a true **PARTITION** state in the cluster.
- If the **HAPNGTCP** command fails and the node with ***PARTITION** status has actually failed, you have a false **PARTITION** state in the cluster.

Related information

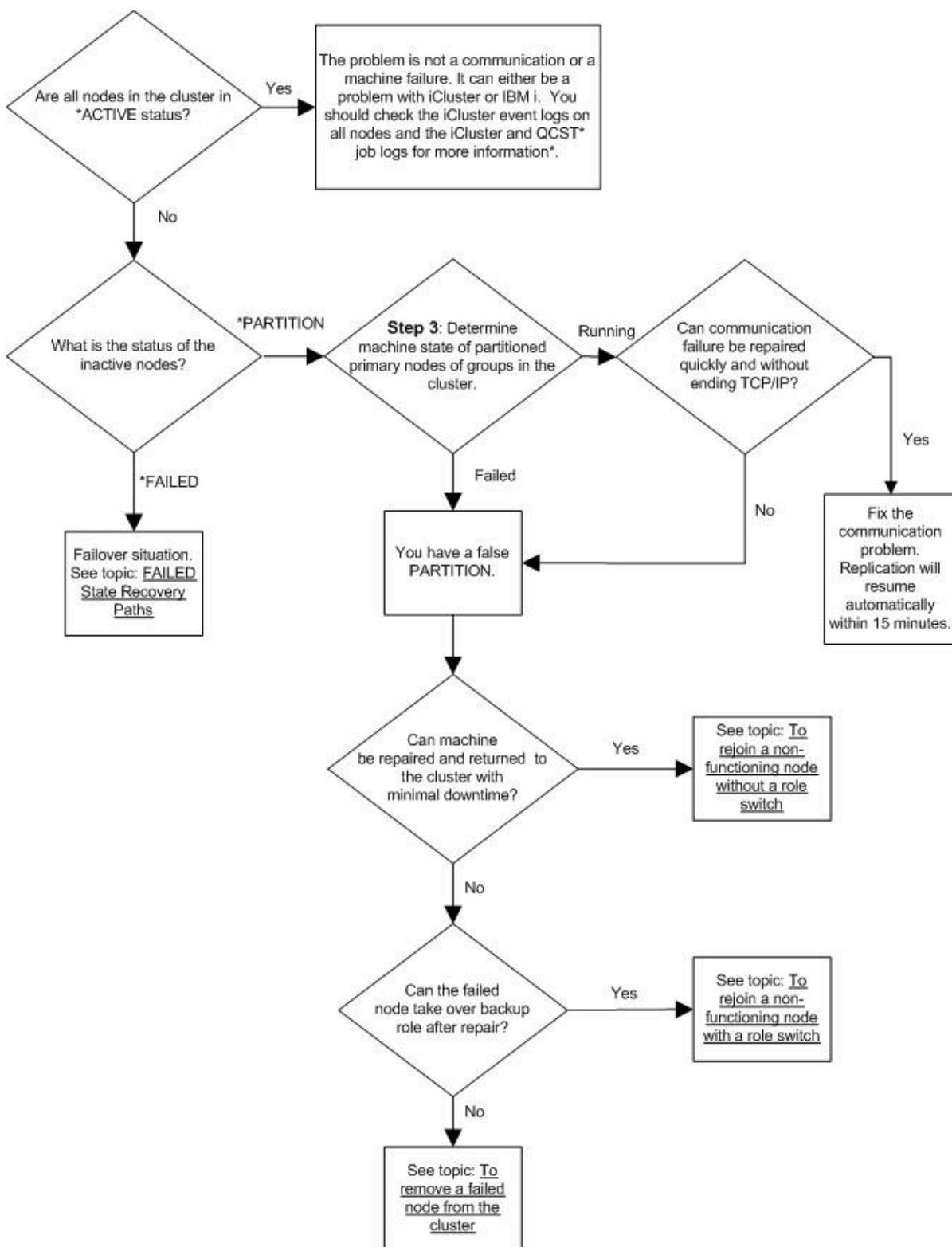
[**HAPNGTCP \(Ping Using TCP command\)**](#)

Recovery paths for a PARTITION state

Understand the recovery path for nodes that are in a **PARTITION** state depending on conditions, such as false or true **PARTITION**, whether the machine will be repaired quickly, and whether the failed node will become the backup node.

For example:

Figure: Recovery Paths for PARTITION State



- Note:** Underlined text in the diagram references a topic in this document that provides information about how to proceed.

Review the Cluster Resource Services jobs that run in the QSYSWRK subsystem on an active cluster node, which are listed in the following table.

Table: Cluster Resource Services Jobs in the QSYSWRK Subsystem

Job Name	Function
QCSTCRGM	PGM-QCSTCRGMA
QCSTCTL	PGM-QCSTCCJOB
QCSTINETD	Not available.
DM_INTGRP	PGM-QCSTCRGJOB
All other jobs whose function name is PGM-QCSTCRGJOB. The job names are the names of *CRG objects (cluster resource groups) in the cluster.	

To view the job logs if they end, issue the command `WRKSPLF QSYS` and search for either the job name or the job function name.

Related information

- [Rejoin a non-functioning node with a role switch](#)
- [Rejoin a non-functioning node without a role switch](#)

Restarting active groups

Restarting active groups with mirroring in the original direction with ROLESWITCH *NO

Restart an active group when the primary node has failed and you want to return the group's original direction of mirroring.

Perform the following procedure for a group whose cluster status is *ACTIVE and whose primary node is in the *FAILED state when all of the following points are true:

- The status of the non-active node is *FAILED.
 - The failed node will return to the cluster in the near future.
 - The group **ROLESWITCH** parameter is set to *NO.
 - The failed node will be the group's primary node when it rejoins the cluster.
1. Repair and restart the failed node.
 2. Make sure that TCP/IP and the *INETD TCP/IP server are active on the node.
 3. Make sure that the XDMCLUSTER subsystem and the DMCHATL job are running.
If not, use the STRSBS command to start the subsystem. The DMCHATL job should start automatically. If it does not, then enter the STRHATCP—Start TCP/IP Listener command. Specify dmcluster as the TCP/IP service name.
 4. Restart the node by running the DMREJOIN (iCluster Rejoin Cluster) command on the failed node or by issuing the DMSTRNODE (iCluster Start Node) command on another active node in the cluster.

CAUTION: Do not issue the DMSTRNODE command on the failed node.

Replication for active groups with ROLESWITCH *NO and whose primary node failed do not restart automatically because the failover processing did not complete for these groups.

-
- Run the DMSTRGRP (iCluster Start Group) command to start the group whose primary node failed and rejoined the cluster.

Related information

[DMREJOIN \(Rejoin Cluster command\)](#)
[DMSTRNODE \(Start Node command\)](#)
[STRHATCP \(Start TCP/IP Listener command\)](#)
[DMSTRGRP \(Start Group command\)](#)

Restarting active groups with mirroring in the original direction with ROLESWITCH *YES

Restart active groups whose cluster status is *ACTIVE and whose primary node is in the *FAILED state.

Perform the following procedure for a group whose cluster status is *ACTIVE and whose primary node is in the *FAILED state when all of the following points are true:

- When the status of the non-active node is *FAILED.
- The failed node will return to the cluster in the near future.
- The group **ROLESWITCH** parameter is set to *YES.
- The previous backup node will not become the new primary node after the failed node rejoins the cluster.

- Repair and restart the failed node.
- Make sure that TCP/IP and the *INETD TCP/IP server are active on the node.
- Verify that the XDMCLUSTER subsystem and the DMCHATL job are running.
If not, use the STRSBS command to start the subsystem. The DMCHATL job should start automatically. If it does not, then enter the STRHATCP (Start TCP/IP Listener) command. Specify `dmcluster` as the TCP/IP service name.
- To restart the node, run the DMREJOIN (iCluster Rejoin Cluster) command on the failed node or the DMSTRNODE (iCluster Start Node) on another active node.

 **CAUTION:** Do not resume working on either node until Step 5 is completed.

- Run the DMSTRSWO (Switchover Group) command for the active groups whose primary node failed and rejoined the cluster.

Related information

[STRHATCP \(Start TCP/IP Listener command\)](#)
[DMREJOIN \(Rejoin Cluster command\)](#)
[DMSTRNODE \(Start Node command\)](#)
[DMSTRSWO \(Switchover Group command\)](#)

Restarting active groups with mirroring in the opposite direction with ROLESWITCH *NO

Restart a group whose cluster status is *ACTIVE.

Perform this procedure for a group whose cluster status is *ACTIVE and whose primary node is in the *FAILED state when all of the following points are true:

- When the status of the non-active node is *FAILED.
- The node will be returned to the cluster in the near future.

- The group **ROLESWITCH** parameter is set to *NO.
 - The previous backup will become the new primary node after the failed node rejoins the cluster.
1. Repair and restart the failed node.
 2. Make sure that TCP/IP and the *INETD TCP/IP server are active on the node.
 3. Verify that the XDMCLUSTER subsystem and the DMCHATL job are running.
If not, use the STRSBS command to start the subsystem. The DMCHATL job should start automatically. If it does not, then enter the STRHATCP (Start TCP/IP Listener) command. Specify `dmcluster` as the TCP/IP service name.
 4. To restart the node, run the DMREJOIN (iCluster Rejoin Cluster) command on the failed node or the DMSTRNODE (iCluster Start Node) on another active node.
 5. Use the DMCHGROLE (iCluster Change Primary Node) command to change the primary node that has just been restarted after the failure to be the backup and the old backup node to be the primary.
 6. Start the group using the DMSTRGRP (iCluster Start Group) command.

Related information

[STRHATCP \(Start TCP/IP Listener command\)](#)
[DMREJOIN \(Rejoin Cluster command\)](#)
[DMSTRNODE \(Start Node command\)](#)
[DMSTRGRP \(Start Group command\)](#)

Restarting active groups with mirroring in the opposite direction with ROLESWITCH *YES

Restart a group whose cluster status is *ACTIVE.

Perform these steps on a group whose cluster status is *ACTIVE and whose primary node has failed when all of the following points are true:

- When the status of the non-active node is *FAILED.
 - The failed node will return to the cluster in the near future.
 - The group **ROLESWITCH** parameter is set to *YES.
 - The original backup node will become the new primary node after the failed node rejoins the cluster.
1. Repair and restart the failed node.
 2. Make sure that TCP/IP and the *INETD TCP/IP server are active on the node.
 3. Verify that the XDMCLUSTER subsystem and the DMCHATL job are running.
If not, use the STRSBS command to start the subsystem. The DMCHATL job should start automatically. If it does not, then enter the STRHATCP (Start TCP/IP Listener) command. Specify `dmcluster` as the TCP/IP service name.
 4. To restart the node, run the DMREJOIN (iCluster Rejoin Cluster) command on the node that failed or the DMSTRNODE (iCluster Start Node) command on another active node.
The groups will be restarted automatically by Cluster Resource Services.

Related information

[STRHATCP \(Start TCP/IP Listener command\)](#)
[DMREJOIN \(Rejoin Cluster command\)](#)
[DMSTRNODE \(Start Node command\)](#)

Rejoining non-functioning nodes

Rejoin a non-functioning node with a role switch

With a role switch, recover a group whose primary node is in a specific state (for example, false PARTITION or FAILED).

Consider these scenarios:

Scenario 1 - False PARTITION

- The machine can be repaired and returned to the cluster with minimal downtime.
- The failed node will take over the backup role after it is repaired.

Scenario 2 - FAILED Node

- The cluster status of the group is *INACTIVE.
- The machine in question can be repaired and returned to the cluster with minimal downtime.
- The failed node will take over the backup role after the repair.

1. Repair the failed node (machine) and restart it.
2. Ensure that TCP/IP and the *INETD TCP/IP server are active on the machine
3. Ensure that the XDMCLUSTER subsystem and the DMCHATL job are running
To start the subsystem, use the STRSBS command. The DMCHATL job should start automatically. If the DMCHATL job does not start automatically, use the STRHATCP (Start TCP/IP Listener) command, and specify `dmcluster` for the TCP/IP service name.
4. To return the failed node to the cluster, run the DMSTRNODE (iCluster Start Node) command for the failed node from another active node.
You can also run the DMREJOIN (iCluster Rejoin Cluster) command on the failed node.

It can take up to 15 minutes for the status of the failed node to become *ACTIVE when viewed from another node if the node was previously in a *PARTITION state.

5. Run the DMCHGROLE (iCluster Change Primary Node) command for the groups whose primary node is the failed node.
This action will cause the backup node for the groups to become the primary node, and the primary node to become the backup node. After this process is completed, work can resume on the new primary nodes.
6. **Optional:** For scenario 1 only, you can restart the inactive groups for which you performed a role switch (see Step 5) using the DMSTRGRP (iCluster Start Group) command.
Verify that mirroring restarts properly.

 **CAUTION:** Do not perform Step 6 for Scenario 2 FAILED Node.

Related information

- [STRHATCP \(Start TCP/IP Listener command\)](#)
- [DMSTRNODE \(Start Node command\)](#)
- [DMREJOIN \(Rejoin Cluster command\)](#)
- [DMCHGROLE \(Change Primary command\)](#)
- [DMSTRGRP \(Start Group command\)](#)

Rejoin a non-functioning node without a role switch

Without a roleswitch, recover a group whose primary node is in a specific state (for example, false PARTITION or

FAILED).

Scenario 1 - False PARTITION

- The status of the non-active node is *PARTITION.
- The machine in question is not running.
- The communication failure cannot be fixed quickly and without ending TCP/IP.
- The machine in question can be repaired and returned to the cluster with minimal downtime.
- The failed node will retain the primary node after it rejoins the cluster.

Scenario 2 - FAILED Node

- The status of the non-active node is *FAILED.
- The cluster status of the group is *INACTIVE.
- The failed node will remain as the group's primary node after it rejoins the cluster.

1. Repair the failed node (machine) and restart it.
2. Make sure that TCP/IP and the *INETD TCP/IP server are active on the node.
3. Make sure that the XDMCLUSTER subsystem and the DMCHATL job are running.

If not, start the subsystem using the STRSBS command. The DMCHATL job should start automatically. If it does not, then you can start it by issuing the STRHATCP (Start TCP/IP Listener) command. Make sure that you specify "dmcluster" as the TCP/IP service name.

4. To return the failed node to the cluster, run the DMSTRNODE (iCluster Start Node) command for the failed node from another active node.

You can also run the DMREJOIN (iCluster Rejoin Cluster) command on the failed node.

It can take up to 15 minutes for the status of the failed node to become *ACTIVE when viewed from another node.

5. **Optional:** Restart the groups that were active before the failure and whose primary node is the failed node.
6. Verify that mirroring restarts correctly for active groups where the failed node is the backup node.

Related information

- [STRHATCP \(Start TCP/IP Listener command\)](#)
[DMSTRNODE \(Start Node command\)](#)
[DMREJOIN \(Rejoin Cluster command\)](#)

Removing a failed node from the cluster

In specific scenarios, remove the failed node from the cluster.

Scenario 1

- The status of the non-active node is *FAILED.
- The cluster status of each group that has the failed node as its primary node is *ACTIVE.
- The node will not be returned to the cluster in the near future.

Scenario 2

- The status of the non-active node is *FAILED.
- The cluster status of each group that has the failed node as its primary node is *INACTIVE.
- The node will not be returned to the cluster in the near future.

1. On an active node, enter the DMRNMNODE (Rename Node) command for the failed node. This command removes the failed node from the cluster, and causes an automatic role switch for groups whose primary node was the failed node.

Note: To restart replication for the groups whose primary or backup node was the failed node, you might need to add a backup node to the cluster and define the new node as the backup node for the groups that do not have a backup node.

After the groups have a backup node, restart replication by performing Step 2 - Step 4.

2. To end all active groups, issue the DMENDGRP (iCluster End Group) command.
3. Perform a save/restore or a refresh of replicated objects to the new backup node.

Note: If a save/restore of replicated objects was performed, enter the DMMRKPOS (Mark Journal Positions) command for the groups prior to starting the group. Start the group using the marked positions.

4. To restart replication, issue the DMSTRGRP (iCluster Start Group) command with the **USEMARKED** parameter set to *YES.

CAUTION: If the failed node that was removed from the cluster will be returned to the cluster at a later time, then delete the cluster from the node before you perform cluster operations on it. To delete the cluster from the node, issue the DMDLTCLSTR (Delete Cluster) command on the failed node only. Do not run the DMDLTCLSTR (Delete Cluster) command on any other node.

Related information

- [DMRMVNODERemove Node command](#)
- [DMENDGRPEnd Group command](#)
- [DMMRKPOSMark Journal Positions command](#)
- [DMSTRGRPSStart Group command](#)
- [DMDLTCLSTRDelete Cluster command](#)

Alertable messages for clustering

Cluster Resource Services generates several types of messages, alertable and informational.

The following alertable messages are generated by Cluster Resource Services:

Table: Alertable Messages

Message ID	Severity	Alert Option	Message Title
CPFBB20	40	*IMMED	Cluster partition detected for cluster [name] by cluster node [node name].
CPFBB06	40	*IMMED	Incoming request from cluster node &3 in cluster &2 rejected.
CPIBB10	50	*IMMED	Cluster resource group exit program &1 in library &2 on node &3 failed.

Message ID	Severity	Alert Option	Message Title
CPIBB11	50	*IMMED	Automatic recovery of cluster object &1 attempted.
CPFBB22	40	*IMMED	Cluster Resource Services communications failure on cluster node &2.
CPFBB47	50	*IMMED	Cluster Resource Services detected an error and may have ended abnormally.
CPFBB48	40	*IMMED	Cluster Resource Services error detected.

The following informational messages are generated by Cluster Resource Services:

Table: Informational Messages

Message ID	Severity	Alert Option	Message Title
CPFBB4F	40	*NO	Automatic fail over not started for cluster resource group [CRG name] in cluster [name].
CPIBB09	00	*NO	Cluster resource group exit program [name] in library [name] on cluster node [system name] called.
CPFBB21	00	*NO	Cluster partition condition no longer exists for cluster [cluster name].

Switching the ownership of the iCluster metadata to another node

When the failover mechanism is Cluster Resource Services, switch the ownership of the iCluster metadata to another node.

1. Make sure that all nodes of the cluster are in *ACTIVE status.

 **Tip:** Use the DMWRKNODE (Work with Nodes) command to list the nodes in the cluster and their status.

2. End the node that is the current owner of the iCluster metadata.
Another node in the cluster will take ownership of the iCluster metadata.
3. Restart the node that was ended.
It will no longer be the primary node.

Related information

[DMWRKNODE \(Work with Nodes command\)](#)

Configuring a new cluster to use Cluster Resource Services

Use the DMSETSVAL (Set Cluster System Values) command to configure a new cluster to use Cluster Resource Services.

1. Install iCluster for the first time on a computer.
2. Enter the following command: DMSETSVAL

- Set the **CLUSTER (Use IBM Cluster Services for i)** parameter to *YES.

Related information

[DMSETSVAL \(Set Cluster System Values\) command](#)

Changing an existing cluster to use Cluster Resource Services

Recreate the cluster to change the failover mechanism for an existing cluster to Cluster Resource Services.

- Note:** Be aware that you must recreate the cluster as a part of this process.
 - Note:** When the RTVHACFGS (Retrieve HA Configuration Source) command is used to generate the CL program to recreate your cluster configuration, you can compile the CL programs and run them. However, do not run the program called SETHASVAL because this program changes the roleswitch mechanism to its previous value of Use Cluster Resource Services *NO.
- Record the configuration of your current clustering environment including nodes, groups, and object specifiers. Use the RTVHACFGS (Retrieve HA Configuration Source) command to generate CL programs that can recreate your cluster configuration.

This command creates a new source file with members: SETHASVAL, CRTHANODES, CRTHAGRPS, RTHAOBJS, CTHRABSFS, CRTJRNMG, CRTSCATTR, CRTUSRINF and CRTALRINF. Each member can be compiled into a CL program to recreate your high availability configuration for system values, nodes, cluster resource groups, native and BSF object specifiers, journal management, sync check attributes, user management, and alert management.
 - To delete the current cluster, run the DMDLTCSTR (Delete Cluster) command on every node in the cluster.
 - From the iCluster Main Menu, press Shift + F10.
 - Type **30** at the command line and press Enter.
 - Press PgDn repeatedly to locate the **Use Cluster Resource Services** field.
 - Change the **Use Cluster Resource Services** field value to ***YES** and press Enter.
 - To delete residual cluster information, run the DMDLTCSTR (Delete Cluster) command on every node in the cluster again.
 - Repeat Step 2 and Step 7 on every node in the cluster.
 - To add each node back to the cluster, run the DMADDNODE (Add Node) command. The first node that is added will become the node that owns the iCluster metadata.
- Note:** If nodes have different operating system versions, add the node with the lowest version first.
- To create the groups that will replicate objects, run the DMADDGRP (Add Group) command. Use the information that you recorded in Step 1.
 - To specify the objects to be in each group, use the DMSELOBJ (Select Objects to Group) and DMADDBSF (Add Path Specifier to Group) commands. Use the information that you recorded in Step 1.
 - On the **Work with iCluster Groups** screen, restart all of the groups.

Related information

[RTVHACFGS \(Retrieve HA Configuration Source command\)](#)

-
- [**DMDLTCLSTR \(Delete Cluster command\)**](#)
 - [**DMADDNODE \(Add Node command\)**](#)
 - [**DMADDGRP \(Add Group command\)**](#)
 - [**DMSELOBJ \(Select Objects to Group command\)**](#)
 - [**DMADDBSF \(Add Path Specifier to Group command\)**](#)

Modifying default cluster system values

To change the default cluster system values, use the input screen for the DMSETSVAL (Set Cluster System Values) command.

You can change the default cluster system values that are set during automatic configuration.

1. Navigate to the main menu.
2. To open the input screen for the DMSETSVAL (Set Cluster System Values) command, type **6** at the command line.
3. Press **Enter**.
4. To navigate, press the **Page Up** or **Page Down** keys.
5. Change values in the following categories:
 - Operational (OPER)—Cluster system values that affect operations within the cluster.
 - Automatic Reactivation (ACT)—Cluster system values that affect the automatic reactivation of suspended objects.
 - Object (OBJ)—Cluster system values that pertain to objects replicated within the cluster.
 - Spooled File (SPLF)—Cluster system values that affect the replication of spooled files within the cluster.
 - Physical File (PF)—Cluster system values that affect the mirroring of physical files within the cluster.
 - Byte Stream File (BSF)—Cluster system values that affect the replication of BSF objects within the cluster.
 - Event Log (EVNTLOG)—Cluster system values that affect the information that is displayed through the event log.
 - Latency (LATENCY)—Cluster system values that affect the latency threshold settings within the cluster.
 - Cluster Values (CLUSTER)—Controls whether iCluster uses Cluster Resource Services as its failover mechanism.
 - Performance (PERFORM)—Controls the performance of iCluster in certain scenarios.
 - *FULL sync check options (FULLSCOPTS)—Cluster system values that affect some aspects of *FULL sync checks by iCluster.
 - Continuous sync check (SYNCCHECK)—Cluster system values that affect the performance of synchronization checks by iCluster.
 - Custom journal entries (CUSTJRNE)—Cluster system values that specify the journal entries to process.
 - Replication monitoring (DMMONHA)—Cluster system values that specify options to start a job to automatically monitor group replication status, event log messages, and suspended objects.

Related information

- [**DMSETSVAL \(Set Cluster System Values\) command**](#)

Role switch operations

Role switch changes the primary node of a group or groups to become the backup node or nodes. Role switches ensure that production applications run correctly on the new primary node.

The current backup node is prepared to become the primary node, in the same way that a failover happens when the group is active.

Changing a primary node to backup node

Use the DMCHGROLE (iCluster Change Primary Node) command for groups in the ***INACTIVE** status.

You can change the role to test an application or when there is planned maintenance for the primary node.

1. Enter the following command:
DMCHGROLE
2. Specify the group or groups to change the role:
 - All groups: In the **Group** field, specify ***ALL**.
 - A single group: In the **Group** field, specify the replication group name.
 - All groups that use a specific primary node:
 - In the **Group** field, specify ***PRIMNODE**.
 - In the **Primary node** field, specify the node name.

The groups retain ***INACTIVE** status until the groups are started with the DMSTRGRP (iCluster Start Group) command.

Related information

[DMCHGROLE \(Change Primary command\)](#)

[DMSTRGRP \(Start Group command\)](#)

Starting a role switch

Use the DMSTRSWO (Switchover Group) command for groups in the ***ACTIVE** status.

Before you begin: The following conditions must be met before the role switch can start:

- Groups must be active.
- Group type must be replication or MQSeries.

Refresh-only groups, and groups with a value that is not ***PRIMARY** in the **Target library** field, are not eligible for role switch.

A role switch can be initiated for all active groups in the cluster or for a specific group to cause a switchover that involves the primary and backup nodes. Initiate a role switch for all groups or a single group.

1. On any active node in the cluster, enter the following command:
DMSTRSWO
2. In the **Group** field, type one of the following values:
 - All groups— ***ALL**
 - A single group—The replication group name

- In the **Restart replication** field, specify whether to restart replication after the role switch is completed. Choose one of the following values:
 - *YES—Restarts replication. By default, this value is used.
 - *NO—Does not restart replication during the role switch. Restart replication later by using the **DMSTRGRP** command with USEMARKED(*YES).

After a role switch is completed, the current primary node in each group becomes the backup node and the current backup node in each group assumes the role of the primary node.

Related information

[DMSTRSWO \(Switchover Group command\)](#)

[DMSTRGRP \(Start Group command\)](#)

Role switch simulator

Performing a full role switch involves planning downtime to end production applications so that testing is done during off hours. Use the iCluster role switch simulator frequently to test the integrity of the backup node.

Simulating role switches

Role switch simulation allows the node to temporarily simulate the primary node for the role switch simulation without disrupting operations on the primary production node.

Before you begin: The following conditions must be met before the role switch simulation can start:

- Groups must be fully active
- Group type must be replication or MQSeries
- Group must have **JRNBACKUP** (Journal objects on backup) parameter set to *YES

Refresh-only groups, and groups with a value that is not *PRIMARY in the **Target library** field, are not eligible for a role switch simulation.

The role switch simulation ends all replication apply processes on the node and enables triggers and constraints. The following changes, if done during a role switch simulation, cannot be rolled back:

- ALTER TABLE
- Changes to *DTAQ objects
- Journalized BSF content changes
- Changes to SQL objects
- CLRPFM of a physical database file (object type *FILE with attribute PF-DTA)
- Record changes on *FILE/PFDTA objects that are renamed or moved during role switch simulation
- Content changes on *DTAARA objects that are renamed or moved during role switch simulation

Commitment cycles that are opened during the role switch simulation must be closed. Open commitment cycles cannot be rolled back.

- On the backup node, enter the following command:

DMSTRSS

- In the **Group** field, specify which groups to simulate a role switch for.

Enter one of the following values:

- <groupname>—Simulate role switch for a single replication group
- *BACKUP—Prepare all groups that are replicating to this backup node and are eligible for a role switch.

3. In the **Allow out of sync objects** field, specify whether groups that are involved in the role switch simulation can have out-of-sync objects.

Enter one of the following values:

- *NO—Do not allow groups to have out-of-sync objects. By default, this value is used.
- *YES—Allow groups to have out-of-sync objects..

4. In the **Allow suspended objects** field, specify whether suspended objects are allowed for the specified group.

Enter one of the following values:

- *NO—Do not allow groups to have suspended objects. By default, this value is used.
- *YES—Allow groups to have suspended objects.

5. Specify whether to allow journaling errors when the role switch simulation changes database file journaling to use both images on the files and data areas in the **Allow journaling errors** field.

Choose one of the following values:

- *NO—Do not allow errors during journal changes. By default, this value is used.
- *YES—Allow errors during journal changes. The role switch simulation proceeds even when errors occur when journaling for database files and data areas is changed from *AFTER image to *BOTH images.

The status of the groups that for the role switch simulation changes to *RSS during the role switch simulation.

Groups whose apply processes are ended but are not part of the role switch simulation have the status

*ENDAPY during the role switch simulation.

6. Test your backup environment. After the role switch simulation is complete, run the DMENDRSS (End Roleswitch Simulator) command.

Related information

[DMSTRSS \(Start Roleswitch Simulator command\)](#)

[DMENDRSS \(End Roleswitch Simulator command\)](#)

Ending role switch simulation

After you run the DMSTRSS (Start Roleswitch Simulator) command and the role switch simulation is completed, end the role switch simulation.

Before you begin: You can end role switch simulation only on the backup node after a role switch simulation was started.

The DMENDRSS (End Roleswitch Simulator) command restores replication processes, database files, and objects to the state that they were in before the role switch simulation.

1. On the backup node, enter the following command:

```
DMENDRSS
```

2. In the **Type of DB journal rollback** field, specify how to roll back the journal changes that occurred during the simulation.

Enter one of the following values:

- *OBJECT—Perform database journal rollback only for objects that are in replication scope for the groups in *RSS state.

Ending the role switch simulation with this option takes longer because the changes to each are rolled back individually. The database processes the journal receivers multiple times, once for each object.

Use this option if changes to objects that are not in replication scope, but are in a library with objects that are in replication scope, might take place during a role switch simulation.

By default, this value is used.

- *FAST—Perform database journal rollback at the library level. All changes reported to a journal for each library are rolled back at the library level for the period of the role switch simulation.

As a result, changes that are rolled back are processed by the database faster.

Do not use this option if *FILE or *DTAARA objects are changed during the role switch simulation, when the objects are not in replication scope, but reside in a library that includes objects that are in replication scope. These changes are rolled back during the end role switch simulator processing.

3. In the **Run sync check** field, specify whether to run a sync check at the end of command processing.



Note: After RSS, it is highly recommended to run the *Checksum sync check.

Enter one of the following values:

- *NO—Do not run a sync check. By default, this value is used.
- *YES—Run a *FULL sync check when the command processing is complete. The REPAIR options *JRN and *AUTH are specified to correct any journaling or authority mismatches in the objects on the backup node.

The replication processes, database files, and objects are restored to the state that they were in before the role switch simulation.

Related information

[DMSTRRSS \(Start Roleswitch Simulator command\)](#)

[DMENDRSS \(End Roleswitch Simulator command\)](#)

Monitoring

Monitoring a logical replication solution is important to maintain the integrity of your high availability environment. Use the Status Monitors to monitor the current status of replication.

Periodically monitor the health of the replication environment using monitoring and synchronization checks.

Use the Status Monitors to verify the following replication values and statuses:

- Replication is active
- Latency values are acceptable for the journal scrape and backup apply processes
- Object throughput is acceptable for the replication processes

Replication monitoring supports email alert messages for group status, event log messages, and object suspension when reportable replication events occur.

You can view and activate suspended objects and out-of-sync objects from the Status Monitors.

Related information

[Configuring email for replication monitoring alerts](#)

Monitoring overview

Monitoring is an important task to ensure a healthy environment for a high availability solution.

Use the status monitors to view replication status and perform operations, including sync check, activate suspended objects, and perform role switches. Monitoring is based on group and journal combinations.

Perform the following operations using this panel:

- Start and end replication
- List and work with object specifiers
- List and work with path specifiers
- View journal details
- View event log messages
- View object status
- Start and end apply processes
- View history log
- Perform a role switch
- Display the **Work with Journals** panel
- View and activate out-of-sync and suspended objects
- Work with iCluster jobs
- Start sync checks
- View node details and display the **Work with Nodes** panel

The following status monitors can be used with iCluster:

Full Cluster Status Monitor

Lists all of the groups in the cluster.

Primary Status Monitor

Lists all of the groups that use the current node as their primary node.

Backup Status Monitor

Lists all of the groups that use the current node as their backup node.

There is also a display-only version of the **Primary Status Monitor** that does not allow operations to be performed, but supports viewing latency, group and journal details, and object status.

Use the monitor that best satisfies your monitoring requirements and replication environment.

Understanding latency

Latency is the time interval between a journal entry for an operation that is written on the primary node and the operation that caused the journal entry to be applied on the backup node.

Latency is sometimes expressed as the number of entries between the last journal entry that was written and the last journal entry that was applied. High latency can occur when the volume of transactions exceeds the available communications bandwidth and processing resources on the backup node.

Latency values can become very large in several hours if the apply process is suspended. The scrape latency is not affected during apply suspension when the staging store is large enough to hold the journal information.

The status monitors show the latency of replication processes. When the numbers in **Source Latency**, **Apply Latency**, and **Total Latency** columns are minutes or seconds, or get smaller when the view is refreshed, then replication is current between the primary node and the backup nodes. The currently active groups display in each view.

You can adjust the latency level to acceptable levels for the cluster. To set the latency check interval and the latency threshold, use Option 6 from the main menu or set latency values using the DMSETSVAL (Set Cluster System Values) command.

Statistics provide information about real time and historical latency. Examine the speed of replication from the primary node to the backup node and adjustment values as required.

Related information

[**Staging objects**](#)

[**DMSETSVAL \(Set Cluster System Values\) command**](#)

Status Monitor views

The status monitors display real-time information across multiple views. The Full Cluster Status Monitor requires a terminal emulator with a 132-column display.

The information on the status monitor display is recalculated every time the screen is refreshed. Latency is calculated only when the current node has an active connection to each backup node.

Terminal emulator settings determine the number of available views. On an 80-column display, the Full Cluster Status Monitor and the **Real Time Overall Latency** views are not available.

On a 132-column display, the following views are available in this order:

- **Real Time Overall Latency**
- **Real Time Object Latency**
- **Real Time Object Position and Totals**

-
- **Real Time Object Throughput**

Related information

- [Real Time Overall Latency view](#)
- [Real Time Object Latency view](#)
- [Real Time Object Position And Totals view](#)
- [Real Time Object Throughput view](#)

Real Time Overall Latency view

The **Real Time Overall Latency** view displays the real time overall latency for all primary/backup/group/journal combinations.

The **Real Time Overall Latency** view provides a summary of the following latency information.

- **Source Latency**—The difference between the timestamp of the journal entry that was last processed by the backup receiver, and the timestamp of the last journal entry that was deposited in the primary journal. For idle or lightly used groups, the receive timestamp is resynchronized every minute to ensure that reported latencies remain below specified thresholds. Angle brackets (<>) are displayed around the apply latency time when the primary latency threshold is exceeded.
- **Apply Latency**—The difference between the timestamp of the journal entry that was last processed by the backup apply process, and the timestamp of the last journal entry that was processed by the backup receive process. Angle brackets (<>) are displayed around the apply latency time when the primary latency threshold is exceeded.
- **Total Latency**—The sum of the source and apply latency times.
- **Total Latency Status**—An approximate estimate of the total latency. Each asterisk (*) or period (.) represents one unit of time, depending on the sum of the primary and apply threshold values. The sum of the thresholds displays 2/3 of the way across the bar graph. If latency thresholds are not exceeded, the bar graph displays periods (.). If latency thresholds are exceeded, the bar displays asterisks (*).
- **Status**—Each monitoring view has a set of columns that display status information for the nodes, groups, and journals that are being monitored.

Related information

- [Understanding latency](#)
- [Status column information](#)
- [DMSETSV \(Set Cluster System Values\) command](#)

Real Time Object Latency view

The **Real Time Object Latency** view displays the real time latency for all group and journal combinations.

The **Real Time Object Latency** view displays the following transaction information:

- **Backup J/E**—On the Primary Status Monitor only. The last journal entry that was processed by the backup apply process.
- **Curr Trans Sent**—The number of transactions that are being processed on the primary node and are received into the staging store since the start of the current replication process. Omitted entries are not included.
- **Curr Trans Applied**—The current number of transactions that are being processed by the primary node since the start of the last reset of the transaction counters that were applied by the apply processes on the backup node.

Inactive apply processes are not counted in the backup node and group summaries.

- **Trans to Process**—The number of transactions that were sent by the primary node, but are not yet applied on the backup node. A negative number might display briefly because of timing differences during sampling of the primary and backup node replication processes. Negative numbers might also display when there are no transactions to process, but extra transactions were sent to satisfy the process requirements of the journal entries. For groups with a *LEVEL1 or *LEVEL2 commitment control level, this number of transactions to process does not include the number of transactions that were processed but not applied because their commit control units were not committed on the primary node.
- **Status**—Each monitoring view has a set of columns that display status information for the nodes, groups, and journals that are being monitored.

Latency counts for all nodes are recalculated every time that the screen is refreshed. Transaction counts are reset under the following conditions:

- After the DMSETPOS (Set Journal Start Position), DMMRKPOS (Mark Journal Positions), or DMREGPOS (Register Positions) commands are run.
- Refresh before mirroring

The columns for transaction counts are narrower on the 80-column display. Transaction count resets vary by display:

- Transaction counts are reset on the 132-column display after the counts exceed 100,000,000,000,000,000,000.
- Transaction counts are reset on the 80-column display if the numbers exceed the width of the column that is displaying them.

Related information

[Status column information](#)

[DMSETPOS \(Set Journal Start Position command\)](#)

[DMMRKPOS \(Mark Journal Positions command\)](#)

Real Time Object Position And Totals view

The **Real Time Object Position and Totals** view displays the real time position and totals for all active and inactive group/journal combinations.

Active combinations are highlighted.

The **Real Time Object Position and Totals** view displays the following journal processing and transaction information:

- **Last J/E**—The last journal entry that was written to the journal.
- **Primary J/E**—The journal entry where the journal scraper process is scraping the journal.
- **Backup J/E**—The last journal entry that was applied to the backup node by the apply process.
- **Trans to Process**—The number of transactions that were sent by the primary node, but were not yet applied on the backup node.
- **Status**—Each monitoring view has a set of columns that display status information for the nodes, groups, and journals that are being monitored.

Related information

[Status column information](#)

Real Time Object Throughput view

The **Real Time Object Throughput** view displays the real time throughput in transactions per hour for the backup node apply processes for all active group and journal combinations.

Active combinations are highlighted. Overall and current throughputs provide a summary of throughput values for all active backup nodes. The throughout statistics show throughput in transactions per hour. Throughput values do not display for inactive group/journal combinations.

The **Real Time Object Throughput** view displays the following information:

- **Apply Str/End**—The start/end date of the latest replication process for the backup node/ journal combination. If the **Status** column displays a status of I (inactive), then this field is displaying the end time. Otherwise, it indicates the start date.
- **Elapsed Days Hr Mn**—The elapsed time since the latest apply process for the journal of the backup node/ group combination began. The elapsed time is the time since the last monitoring restart (press F10) or the monitor refresh (press F5).
- **Overall Trans/Hr**—The number of transactions per hour, based on the elapsed time of the backup node apply job.
- **Current Trans/Hr**—The number of transactions per hour, based on the transactions that were performed since the last time the monitor timer was restarted. A hyphen (-) displays when the elapsed time is zero (0) and a value cannot be calculated. Press F10 to restart the timer and recalculate the value, based on the elapsed time since the timer was restarted.
- **Status**—Each monitoring view has a set of columns that display status information for the nodes, groups, and journals that are being monitored.

Throughput calculation

Overall throughput on the backup node apply process is calculated by using figures (time period and transactions processed) that are based on the elapsed run time of the apply job on the backup node. The time period is from the start of the apply job on the backup node to the last time that the information was updated. This throughput rate is expressed in transactions per hour.

The throughput rate is calculated with the following formula:

(Backup Node Update Timestamp) minus (Backup Node Start Processing Timestamp)

The number of transactions is the total of all transactions that were processed on the backup node.

Current throughput is based on the throughput of the backup node apply process by using the displayed values, or the values since the previous restart, to the time when a refresh of the monitor is requested.

Related information

[Status column information](#)

Status column information

Each monitoring view has a set of columns that display status information for the nodes, groups, and journals that are being monitored.

Remote journal process, scrape process on primary node, apply process on backup node (RPB)

The **RPB** status column represents the status of replication processes for remote journals, if they are being used for replication, (R); the scrape process on the primary node (P); and the apply process on the backup (B) nodes.

The following codes can be displayed in the **RPB status** column on the monitor screens:

A

The replication process is active on the remote journal, primary node, or backup node:

- **R** column: The remote journal is active and is receiving journal entries from the source journal.
- **P** column: The journal scrape process on the primary node is active.
- **B** column: The apply process is active for the journal on the backup node.

D

B column: The apply process for the journal on the backup node is delayed, based on the Apply Delay setting for the group.

I

The replication process is inactive on the remote journal, primary node, or backup node:

- **R** column: The remote journal is inactive and is not receiving journal entries from the source journal.
- **P** column: The journal scrape process on the primary node is inactive.
- **B** column: The apply process is inactive for the journal on the backup node.

L

R column: The remote journal has latency receiving the local journal entries from the primary node.

R

The group is a refresh-only group and its processes are active:

- **P** column: The journal scrape process on the primary node is inactive.
- **B** column: The apply process is inactive for the journal on the backup node.

P column: The group is a refresh-only group and its processes are active.

S

P column: A group is starting but is not fully active, or a group is in the process of shutting down. Communication is not occurring with the backup node.

U

R column: The remote journal is not being used. The journal is not displayed when it is no longer required by a group on both the primary and backup nodes.

-

The status is unknown:

- **R** column: The status of the remote journal is unknown.
- **P** column: The status of the journal scrape process on the primary node is unknown.
- **B** column: The status of the apply process for the journal on the backup node is unknown, or the primary node is not actively communicating with the backup node.

The status is unknown for a group that has never been referenced in replication.

*

R column: The remote journal is unused.

(blank)

R column: The journal is not a remote journal.

Suspended objects (S/O)

The **S/O** status column displays the number of suspended objects in the group. Suspended objects match the object specifier but cannot be replicated. This field is blank when no objects are suspended.

Out-of-Sync (OOS)

The **OOS** status column displays the sum of the objects that are out-of-sync for each group. This number is the result of the last sync check minus the OOS objects that were activated since the last sync check. The **OOS** column is available only with a 132-column display.

Sync check (S)

The **S** status column displays the status of sync check for each group:

- **Y**—Sync check processes are active for the group.
- blank—Sync check processes are not active for the group.

Role switch readiness (R)

The **R** status column displays the role switch readiness for the group:

- **Y**—The group is ready for a role switch.
- blank—The group is not ready for a role switch.

To set the role switch readiness criteria for a group, use the **Allow OOS for switch readiness** and **Allow SUS for switch readiness** group and system parameters. Role switch readiness requires that the group is fully active, and that the group is eligible for role switch.

Operational (OP)

This **OP** status column displays the current operational status for each journal.

- **RBD**—The group is waiting for the system to finish rebuilding a logical file.
- **RFS**—The operation is refreshing a file or IFS object. This status can be the result of a group refresh, or an individual file or IFS object that is being activated by a refresh. For database file members, the progress percentage of the refresh is appended.
- **RGZ**—The operation is reorganizing a file.
- **SWO**—The group is performing a switchover.
- **(blank)**—The group is used only for mirroring.

The **OP** column is available only with a 132-column display.

Current object (Curr Object)

For journals—The name or path of the object. Native objects display the LIBRARY/FILENAME(FILEMEMBER). IFS objects display the last 30 characters of the path and file name.

During a switchover, the current switchover step is shown.

Related information

- [**Real Time Overall Latency view**](#)
- [**Real Time Object Latency view**](#)
- [**Real Time Object Position And Totals view**](#)
- [**Real Time Object Throughput view**](#)

Monitoring the replication environment

Monitor the status of the replication environment, which identifies conditions that might require your attention.

Before you begin: The Full Cluster Status Monitor requires a terminal emulator with a 132-column display.

1. Use one of the following options from the main menu to start monitoring of the replication environment:
 - Option 8 from any active node (WRKCSMON): Displays the Full Cluster Status Monitor that shows the status of group and journal combinations on all nodes.
 - Option 80 from the primary node (WRKHASMON): Displays the Status Monitor for groups that use the current node as the primary node.
 - Option 81 from the backup node (WRKHATMON): Displays the Status Monitor for groups that use the current node as the backup node.
 - Option 82 from the primary node (DSPHASMON): Displays a view-only version of the Status Monitor for groups that use the current node as the primary node. Prevents operational control of iCluster.

For the WRKCSMON command, the **Real Time Overall Latency** view is displayed.

Real Time Overall Latency										System: ICDEV54A									
Position to				Elapsed time . . . : 01:00:13															
Type options, press Enter.																			
1=Start 2=Obj Specs 3=BSF Specs 4=End 5=Details 6=Msgs 7=BSF,SQL Sts 8=Obj Sts 10=StrApY 11=EndApY 12=History 13=Activate OOS 20=Roleswitch 41=work w. jrn 54=synchck 56=User synchck 93=Node list 94=iCluster jobs																			
Source Apply Total																			
Opt Prm/Bkp/Grp/Jrn HH:MM:SS HH:MM:SS HH:MM:SS Total Latency Status																			
RPT S/O OOS S R OP																			
ICDEV54A AA																			
ICDEV61A --																			
IVDEVD --																			
QAUDJRN I-																			
ICDEV71A AA																			
IVDLOTEST AA																			
QAUDJRN :31 :00 :31 .																			
IVGRP AA																			
QAUDJRN :25 :00 :25 .																			
HADJRN :00 :00 :00																			
AAA																			
IVIASP1 AA																			
QAUDJRN :25 :00 :25 .																			
HADJRN :00 :00 :00																			
AA																			
IVMANY AA																			
QAUDJRN :19 :00 :19 .																			
AA																			
More...																			
F3=Exit F4=Prompt F5=Refresh F9=Sync Check Results F10=Re-Start F11=Latency (view 2) F12=Cancel F21=Command																			

2. **Optional:** Press F11 to toggle the view between **Real Time Overall Latency**, **Real Time Object Latency**, **Real Time Object Position and Totals**, and **Real Time Object Throughput**.
3. **Optional:** Position your cursor in any field and press F1 to view help. Press F2 to view extended help.
4. **Optional:** Press F5 to refresh the monitoring display.
5. **Optional:** Press F10 to reset the column timer of elapsed time to zero for the calculation of the values in the **Current Trans/Hr** on the **Real Time Object Throughput** monitor display.
6. Press F3 to return to the menu or command line.

Related information

[Searching while monitoring](#)

[Configuring replication monitoring](#)

[Configuring email for replication monitoring alerts](#)

[Configuring replication monitoring alerts](#)

[Performing tasks from the status monitors](#)

[WRKCSMON \(Work with Full Cluster Monitor command\)](#)

[WRKHASMON \(Work with Status Monitor on Primary Node command\)](#)

[WRKHATMON \(Work with Status Monitor on Backup Node command\)](#)

[DSPHASMON \(Display Source Monitor command\)](#)

Performing tasks from the status monitors

Use the status monitors to view replication status and perform operations.

1. To verify that the objects on the backup node are the same as the objects on the primary node, view the Out-of-Sync (OOS) Count Column.
The replication status for a group is good when the **Status** columns for **S/O** and **OOS** columns are blank, and **AA** displays in the **PB** column for all group-journal entries for a single group.
2. To perform an operation on a node, group, journal, or group/journal combination in the status monitor, type one of the following options in the **Opt** column for that node, group, or journal:

Option	Action
1	Start replication.
2	List the object specifiers.
3	List the path specifiers.
4	End replication.
5	Display journal receiver details.
6	Display event log messages that relate to the group.
7	Display IFS and SQL object status.
8	Display the native object status for a group/journal combination.
10	Start the apply processes on the backup node.
11	End the apply processes on the backup node.
12	Display the Historical Latency panel.
13	Activate out-of-sync objects.
20	Start a role switch.
41	Display the Work with Journals (DMWRKJRN) panel.
54	Start a sync check.
56	Start a sync check for a specific object or objects that are selected to the group.
93	Display the Work with Nodes (DMWRKNODE) panel to view a list of nodes.
94	Display the Active Jobs (WRKACTJOB) panel to view the active jobs that are involved in replication.

3. To perform a function in the status monitors, press one of the following function keys:

Function key	Action
F5	Refresh the monitoring view.
F9	View the list of spooled files that contain sync check results.
F10	Restart monitoring to reset the elapsed time.

Related information

[DMWRKJRN \(Work with Journals command\)](#)

[DMWRKNODE \(Work with Nodes command\)](#)

Searching while monitoring

To search, use the **Position to** field to minimize scrolling in a long list of items in the Full Cluster Status Monitor and other monitors. If the node, group, and journal combinations do not fit on a single monitor screen, you can quickly move to the top or bottom of the list, or to a specific node, group, or journal.

To position to the next occurrence of the name in the list, enter a value (for example, a group name) in the **Position to** field. You can also specify primary node, backup node, group, or journal, followed by the name. When an entry type is not specified, entries are searched in the following order:

1. Primary node
2. Backup node
3. Group
4. Journal

In the **Position to** field, enter one of the following values:

- *name*: The next occurrence of *name* in the list. The *name* can be the name of a node, group, or journal.
- *TOP: Top of the list.
- *BOT: Bottom of the list.
- P:*node-name*: The *node-name* node that is used as a group primary node.
- B:*node-name*: The next occurrence of the *node-name* node that is used as a group backup node.
- G:*group-name*: The *group-name* group.
- J:*journal-name*: The next occurrence of the *journal-name* journal.

Configuring replication monitoring

The DMMONHA job monitors group replication status, event log messages, and suspended objects. To configure replication, set the replication monitoring values for this job.

The DMMONHA job runs only on the node that owns the iCluster metadata.

To enable and configure replication monitoring, set replication monitoring values by prompting the **DMSETSVL -Set Cluster System Values** command and setting the following DMMONHA replication monitoring job options:

Job option	Action
Enable replication monitoring	Automatically enable replication monitoring.

Job option	Action
Monitor event log	Specifies the level, or a set of levels, of severity of event log messages, or specific event log message IDs, that trigger an alert.
Monitor suspended objects	Specifies to check for suspended objects.
Monitor group status	Specifies to check group replication status.
Clean up event logs	Specifies to clear expired messages from the event logs for all active nodes in the cluster.
Check interval (min)	Specifies a number of minutes to define the interval between the checks.
Objects suspended for (min)	Specifies a number of minutes to define the minimum duration for an object to be suspended before the suspended object status is reported in an alert message. Applies only when Monitor suspended options is *YES.
Group inactive or in doubt for	Specifies a number of minutes to define the minimum duration for a group to be inactive before the inactive group status is reported in an alert message. Parameter applies only when Monitor group status is *YES.
Event message IDs to monitor	Specifies a list of message identifiers in the event log for monitoring by the DMMONHA job. Messages of severity 40 are always checked when event log checking is enabled.
Event message IDs to omit	Specifies a list of message identifiers in the event log to omit from monitoring by the DMMONHA job. Parameter applies only when Monitor event log is *SELECT.
Destination type	Specifies the destinations to use for the alert messages that are generated by the DMMONHA job

Related information

[DMSETVAL \(Set Cluster System Values\) command](#)

Configuring email for replication monitoring alerts

Replication monitoring for group status, event log messages, and object suspension supports email alert messages when reportable replication events occur. Configure email alerts when reportable replication events occur.

Configuring email for these alerts is a task that is typically performed by the IBM i administrator.

1. Configure TCP/IP for email on the cluster nodes.
2. Configure the SMTP server on the cluster nodes.
3. Ensure that the SMTP TCP/IP service is started.
4. Enroll the DMCLUSTE directory as an email user:
 - a. Ensure that the directory entry (WRKDIRE) for DMCLUSTE exists on the host.
 - b. Verify that DMCLUSTER is listed as the user profile for the DMCLUSTE directory entry.
 - c. Change the directory entry and press F19 to change the name for SMTP.
 - d. Enter the SMTP user ID (should be DMCLUSTER) and the SMTP domain.

To find the domain name for the current node, prompt the IBM i Change TCP/IP Domain Information (CHGTCPDMN) command.

5. To add the email alert destination, run the **DMWRKALR-iCluster Alert Destinations** command.
6. Press F6 to add an alert.
7. In the **Address type** field, specify ***EMAIL**. By default, this value is used.
8. In the **Destination** field, specify a valid email address in SMTP format.
9. In the **Character set** field, specify the character set designator for the multipurpose Internet mail extensions (MIME).
10. In the **Encoding** field, specify the encoding designator for the Content-Transfer-Encoding field in the MIME-format email messages.

Related information

[DMWRKALR \(iCluster Alert Destinations command\)](#)

Configuring replication monitoring alerts

Replication monitoring for group status, event log messages, and object suspension supports alert messages when reportable replication events occur. Configure message queue alerts when reportable replication events occur.

1. To add the message queue alert destination, run the **DMWRKALR-iCluster Alert Destinations** command.
2. Press F6 to add an alert.
3. In the **Address type** field, specify ***MSGQ**.
4. In the **Message queue** field, specify the target message queue for message queue entries from the DMMONHA job.
5. In the **Library** field, specify the library where the message queue resides.

Related information

[DMWRKALR \(iCluster Alert Destinations command\)](#)

Object and group status views

Work with Object Status views

Display the **Work with Object Status** screen from any status monitor to view native object status for the group/journal combination.

The **Work with Object Status** screen displays the status of the selected object.

The views of the **Work with Object Status** panel group are **Suspended objects**, **Active objects**, and **All objects**. Only journaled objects (database files, data areas and data queues) are listed in the **Active objects** view. In the **All objects** view, all journaled objects and suspended non-journaled objects are listed.

The group, journal, and journal library are displayed at the top of the screen. The type of object that is selected determines the information that is displayed.

This screen displays the status of objects for a group. The group name and its primary and backup nodes are displayed at the top of the screen. Use the "Position to" fields to locate an entry in the native object list.

- Position to object name, object type, and object library

Use the "Position to" fields to locate an entry in the native object list using three fields, for the object name,

object type, and object library, in that order. You can enter a value into one, two or all of the fields, leaving one or two of the fields blank.

The first field in the list is used for the object's name, the second field is for the object's type and the third field in the list is for the object's library.

If you use the object type field (the second field), you must enter an object type exactly as it appears in the list of objects in order to position to an entry in the list.

If you use the object name or library fields, you can enter either the full name or library, or just the first part of the object name or library. The "Position to" processing will position to the first entry in the list whose object name or library is equal to or lexically greater than the name or library entered into the "Position to" name or library fields, and matches the object type field, if used.

If no entries in the list match the "Position to" values specified, the list will be positioned to the last entry in the list.

"Position to" is useful to minimize scrolling through a long list of objects. Fill one or more of the fields with the name, type or library of the object you are looking for and press Enter.

The following object status information displays across multiple views of the **Work with Object Status** screen:

- **Status**—One of the following statuses of each object:
 - **ACTIVE**—The object is available for replication.
 - **ACTOOS**—The object is out-of-sync, but is still being actively replicated.
 - **PNDACT**—Active pending. The object is suspended and waiting to become active through a journal entry that must be scraped.
 - **PNDSUS**—The object is active and waiting to be suspended through a journal entry that must be scraped.
 - **SUSPND**—The object is suspended.
- **Object**—The object name.
- **Type**—The type of the object (for example, *FILE, *DTAARA, *USRPRF).
- **Library**—The library that the object belongs to.
- **Attr**—The extended attribute for the object.
- **Jrn Pos**—The journal position where the object was suspended (for objects in SUSPND status).
- **Reason**—The reason why the object was suspended. A reason code is not displayed for objects that are not suspended.

The following additional information is displayed for a group:

- **OOS Reason**—The reason for the out-of-sync status. Applies to native objects only.
- **OOS Date-time**—The journal position where the object was suspended (for objects in SUSPND status).
- **Sus Reason**—The reason why the object was suspended.
- **Auto Rcvy**—Specifies if the object is eligible for auto-recovery.

Related information

[Object status reason codes](#)

[Performing tasks from the status monitors](#)

[DMSETSVAL \(Set Cluster System Values\) command](#)

Work with BSF/SQL Status views

Use the **Work with BSF/SQL Status** screen lists the BSF and SQL object status for the selected group/journal combination.

The **Work with BSF/SQL Object Status** views are **Suspended objects** and **Active objects**. Only journaled path objects are listed in the **Active objects** view. This view applies only to journaled BSF and is populated only when the **Keep list of active BSF** system value is *YES.

The group name and name of the journal are displayed at the top of the screen. Use the "Position to" field to locate an entry in the list.

- Position to object:

Use the "Position to" field to locate an entry in the list using the pathname or partial pathname of the object, up to the first 48 characters of the pathname. The list will be positioned to the first entry in the list whose pathname is lexically equal to or greater than the value entered into the Position to field.



Note: The characters entered into the field must match the characters in the pathname in the list exactly, including case. Single or double quote delimiters should not be used in the field unless they are part of the pathname being searched for. Note that uppercase characters are lexically greater than their lowercase equivalents.

If no entries in the list match the "Position to" value specified, the list will be positioned to the last entry in the list. "Position to" is useful to minimize scrolling through a long list of objects. Fill the field with the pathname or partial pathname of the object you are looking for and press Enter.

This screen also displays the following information:

The following object status information is provided across multiple views of the **Work with BSF/SQL Object Status** screen:

- **Status**—One of the following statuses of each object:
 - **ACTIVE**—The object is available for replication.
 - **PNDACT**—Active pending. The object is currently suspended and waiting to become active through a journal entry that must be scraped.
 - **PNDSUS**—The object is active and waiting to be suspended through a journal entry that must be scraped.
 - **SUSPND**—The object is currently suspended.
- **Type**—The type of the object (for example, *FILE, *DTAARA, *USRPRF).
- **Object**—The replication object.
- **Jrn Pos**—The journal position where the object was suspended (for objects in SUSPND status).
- **Reason**—The reason why the object was suspended. A reason code is not displayed for objects that are not suspended.

Related information

[Suspended objects](#)

[Object status reason codes](#)

[Out-of-sync reason codes](#)

Object status reason codes

Object status is shown as three-character codes. The full description of the object status reason codes provides

information about why the object was suspended.

The reason codes and their meanings are as follows:

- (blank)—The object is not suspended.
- ABU—The object was activated by the user.
- AIS—The object will be activated as soon as the activation entry is scraped from the journal.
- APD—The object activation started, but is not finished.
- AUD—Object auditing cannot be started for a BSF object.
- AUS—Private authorities that are associated with the BSF object could not be replicated or retrieved.
- AUT—Private authorities for an object could not be retrieved.
- BTN—Metadata for a BSF object cannot be found on the backup node.
- CDA—The change content operation to a data area failed.
- CHK—Temporary state during a refresh file attempt during a rename or move operation.
- CIL—Unable to determine if a BSF object is a hard link.
- CJN—A required journal entry that is associated with the object could not be found in the audit journal.
- COM—An object could not be refreshed.
- CPT—A journal entry that is required to refresh the object could not be added to the database journal.
- CRT—The BSF object cannot be created on the backup node.
- DLT—A BSF object could not be removed from the backup node.
- DTA—The BSF object could not be opened on the primary node for refreshing.
- EJF—An object was suspended on the primary node because journaling for the object ended on this node.
- EXS—iCluster attempted to create an IFS folder that already existed.
- FLF—A file was suspended on the backup node because a file-level failure occurred during replication
- IDC—The user profile is prevented from having the same UID or GID values on the backup node as on the primary node because of a UID or GID conflict with another user profile on the backup node. Changes to the user profile are still being replicated.
- INT—An object was suspended as a result of an internal failure. Synchronization between objects on the primary and backup nodes cannot be verified.
- IOF—A read or write operation failed.
- JPF—A logical file was suspended on the primary node because the associated physical file could not be journaled.
- JPO—A required journal entry that is related to the object could not be found in the database journal.
- JRN—The file was suspended on the primary node because iCluster did not start journaling on the file.
- LCK—A lock on an object could not be obtained.
- LNK—The BSF object is a hard link (replication of hard links is not supported by iCluster).
- MCN—The object is temporarily suspended during the renaming or moving of a database table, where the original object is selected with MIRRCNTS(*NO) and the new object with MIRRCNTS(*YES).

The object is activated by iCluster when the renaming or moving of the database table is complete on the target.

- MDF—A problem occurred while creating or updating metadata that is related to the object.
- MLF—An object was suspended on the backup node because a member-level failure (rename, delete, reorganize, and so on) occurred during replication.
- MRR—An object was suspended on the primary node. The object should be refreshed manually, but it has yet to be activated.
- NFD—The object was not found on the system.
- NGP—A logical file was suspended on the primary node because the associated physical file was not replicated

in the same group as the logical file.

- NRE—A file has no record in the metadata.
- NSI—A BSF object exists on the backup node, but not on the primary node.
- NSO—A physical file has fields of unsupported type (datalink or LOB fields with MINENTDTA).
- NTI—The state associated with a BSF object could not be found on the backup node.
- NVR—The object was suspended because this object type cannot be replicated. The object cannot be activated.
- OLF—An object level failure occurred during an attempt to rename or move a non-journaled object.
- OWN—The owner of BSF object could not be changed on the backup node.
- PCK—Compression of the BSF object path was unsuccessful.
- PDA—Uncommitted DO "delete object" entry is received for an active object under commitment control.
- PDS—Uncommitted DO "delete object" entry is received for a suspended object under commitment control.
- PDU—Uncommitted DO "delete object" entry is received for an object suspended by the user under commitment control.
- PGP—The primary group of a BSF object could not be changed on the backup node.
- PND—Activate pending for a non-journaled object.
- RBC—The file was part of a canceled rollback operation on the primary node.
- RDQ—The receive operation to a data queue failed.
- RFF—A refresh of a BSF object was unsuccessful.
- RFS—A refresh of a BSF object could not be started from the primary node.
- RGF—The file was suspended because a member could not be reorganized or cleared on the backup node
- RLE—An object was suspended on the backup node because the number of I/O errors that were generated while replicating to the object exceeded the **Max. record level errors** cluster system value.
- RLS—The object cannot be refreshed to the backup node because its target release is incompatible with the operating system release level of the backup node.
- RMV—A RMVJRNCHG journal entry was processed for the object. The object is suspended and will be refreshed later by the automatic reactivation function.
- RNM—Rename or move operation failed.
- RRD—A record-by-record refresh of the file failed due to record read failure.
- RSF—An object was suspended on the backup node. The object should have been refreshed, but the restore operation failed on the backup node.
- RST—A BSF object was restored. The object must be refreshed.
- RTN—An unexpected return code occurred for an object.
- RTV—The file or object description could not be retrieved.
- RWA—An object was suspended on the primary node because a record-by-record refresh failed.
- SBU—An object was explicitly suspended on the primary node as a result of the **DMSUSOBJ -Suspend Objects**.
- SCT—The contents of an object could not be replicated.
- SDQ—The send operation to a data queue failed.
- SFD—Unable to retrieve the file identifier of a BSF object.
- SIS—A suspend entry was issued for an object.
- SIZ—An object was suspended on the primary node. A refresh of the object was required, but the size of the object was greater than the value set in the **Maximum refresh size** cluster system value.
- SND—An object or its authorities could not be replicated.
- SPF—A logical file was suspended on the primary node because the associated physical file was suspended.
- SPL—An *OUTQ object suspended spooled files. The *OUTQ object itself is not suspended and will not be included in auto-reactivation processing (if enabled).
- SRF—BSF object refresh failed.
- SSC—A BSF object was suspended on the primary node.

-
- STR—Journaling for a BSF object was started. The object must be refreshed.
 - SVF—An object was suspended on the primary node. The object should have been refreshed, but the save operation failed. This reason code appears only for the primary node.
 - SWA—Activate pending for files or objects that are being manually refreshed. The file or object is waiting for the activate command.
 - SWO—The object was suspended on the backup before a switchover.
 - TNE—The BSF object does not exist on the backup node.
 - TNR—The path of a BSF object could not be resolved on the backup node.
 - TNS—The BSF object is a type of object that cannot be replicated.
 - TRG—The trigger information for the object could not be retrieved.
 - UBI—An operation was found that could not be undone because the object in question does not have journaled before-images.
 - UCC—An error occurred undoing a content change to a journaled object.
 - UKM—The file is suspended on the backup node. The file object specifier was added with **PFKEY**(*AUTO), however a unique key could not be found for the file on the backup node.
 - UOC—An object level change was encountered that cannot be undone.
 - UPD—A member of a source physical file is open for update by another application. Automatic reactivation will reactivate the file after the application closes the file.
 - UUT—The object type in question is not eligible for undoing.

Out-of-sync reason codes

The out-of-sync (OOS) status is shown as two-character or three-character code. The out-of-sync reason codes represent an attribute that has been found to be out-of-sync for the object.

The out-of-sync reason codes and their meanings are as follows:

- ACS—Allocate & Contiguous Storage
- AD—Allow Delete Indicator
- AFN—Alternative Field Name
- AP—Access Path Indicator
- APJ—Access Path Journal Indicator
- APM—Access Path Maintenance
- APR—Access Path Recovery
- APS—Access Path Size Indicator
- APT—Access Path Type
- AR—Allow Read Indicator
- ASP—Auxiliary Storage Pool
- AST—Alternative Sequence Table
- AU—Allow Update Indicator
- AUL—Authorization list
- AUT—authority info
- AW—Allow Write Indicator
- BOF—Based-on file(s)
- CBU—Created by user
- CCS—CCSID For Text Description or BSF object CCSID
- CDP—codepage

-
- CH1—Column Heading 1
 - CH2—Column Heading 2
 - CH3—Column Heading 3
 - CHC—Field Column Heading CCSID
 - CID—Country Identifier
 - CNR—Current Number Of Records
 - CNT—Contents
 - CRD—Creation date&time
 - DAL—Data area length
 - DBD—DBCS Or Graphic Data
 - DBL—DBCS Or Graphic Literals
 - DCP—Decimal Position
 - DES—File or Object Text Description
 - DSO—Dynamic Select/Omit Indicator
 - DTD—Date/Time Data Indicator
 - DTF—Date And Time Format
 - DTI—Date And Time Field Indicator
 - DTS—Date And Time Separator
 - DTY—Field or data area data Type
 - EC—Edit Code
 - EW—Edit Word
 - EWC—Field Edit Word CCSID
 - EXA—File/Object Extended Attribute
 - EXP—Expiration Date
 - FAS—Field Data CCSID
 - FDC—Field Text Description CCSID
 - FDS—Field Text Description
 - FKA—Force Keyed Access Path
 - FPK—Floating Point Key Indicator
 - FLN—Field Length In Bytes
 - FNM—Field Name
 - FUS—Field Usage
 - GD—Graphic Data Indicator
 - GKL—Generic Key Length
 - GKN—Generic Key Field Count
 - GRF—Graphic Field Indicator
 - IBP—Input Buffer Position
 - ICR—Increment Number Of Records
 - IFN—Internal Field Name
 - INR—Initial Number Of Records
 - JFT—Join File Type
 - JLFI—Join Logical File Indicator
 - JM—Join Member
 - JRN—Journaling
 - LCK—Object not available (locked)

-
- LF—Logical File Indicator
 - LNG—Language Identifier
 - LVL—File Level Identifier
 - MBD—Member Text Description
 - MBN—File Member Name
 - MCC—Multiple CCSID Indicator
 - MDC—Member Text Description CCSID
 - MDR—Maximum Deleted Records
 - MFW—Maximum File Wait Time
 - MKL—Maximum Key Length
 - MMB—Maximum Members
 - MNF—Maximum Number Of Fields
 - MNI—Maximum Number Of Increments
 - MRL—Maximum Record Length
 - MRW—Maximum Record Wait Time
 - NAU—Number of auth. users
 - NBM—Number of Based-on Members
 - NBO—Number of based-on files
 - NCI—Number of Constraint Indexes
 - NCF—Null-capable Field Indicator
 - NCO—Number of Constraints
 - NDC—Number of DBCS Characters
 - NDG—Number of Digits
 - NDM—Number of Data Members or BSF mode
 - NDR—Number of Deleted Records
 - NFD—Not found on backup
 - NFL—Number of Fields
 - NKF—Number of Key Fields
 - NRF—Number of Record Formats
 - NSF—Number of spooled files
 - NSO—Number of Select And Omit statements
 - NTR—Number of Triggers
 - NVA—Null Values Allowed
 - NVD—Null Value Duplicate Indicator
 - NVI—Null Value Data Indicator
 - NVK—Null Value Key Indicator
 - OBP—Output Buffer Position
 - OBS—Not found on the primary node
 - ODP—Open Data Path Sharing
 - OWN—Object owner
 - PAC—Public Authority At Creation
 - PDD—Program Described Indicator
 - PGP—Primary group
 - PK—Primary Key Indicator
 - PSU—Preferred Storage Unit

-
- RDC—Record Text Description CCSID
 - RDR—Reuse Deleted Record Indicator
 - RFI—Record Format ID
 - RFL—Record Format Level Check
 - RFN—Record Format Name
 - RFS—Record Format Selector Program
 - RFW—Records To Force A Write
 - RIM—Replication Information Missing
 - RLN—Record Length
 - RMF—Remote File Indicator
 - RTD—Record Text Description
 - SC—Substitute Character Indicator
 - SCC—CCSID For Select/Omit Constant
 - SCT—CCSID For The Table
 - SEN—DLO Sensitivity level
 - SFT—Source Type
 - SOI—Select/Omit Indicator
 - SPF—Source Physical File Indicator
 - SQLI—SQL Index Indicator
 - SQL—SQL Table Indicator
 - SQT—SQL File Type
 - SQV—SQL View Indicator
 - SRF—Source File
 - SRM—Source File Member Name
 - SSC—UCS2 Sort Sequence Table CCSID
 - SSL—UCS2 Sort Sequence Table Len
 - SST—Sort Sequence Table Name
 - SSW—SST Weight Indicator
 - SZE—object data size mismatch
 - TMS—Total of Member Sizes
 - TNM—Total Number Of Members
 - TRG—Trigger Data (System)
 - TYP—Object type
 - UC—Unique Constraint Indicator
 - UFL—UCS-2 Displayed Field Length
 - URD—Use Reset Date
 - VF—Variable Field Indicator
 - VL—Variable Length Indicator
 - VLD—Variable Length Data Indicator
 - VR1—First Supported Version Release
 - VRM—Version Release And Mod Level
 - WCO—With Check Option

Viewing and changing object status

View object details and perform operations on the **Work with Object Status** screen.

The **Work with Object Status** screen displays information for the objects that belong to the group/journal combination that was selected from status monitor screen. The information is displayed across multiple views.

1. To start the **Work with Object Status** screen, type **8** in the **Opt** column for a group-journal entry in the status monitor and press Enter.
2. **Optional:** To cycle to the next view, press F16 (Shift + F4).
The views cycle from suspended objects only, active objects only, and all objects. The available options are updated for each view.
3. **Optional:** To change the sort order, press F11.
The default sort order is by library (Sort by Lib). To sort by file name, press F11 again (Sort by Object). To sort by file type (Sort by Type), press F11 again.
4. To perform the operation in the **Work with Object Status** screen, type one of the following options in the **Opt** column for an object:

Option	Action
1	Activate a suspended object.
4	Suspend an object.
6	Display journal entry that resulted in the object being suspended. Available only on the primary node for the suspended objects.
9	Display event log messages.
10	Display the Work with Suspended Spooled Files screen.
56	Start a sync check for the specific object in the group.

Working with BSF and SQL objects

View object details and perform operations on the **Work with BSF/SQL Status** screens.

1. From a status monitor, type **7** in the **Opt** column for the group or journal.
2. **Optional:** To cycle to the next view, press F16 (Shift + F4).
The views cycle from suspended objects only to active objects only. The available options are updated for each view.
3. To perform the operation in the **Work with BSF/SQL Status** screen, type one of the following options in the **Opt** column for an object:

Option	Action
1	Activate a suspended object.
5	Display the full path name of the object.
6	Display the journal entry that caused the object to be

Option	Action
	suspended.
56	Start a sync check for the specific object in the group.

Historical Status Monitor views

Historical Latency view

Use the historical latency statistics to view previous replication activity, activity snapshots, and job start and end statistics. The iCluster history log displays historical information for a journal.

The **Historical Latency** view displays the following information across multiple views:

- **Type**—The type of entry. Existing activity snapshots and ending records for the selection are displayed in ascending order by the creation timestamp. An asterisk (*) denotes that a portion of the data that is required to display the entry is unavailable.
 - **STR**—The entry is the first journal entry that was processed.
 - **END**—The entry is the last journal entry that was processed.
 - **MON**—The entry is the current journal entry that is being processed.
- **Date**—The date when the activity occurred or the end record was defined.
- **Time**—The time when the activity occurred or the end record was defined.
- **Last J/E**—The last journal entry written to the journal.
- **Primary J/E**—The journal entry where the journal scraper process is scraping the journal.
- **Backup J/E**—The last journal entry applied to the backup node by the apply process.
- **Trans to Process**—The number of transactions that were sent by the primary node, but are not applied on the backup node.

Historical Object Position and Totals view

Use the **Historical Object Position and Totals** view to display the historical journal position and totals for a group and journal combination. Historical information is obtained by reading the records that were created by the batch collection process.

For the selected group/journal combination, the screen displays the name of the backup node, group, journal, and library.

The **Historical Object Position and Totals** view displays the following information:

- **Type**—The type of entry. Existing activity snapshots and ending records for the selection are displayed in ascending order by the creation timestamp. An asterisk (*) denotes that a portion of the data that is required to display the entry is unavailable.
 - **STR**—The entry is the first journal entry that was processed.
 - **END**—The entry is the last journal entry that was processed.
 - **MON**—The entry is the current journal entry that is being processed.
- **Date**—The date when the activity occurred or the end record was defined.
- **Time**—The time when the activity occurred or the end record was defined.
- **Backup J/E**—The last journal entry that was applied by the apply process.

- **Total Trans Sent**—The total number of transactions that were processed since the last reset of the transaction counters. These transactions will be sent to the backup node (entries that are associated with files to replicate). This number does not include omitted entries. Transaction counts are reset under the following conditions:
 - After issuing the DMSETPOS (Set Journal Start Position), DMMRKPOS (Mark Journal Positions), or DMREGPOS (Register Positions) commands.
 - Refresh before mirroring.
 - Counts exceed approximately 10 billion.
- **Total Trans Applied**—The total number of transactions that were processed by the backup node since the last reset of the transaction counters. These transactions were sent to the backup node and applied by the apply process.
- **Trans to Process**—The last journal entry that was applied to the backup node by the apply process.

Related information

[DMMRKPOS \(Mark Journal Positions command\)](#)

[DMSETPOS \(Set Journal Start Position command\)](#)

Historical Object Throughput view

Use the **Historical Object Throughput** view to display the historical throughput for a group and journal combination. Historical information is obtained by reading the records created by the batch collection process.

Existing activity snapshots and ending records for the selection are displayed in ascending order by the creation timestamp in the **Type** column. An asterisk (*) in this column denotes that a portion of the data required to display the entry is unavailable.

This screen lists all the historical activity for the group/journal combination. It displays the name of the backup node, as well as the name of the journal previously in use.

This screen displays the following items of information that are briefly described along with the **Position to date** and **Position to time** fields:

- **Type**—The type of entry. Existing activity snapshots and ending records for the selection are displayed in ascending order by the creation timestamp. An asterisk (*) denotes that a portion of the data that is required to display the entry is unavailable.
 - **STR**—The entry is the first journal entry that was processed.
 - **END**—The entry is the last journal entry that was processed.
 - **MON**—The entry is the current journal entry that is being processed.
- **Date**—The date when the activity occurred or the end record was defined.
- **Time**—The time when the activity occurred or the end record was defined.
- **Elapsed - Days Hr Mn**—The elapsed time since the latest replication (mirroring) process began for the backup node/journal combination. This timestamp is used to calculate throughput.
- **Overall Trans/Hr**—The number of transactions that were performed in an hour, based on the life of the backup job.
- **Current Trans/Hr**—The number of transactions that were performed in an hour, based on the life of the backup job.

Viewing the Historical Status Monitor

Use the iCluster history log to display historical information for a journal. Historical information is valuable to assess the replication environment with a perspective of time-sensitive or date-sensitive transactions.

1. To start the Historical Status Monitor screen, type **12** in the **Opt** column for a journal in the status monitor and press Enter.
2. **Optional:** To cycle to the next view, press F11.
The views cycle from **Historical Latency**, **Historical Object Position and Totals**, and **Historical Object Throughput**.
3. **Optional:** To refresh the view, press F5.
4. **Optional:** To move to a specific date in the history list, in the **Position to date** field, type the date in local system date format and press Enter.
5. **Optional:** To move to the next occurrence of a specific time in the history list, in the **Position to time** field, type the time and press Enter.

Changing the Historical Latency collection process

To improve performance, change the frequency of the Historical Latency collection process. If the Historical Latency process reporting frequency is set to a high rate, the replication performance might be adversely affected.

Historical latency statistics in the Historical Status Monitor show previous replication activity, activity snapshots, and job start and end statistics.

1. On the primary node, enter the following command:
`CHGHASMON`
2. In the **Backup node name** field, specify where to start the Historical Latency collection.
3. In the **Group name** field, specify where to collect the data.
Choose one of the following values:
 - <name>—Specifies the group name for replication monitoring.
 - *ALL—Specifies all groups replicating to the backup node specified in the TARGET parameter.
4. Press Enter.
5. In the **Start date** field, specify the start date for replication monitoring.
Choose one of the following values:
 - <date>—The date to start monitoring.
 - *TODAY—Start monitoring on the current date. By default, this value is used.
 - *SAME—Keeps the current setting for this parameter.
6. In the **Start time** field, specify the start time for replication monitoring.
Choose one of the following values:
 - <time>—A time value in *HHMMSS* format, 000000 - 235959.
 - *IMMED—Start monitoring immediately if the **STRDATE** parameter is set to *TODAY or an earlier date. If the starting date is set to a later date, monitoring starts at 000000 on the specified date. By default, this value is used.
 - *SAME—Keeps the current setting for this parameter.
7. In the **Polling time interval** field, specify the polling interval for monitoring replication.
Choose one of the following values:
 - <polling interval>—A time value in *HHMMSS* format, 000010 - 235959. The default value is 001500 (15 minutes).
 - *SAME—Keeps the current setting for this parameter.
8. In the **Ending date** field, specify the ending date for monitoring replication.
Choose one of the following values:
 - <date>—The date to end monitoring, expressed in local system date format.
 - *TODAY—The current date.
 - *NONE—Stop monitoring after all replication processes end. By default, this value is used.

- *SAME—Keeps the current setting for this parameter.
9. In the **Ending time** field, specify the ending time for monitoring replication.
Choose one of the following values:
- <time>—A time value in HHMMSS format, 000000 - 235959.
 - *NONE—Stops monitoring after all replication processes end. By default, this value is used.
 - *SAME—Keeps the current setting for this parameter.
10. Press Enter.

Related information

[CHGHASMON \(Change History Monitor on Primary Node command\)](#)

Deleting historical information from the Historical Latency

Delete historical information from the history log. For example, delete historical information if disk utilization is too high, or if the historical information for a previous year is no longer required.

1. On the primary node, enter the following command:
PRGHASMON
2. In the **Backup system** field, specify the name of the backup node for the records to remove.
Choose one of the following values:
 - <name>—Specifies the backup node name.
 - *ALL—Specifies all backup nodes.
3. In the **Ending date** field, specify the latest date for records to delete.
Choose one of the following values:
 - <date>—A date in the local system date format.
 - *TODAY—Specifies all dates. By default, this value is used.
4. In the **Ending time** field, specify the latest time for records to delete.
Choose one of the following values:
 - <time>—Specifies a time value that is expressed in HHMMSS format, from 000000 - 235959.
 - *NOW—Specifies the current time. By default, this value is used.

Related information

[PRGHASMON \(Purge History Monitor on Primary Node command\)](#)

Reporting

You can view or generate reports from the iCluster Reports Menu.

The available reports include:

- **iCluster Configuration Report (DMCFGRPT command)**: Use this command to generate a detailed report of the iCluster configuration. You can also filter the report to include information on nodes, groups, object specifiers, simple object specifiers, system values, and output.
- **iCluster System Report (DMSYSINF command)**: Use this command to generate a detailed report of system information, which includes cluster nodes, system name, IP address, operating system and iCluster software versions, hardware information, and so on.
- **iCluster Group Status Report (DMGRPSTS command)**: Use this command to view a summary, and optional detailed information, for one or all replication groups in the cluster.
- **iCluster Sync Check Report (DMSCRPT command)**: Use this command to generate the iCluster Sync Check Report on the backup node to view the sync check results.
- **iCluster Replication Coverage Report (DMRPLCVRPT command)**: Use this command to generate a report that lists the replication coverage of objects (native objects and BSF objects) replicated by a group or groups owned by iCluster, user libraries, IFS directories, and DLO folders on the current node.
- **iCluster Event Log Report (DMDSPLOG command)**: Use this command to generate a report that lists messages in the iCluster event log.
- **Files with Triggers Report (DMRPTTRG command)**: Use this command to generate a report that lists the files with triggers on the current node that are in replication scope of the specified group or groups, and have the current node as the primary node.
- **Files with Constraints Report (DMRPTCST command)**: Use this command to generate a report that lists the files with constraints on the current node that are in the replication scope of the specified group or groups, and have the current node as the primary node.
- **iCluster Suspended Objects Report (DMSUSOBRPT command)**: Use this command to generate a report which lists the objects that are currently in suspended state in your iCluster replication configuration.
- **Journal Analysis Report (DMANZJRN command)**: Use this command to generate a report that lists the database journal transactions by time, object, job, and application.
- **Cross-Library Logical Files Report (DMRPTXLF command)**: Use this command to generate a report that lists the cross-library logical files and physical files.
- **Object Counts Report (DMOBJCNT command)**: Use this command to generate a report that lists the object counts and sizes for the specified library or libraries. Object counts are useful in the initial configuration to identify the objects to replicate.
- **iCluster Audit Report (DMAUDITRPT command)**: Use the command to generate a report that lists the discrepancies between the user objects on the current machine as the backup node, and the primary node specified on the command.

Related information

- [**DMCFGRPT \(Configuration Report command\)**](#)
- [**DMSYSINF \(System Information command\)**](#)
- [**DMGRPSTS \(Group Status Report command\)**](#)
- [**DMSCRPT \(Sync Check Report command\)**](#)
- [**DMRPLCVRPT \(iCluster Replication Coverage command\)**](#)
- [**DMDSPLOG \(Display Cluster Event Log command\)**](#)
- [**DMRPTTRG \(Report Files with Triggers command\)**](#)

-
- [DMRPTCST \(Report Files with Constraints command\)](#)
 - [DMSUSOBRPT \(Suspended Object Report command\)](#)
 - [DMANZJRN \(Analyze Journals command\)](#)
 - [DMRPTXLF \(Report Cross-library LFs command\)](#)
 - [DMOBJCNT \(Report Object Counts command\)](#)
 - [DMAUDITRPT \(Audit Report command\)](#)

iCluster Configuration Report (DMCFGRPT command)

Use the DMCFGRPT (iCluster Configuration Report) command to generate a detailed report of the iCluster configuration. Filter the report to include information on nodes, groups, object specifiers, system values, and so on.

Example: iCluster Configuration Report sample for nodes

The node configuration includes the description, port, product library, job description and library, user profile replication status, staging store information, and other node replication settings. The following image shows a sample node configuration report.

File	DMCFGRPT	Page/Line	1/9
Control	_____	Columns	1 - 78
Find	_____		

icluster Configuration Report

Node ICTST61A

Description	4545
Port	ICLUSTER
icluster product library	CSJOB
Job description	ICLUSTER
Library	
Replicated user profile status	*DISABLED
Hold config obj src on backup	*YES
Staging store size (in MB)	40960
Staging store library	DMSTORE
Check primary link	*YES
Link check frequency (sec)	60
Link check reply timeout (sec)	30
Maximum link check retries	5

Example: iCluster Configuration Report sample for groups

The group configuration includes the group type, primary and backup nodes, target library, staging store library, journal location, and other group settings. The following image shows a sample group configuration report.

File : : DMCFGRT
Control : : _____
Find : :

Page/Line 1/10
Columns 1 - 78

icluster Configuration Report

Group MGWTTEST

Group type	*REPL
Description	
Primary node	ICTST61A
Primary IASP device name	*SYSBAS
Backup node	ICTST71B
Backup IASP device name	*SYSBAS
Target library	*PRIMARY
Staging store library	DMSTORE
Journal location	*REMOTE
Failover message queue name	*NONE
Do roleswitch at failover	*NO
Check journaling at roleswitch	*YES
User exit before roleswitch	*NONE

Related information

DMCFGRPT (Configuration Report command)

iCluster System Report (DMSYSINF command)

Use the DMSYSINF (System Information) command to generate a detailed report of system information, which includes cluster nodes, system name, IP address, operating system and iCluster software versions, hardware information, and so on.

Review the system report information that is described in the following table.

System information	Field
Node on which the command is run	Local Node Name
System on which the command is run	Local System Name
Installed iCluster version where this command is run	iCluster version
Failover mechanism is SwitchOverSystem when Use Cluster Resource Services is *NO.	Use Cluster Resource Services
Node names	Node
System name	System
Node IP addresses	IP address
IBM i operating system version	OS Rls
IBM Power Systems server type and model number	Type-Mod
IBM Power Systems Processor Group	P-Grp
IBM Power Systems server hardware serial number	Serial #
System time that the command was run	Sys Time

System information	Field
The disk space in MB that the staging store uses, including node and group level (if any)	St. Store Used (MB)

Example:Rocket iCluster System Report sample

```
Rocket iCluster System Report
14/08/14 14:10:59
Local Node Name : ICDM071A
Local System Name: ICDM071A
-----
iCluster Version: ic 7.1.2033.0 TR 2
Use Cluster Resource Services: *NO
-----
Node   System   IP Address          OS Rls Type-Mod P-Grp Serial # Sys Time St. Store Used (MB)
ICDM071A ICDM071A 10.17.8.153    V7R1MO 8233-E8B P20 065382R 14:10:59      0.000
ICDM071B ICDM071B 10.17.8.154    V7R1MO 8233-E8B P20 065382R 14:11:01      0.000
```

Related information

[DMSYSINF \(System Information command\)](#)

iCluster Group Status Report (DMGRPSTS command)

Use the DMGRPSTS (Group Status Report) command to view a summary, and optional detailed information, for one or all replication groups in the cluster.

The report lists the following summary information for each group:

- Group name
- Group type (*REPL, *HADR, *MQSERIES)
- Group status
- Switchover readiness for replication and MQSeries groups
- Sync check active
- Primary and backup nodes
- Group's staging store library
- Amount of data, in megabytes, in the staging store library. If the library is the node-level staging store library, the amount shown is the amount of data for all the groups using the node-level staging store.
- Description

Specify the report to provide optional details for each group journal combination. Optional details provide the following information with one entry for each group journal combination in a separate **Group Details** section:

- Journal name and library
- Remote journal status, if applicable
- Journal scrape-receive process status
- Backup apply process status
- Latency for scrape-receive process
- Latency for backup apply process
- Total latency
- Number of suspended objects
- Number of out-of-sync objects

Specify the report to provide optional journal position details for each group journal combination. Optional journal position details provide the following information with one entry for each group journal combination in a separate **Journal Positions Report** section.

- Journal name and library
- Last journal entry as written by Primary node applications
- Current receiver and receiver library being written on Primary node
- Current receiver entry being scraped by iCluster on Primary node
- Current receiver and receiver library being scraped on Primary node
- Current receiver entry being applied by iCluster on Backup node
- Current receiver and library being applied on Backup node

Example: iCluster Group Status Report sample

The following image shows the status report for group IVGRP with the group-journal details.

iCluster Status Report for Group(s) IVGRP											Page 1	16/06/16 08:58:01
Group	Grp type	Status	SwRdy	Synck	Num Sus	Num OOS	Primary	Backup	Store Lib	MB Stored	Description	
IVGRP	*REPL	*ACTIVE	*NO		2		ICDEV54A	ICDEV71A	IVGRPSTORE	0.000		
Group-journal Details Report												
Group	Journal	Jrn Lib	RmtJrnsts	Scrape	Apply	Src Ltny	Apy Ltny	Total Ltny	Num Ltny	Num Susp	Num OOS	
IVGRP	HADJRN	ICLUSTER	*ACTIVE	*YES	*YES	00	00	00	00	2		
IVGRP	QAUDJRN	QSYS		*YES	*YES	00	00	00	00			
Journal Positions Report for Group IVGRP												
Journal	Jrn lib	Last journal entry	Receiver	Rcvr lib	Scraper	jrn entry	Receiver	Rcvr lib	Apply jrn entry	Receiver	Rcvr lib	
HADJRN	ICLUSTER	9158 HADRCV1718	ICLUSTER				9158 HADRCV1718	DWRMTJRN		9158 HADRCV1718	DWRMTJRN	
QAUDJRN	QSYS	93352 QAUDJR9696	QSYS				93351 QAUDJR9696	QSYS		93351 QAUDJR9696	QSYS	

Related information

[DMGRPSTS \(Group Status Report command\)](#)

iCluster Sync Check Report (DMSCRPT command)

Use the DMSCRPT (Sync Check Report) to generate the iCluster Sync Check Report on the backup node to view the sync check results.

This sync check report includes details for groups with the local node as their backup node. The report lists the group name, the primary node name, the group status, the sync check type, and the number of checked and failed objects. The report will also list the objects that were locked when the sync check was run. The report is based on the sync check metadata. Before you run the report, run at least one sync check for a group with this backup node.

The report is displayed by default, unless the command is submitted to batch, as well as to a spooled file. The sync check report is sent to output queue that is listed in the CSJOB job description in the ICLUSTER library on the group's backup node. If the output queue listed in the CSJOB job description on the backup node does not exist on the node to which the report is sent, the report is put into the output queue specified for spooled files received through a network (use the DSPNETA command to see the name of this output queue). Optionally, the report can be sent only to a spooled file.

Example: iCluster Sync Check Report sample

The following command produces this sample report:

```
DMSCRPT GROUP(HYMIRRMTS) OUTPUT(*PRINT) DETAILS(*YES)
```

06/15/16 08:16:43

icluster Sync Check Report

Page 1

Group name: HYMIRRMTS
 Backup node: ICDEV71B
 Number of checks with oos objects : 2
 Number of sync checks reported ...: 4

List of Completed Sync Checks

Group	Primary node	Group status	Sync check type	# checked	# failed	# unchecked	SC end date/time	SC duration
HYMIRRMTS	ICTST61A	*ACTIVE	*OBJATTR	31	0	0	06/15/16 08:16:36	00:00:17
HYMIRRMTS	ICTST61A	*ACTIVE	*CHECKSUM	2	0	1	06/15/16 08:16:03	00:00:07
HYMIRRMTS	ICTST61A	*ACTIVE	*FILEATTR	4	0	1	06/15/16 08:16:12	00:00:02
HYMIRRMTS	ICTST61A	*ACTIVE	*CHKOBJSLET	31	0	0	06/15/16 08:16:43	00:00:01

Out of sync or unavailable object List

Group	Path Name or Object	Source Library	Target library	Object type	Extended attr	size (KB)	Not found	Reason code	SC end date/time
HYMIRRMTS	DBLOT	HYMIRSRC1	HYMIRSRC1	*FILE	PFDTA	44		LCK	06/15/16 08:16:03
HYMIRRMTS	DBLOT	HYMIRSRC1	HYMIRSRC1	*FILE	PFDTA	0		ERR	06/15/16 08:16:12

Sync check report complete.

Related information

[DMSCRPT \(Sync Check Report command\)](#)**iCluster Replication Coverage Report (DMRPLCVRPT command)**

Use the DMRPLCVRPT (iCluster Replication Coverage) command to generate a report that lists the replication coverage of objects (native objects and BSF objects) replicated by a group or groups owned by iCluster, user libraries, IFS directories, and DLO folders on the current node.

Specify the replication coverage type, and how to sort the report output.

Library objects are listed separately. The report truncates directory path names that are longer than 50 characters. Truncated path names are assigned a numerical identifier, and full path names are listed in a mapping table at the end of the report.

The latest report data is in the **DMRPLCVG** database file in the **ICLUSTER** library. Query the **DMRPLCVG** database file to create replication coverage lists.

The report displays libraries that are owned by the system, but these libraries are always marked as ineligible for replication.

Replication ineligibility reason codes

The replication ineligibility reason codes in the **RC** column are numbers that correspond to the following reasons:

- The object is owned by the **DMCLUSTER** user profile.
- The object is owned by a Rocket LM(i) user profile and is not eligible for LM(i) high availability with iCluster.
- The object is a licensed program product library or directory.
- The object is an iCluster work library.
- The object is an iCluster staging store library or directory.
- The object is an iCluster target library or directory for a replication group that has the current node as its backup node.
- The library contains only journals and journal receivers.
- The directory is in a file system that is not supported for replication.
- The library is owned by the system.

Example: iCluster Replication Coverage for Libraries report-libraries not replicated

The following sample report shows the replication coverage for libraries that are not replicated.

iCluster Replication Coverage for Libraries

Library	IASP	dev	Elig	RC	Repl	Group	Backup	Owner	Created	# obj	Attr Description
\$\$HYLIB	*SYSBAS	*YES	*NO			HYU_X	2012/09/11		1 PROD		
ACMSCTL	*SYSBAS	*YES	*NO			ALDONCMS	2013/06/13	598	PROD Aldon/CMS Control Library		
APLISTLIB	*SYSBAS	*YES	*NO			HYU_X	2012/05/17		2 PROD		
BACKUP	*SYSBAS	*YES	*NO			BACKUP	1989/03/28	111	PROD Backup holds objects used in the backup		
BACKUPLIST	*SYSBAS	*YES	*NO			BACKUP	2010/12/14	16	PROD Library to hold backup listings and outfi		
CAMOL	*SYSBAS	*YES	*NO			HYU_X	2013/08/08	10	PROD		
CFGHAXAPPS	*SYSBAS	*YES	*NO			ILZEV_X	2012/10/11	12	PROD		
DMCLUSTER	*SYSBAS	*YES	*NO			ILZEV_X	2012/06/30	562	PROD DM Product Utility Library		
DMTHA	*SYSBAS	*YES	*NO			HYU_X	2012/04/14	4	PROD *DND needed for some iCAT testcases		
DMXHA	*SYSBAS	*YES	*NO			MARKW_X	2012/01/30	512	PROD DMXtras -- Customised version		
DMXIC50TL1	*SYSBAS	*YES	*NO			MARKW_X	2012/03/01	606	PROD copy of ICLUSTER V5.2 TL1		
DMXIC500	*SYSBAS	*YES	*NO			MIKEW_X	2012/01/19	606	PROD copy of icluster library - for compilat		
DMXIC501	*SYSBAS	*YES	*NO			MARKW_X	2012/10/29	335	PROD copy of ICLUSTER V7.1 IF2		
DMXIC710	*SYSBAS	*YES	*NO			MARKW_X	2012/10/29	1238	PROD Copy of icluster V7.1 TR1		
DMXPATCH	*SYSBAS	*YES	*NO			MARKW_X	2013/06/10	7	PROD		

Related information

[DMRPLCVRPT \(iCluster Replication Coverage command\)](#)

iCluster Event Log Report (DMDSPLOG command)

Use the DMDSPLOG (Display Cluster Event Log) command to generate a report that lists messages in the iCluster event log.

Use the following filters to specify which messages to include in the event log:

- **Event type**
- **Date range**
- **Time range**
- **Message IDs**
- **Message type**
- **Level of detail**

Related information

[DMDSPLOG \(Display Cluster Event Log command\)](#)

Files with Triggers Report (DMRPTTRG command)

Use the DMRPTTRG (Report Files With Triggers) command to generate a report that lists the files with triggers on the current node that are in replication scope of the specified group or groups, and have the current node as the primary node.

The report lists the number of database files that are processed for each group. The report also lists the ASP group, file name, library and type, trigger state, current status, time and event, update condition, trigger program name and library, whether repeat changes are allowed, whether the program is thread safe and the multi-threaded job action, trigger type, and the trigger name and library.

Example: Files with Triggers Report sample

The following images show a sample of the report.

***** GROUP @GWDBMARK, number of files processed: 86 *****

No triggers to report for files replicated by group @GWDBMARK.

***** GROUP @PAYROLL, number of files processed: 56 *****

** PAYROLL/ACUSTOMER: 1 triggers **

ASP group	File Library	PF or Trig	LF State	Trig Sts	Time	Upd Evnt	Cond	Program Library	Program Name	Alw Chg	Rpt Saf	Multi Thd JobAc	Trigger Type	Trigger Library	Trigger Name
-----------	--------------	------------	----------	----------	------	----------	------	-----------------	--------------	---------	---------	-----------------	--------------	-----------------	--------------

*SYSBAS	PAYROLL	ACUSTOMER	PF	Disabl	Oper	Bef	Upd	Alws	PAYROLL	CUSTUPD2	No	Run	NTV	PAYROLL	QSYS_TRIGGER_PAYROL
---------	---------	-----------	----	--------	------	-----	-----	------	---------	----------	----	-----	-----	---------	---------------------

***** GROUP AHVTDB1, number of files processed: 0 *****

No triggers to report for files replicated by group AHVTDB1.

***** GROUP AWDIFS1, number of files processed: 0 *****

No triggers to report for files replicated by group AWDIFS1.

***** GROUP AWDIFS2, number of files processed: 0 *****

No triggers to report for files replicated by group AWDIFS2.

Related information

[DMRPTTRG \(Report Files with Triggers command\)](#)

iCluster Audit Report (DMAUDITRPT command)

Use the DMAUDITRPT (Audit Report) command to generate a report that lists the discrepancies between the user objects on the current machine as the backup node, and the primary node specified on the command.

The report also provides an audit reason code related to the replication status of each object in the list. If an object is in the replication scope of a group replicating from the specified primary node to the current node, or if it is deliberately excluded from replication by a group that is replicating between the two nodes, the group to which the object is selected will be listed in the audit report.

Any object that is listed in the report exists on only one of the nodes being checked, either the primary node or the backup node. The node where the object exists is listed in the report. No objects that exist on both nodes are listed in the report.

This command can be issued on any node in the cluster, as long as this node is the backup node for all groups that replicate from the specified primary node to the node where the command is invoked. All groups of type *REPL and *MQSERIES that replicate from the specified primary node to the current node must be active. None of the groups that replicate from the specified primary node to the current node can use target library or ASP device redirection for the objects on the backup node. In other words, the objects on the backup node must be on the same ASP device and in the same library as on the primary node.

Pathnames longer than 80 characters cannot be listed directly in the report. For these path objects, the report will give a truncated version of the pathname, along with a numerical identifier for the full pathname. The full pathnames of truncated pathnames are listed at the end of the report in a mapping table.

The results of the audit can also be found in the database file that is specified on the command. If the audit results file does not exist in the specified library when the command is run, it is created, otherwise it is cleared of data when

the command processing starts. Write queries to extract the records from a selected file to examine further or, create your own audit report.

- ! **Note:** Your IBM i user profile must be enrolled in the system directory in order to use the DMAUDITRPT command for DLO folders, and it is recommended that your user profile has *ALLOBJ authority in order to have access to all libraries and directories on the system.

The following audit reason codes are used for the objects listed in the report:

- DLY: The object is in replication scope, but is out of sync. Its replication may be delayed due to replication latency that exceeds the latency threshold.
- EXC: The object is deliberately excluded from replication.
- NFD: The object is in replication scope but is only found on the primary node.
- OBS: The object is in replication scope but is only found on the current (backup) node.
- NRP: The object can be replicated but it is not in the replication scope of any group.
- NVR: The object is a type that cannot be replicated by iCluster.

- ! **Note:** Only groups that are currently defined as replication groups, that is, groups of type *REPL, *RFSH, or *MQSERIES, are considered when determining the audit reason code for an object that only exists on one of the nodes being audited. *STDBY groups are not considered when determining the audit reason code.

The audit report lists the user objects (library objects and DLO objects) and the user IFS files that are out of sync between the two systems being checked. Note that objects which fit any of the following criteria are not eligible for the audit report and will not be listed:

- The object is owned by a system profile. (For the full list of system profiles see the IBM Knowledge Center at <http://www.ibm.com/support/knowledgecenter>.)
- The object is owned by the DMCLUSTER user profile.
- The object is owned by one of the Rocket Aldon LM(i) user profiles and is not eligible for Aldon LM(i) high availability with iCluster.
- The object is in a licensed program product library or directory.
- The object is in an iCluster work library.
- The object is in an iCluster staging store library or directory.
- The object is a journal receiver.
- The path object is in a file system that is not supported for replication.
- The library used is a system-owned library.

This command must be invoked on a node in the cluster, as long as this node is the backup node for all groups that replicate from the specified primary node to the node where the command is invoked on an active node in the cluster.

Example: iCluster Audit Report sample

Primary node . . . ICDEV71A
 Backup node . . . ICDEV71B

Audit report reason codes:

DLY: The object is out of sync but its replication may be delayed due to replication latency that exceeds the latency threshold.
 EXC: The object is deliberately excluded from replication.
 NFD: The object is in replication scope but is only found on the primary node.
 OBS: The object is in replication scope but is only found on the current (backup) node.
 NRP: The object can be replicated but it is not in the replication scope of any group.
 NVR: The object is a type that cannot be replicated by iCluster.

iCluster Audit Report for Native objects

Node	IASP	dev	Library	Object	Obj type	Obj attr	RC	Group	Description
ICDEV71A	*SYSBAS	AAAJRNLIB	COSPJRN	*JRN		NVR			
ICDEV71A	*SYSBAS	AAAJRNLLOC	PAYJRN	*JRN		NVR			
ICDEV71A	*SYSBAS	AAAJRNRMT	PAYJRN	*JRN		NVR			
ICDEV71A	*SYSBAS	ABC	PD	*FILE	PF	NRP			
ICDEV71A	*SYSBAS	ABC	PF	*FILE	PF	NRP			
ICDEV71A	*SYSBAS	APLISTLIB	HYLIST1	*DTAAARA		NRP			
ICDEV71A	*SYSBAS	APLISTLIB	HYLIST1	*FILE	PF	NRP			
ICDEV71A	*SYSBAS	QSYS	BARCODE	*MSGQ		NRP			
ICDEV71A	*SYSBAS	QSYS	BLANHAM	*USRPRF					Work Station Message Queue
ICDEV71A	*SYSBAS	QSYS	DMIVSTORE	*LIB	PROD	NRP			NFD SYSTEM
ICDEV71A	*SYSBAS	QSYS	DNCNLSTRLF	*LIB	PROD	NRP			Rocket iCluster staging store Library
ICDEV71A	*SYSBAS	QSYS	IVTEST2	*LIB	PROD	NRP			Library created by Rocket iCluster
ICDEV71A	*SYSBAS	QSYS	IVUSER	*USRPRF					NFD SYSTEM
ICDEV71A	*SYSBAS	QSYS	IVUSRSPC	*LIB	PROD	NRP			
ICDEV71A	*SYSBAS	QSYS	IVGIAENV	*LIB	PROD	NRP			

iCluster Audit Report for Path objects

Node	IASP	dev	Pathname (pathname ID)	obj type	RC	Group
ICDEV71B	*SYSBAS	/backup/mnt/ALDEV1		*DIR		NRP
ICDEV71B	*SYSBAS	/QDLS/DOC-X/AAAAP.DOC		*DOC		NRP
ICDEV71B	*SYSBAS	/QDLS/DOC-X/BBBBBP.DOC		*DOC		NRP
ICDEV71B	*SYSBAS	/QDLS/DOC-X/BSSR.DOC		*DOC		NRP
ICDEV71B	*SYSBAS	/QDLS/DOC-X/CCC R.DOC		*DOC		NRP
ICDEV71B	*SYSBAS	/QDLS/DOC-X/CCCCP.DOC		*DOC		NRP
ICDEV71B	*SYSBAS	/core		*DOC		NRP
ICDEV71B	*SYSBAS	/dmsndobj/file1		*STMF		NRP
ICDEV71B	*SYSBAS	/dmsndobj/file2		*STMF		NRP
ICDEV71B	*SYSBAS	/etc/ibm/director/twgagent/twgagent.uid		*STMF		NRP
ICDEV71B	*SYSBAS	/home/abc/def/a.txt		*STMF		NRP

Related information

[DMAUDITRPT \(Audit Report command\)](#)

Files with Constraints Report (DMRPTCST command)

Use the command to generate a report that lists the files with constraints on the current node that are in replication scope of the specified group or groups, and have the current node as the primary node.

The report lists the number of database files that are processed for each group. The report lists the ASP group name, the file library, the file name, constraint type, constraint definition, and state, the database key field, key field name and length, constraint library, and constraint name.

Example: Files with Constraints Report sample

The following sample report pages shows the constraints for replication group IVGRP.

***** GROUP IVGRP, number of files processed: 108 *****

** IVTEST/CUST10KA: 1 constraints **

ASP Group	File Library	File Name	Constr Type	Cst Def State	Constr Fld	Key Len	Key Names	Field Library	Num Constraint	
									Field	Constraint Name
*SYSBAS	IVTEST	CUST10KA	*PRIKEY	Est Enabld	1	6	CUSTNO	IVTEST		Q_IVTEST_CUST10KA_CUSTNO_00001

** IVTEST/FILE1: 2 constraints **

ASP Group	File Library	File Name	Constr Type	Cst Def State	Constr Fld	Key Len	Key Names	Field Library	Num Constraint	
									Field	Constraint Name
*SYSBAS	IVTEST	FILE1	*PRIKEY	Est Enabld	1	20	DATA	IVTEST		Q_IVTEST_FILE1_DATA_00001
*SYSBAS	IVTEST	FILE1	*CHKCST	Est Enabld				IVTEST		Q_IVTEST_FILE1_PRIKEYCHK_00001

** IVTEST/FILE2: 2 constraints **

ASP Group	File Library	File Name	Constr Type	Cst Def State	Constr Fld	Key Len	Key Names	Field Library	Num Constraint	
									Field	Constraint Name
*SYSBAS	IVTEST	FILE2	*PRIKEY	Est Enabld	1	20	DATA	IVTEST		Q_IVTEST_FILE2_DATA_00001
*SYSBAS	IVTEST	FILE2	*CHKCST	Est Enabld				IVTEST		Q_IVTEST_FILE2_PRIKEYCHK_00001

Related information

[DMRPTCST \(Report Files with Constraints command\)](#)

Journal Analysis Report (DMANZJRN command)

Use the DMANZJRN (Analyze Journals) to generate a report that lists the database journal transactions by time, object, job, and application. This report provides insight into the activities on your system. Use the journal analysis statistics for tuning and performance analysis.

The DMANZJRN command runs only in batch mode and generates spooled files. The basic journal analysis data is stored in the DMANZJRNPF file during DMANZJRN command processing. The journal receivers that are processed for analysis are stored in the DMANZJRCV file.

The following command produced these sample reports:

```
DMANZJRN JRN(AAAJRNLIB/WTRJRN) CLRANZ(*YES)
```

DMANZJBAND

The bandwidth analysis report performs journal analysis and provides bandwidth use estimates on communication use between nodes. The report lists the total numbers and sizes of journal entries by date and hour.

08/14/14 14:39:35			iCluster Journal Analysis Report Journal Bandwidth						Page 1	
Date	Hour	Database Changes	Bytes Per Hour	3 Mbps % Line	6 Mbps % Line	9 Mbps % Line	18 Mbps % Line	36 Mbps % Line	100 Mbps % Line	

2014-08-14	11	137946	34828420	11.07166	5.53583	3.69055	1.84528	0.92264	0.33215
2014-08-14	12	219650	55443316	17.62496	8.81248	5.87499	2.93749	1.46875	0.52875
2014-08-14	13	254646	64421684	20.47910	10.23955	6.82637	3.41318	1.70659	0.61437
2014-08-14	14	260692	66554801	21.15720	10.57860	7.05240	3.52620	1.76310	0.63472

DMANZJNTVO

Native object report lists the library, the object name, the object type, the transaction date, and accumulated transactions by hour, and the total size of the transactions.

08/14/14 14:39:35

iCluster Journal Analysis Report Native Objects

Library	Object	Type	Date	Hour	# of transactions	Size
GWDBMW	XCASINGMAT	*FILE	2014-08-14	14	650	137800
GWDBMW	XCNST_MTHD	*FILE	2014-08-14	14	1349	267102
GWDBMW	XCOMPLETN	*FILE	2014-08-14	14	1120	232960
GWDBMW	XCONDITNS	*FILE	2014-08-14	14	3353	673953
GWDBMW	XSBITEMS	*FILE	2014-08-14	14	3007	1506507
GWDBMW	XSTORETCOD	*FILE	2014-08-14	14	3487	938003
GWDBMW	XTYPELIFT	*FILE	2014-08-14	14	983	212328
GWDBMW	XTYPEPOWER	*FILE	2014-08-14	14	575	124200
GWDBMW	XUSGSWL	*FILE	2014-08-14	14	3465	852390
GWDBMW	XWATERLEV	*FILE	2014-08-14	14	3434	889406
GWDBMW	XWATERQUA	*FILE	2014-08-14	14	3412	1671880
GWDBMW	XWATERUSE	*FILE	2014-08-14	14	1938	414732
GWDBMW	XWDREMARKS	*FILE	2014-08-14	14	3517	924971

DMANZJBSFO

The IFS report lists the total numbers and sizes of journal entries for journaled IFS objects by object, date, and hour.

DMANZJPGM

The program report lists the programs that are changing the data, the transaction date, and accumulated transactions by hour, and the total size of the transactions.

08/14/14 14:39:35

iCluster Journal Analysis Report Programs

Program	Date	Hour	# of transactions	Size of transactions
CSSRVPGM	2014-07-24	17	1	159
OMSAVOBJ	2014-07-24	17	2	638
OMUTIL	2014-07-24	17	248	94488
OMUTIL	2014-08-01	17	178	62408
BKUPWEEKLY	2014-08-03	01	281	89521
OMSAVOBJ	2014-08-05	13	1	319
OMUTIL	2014-08-05	13	178	62408
DM_AGENT	2014-08-06	09	3	647
OMUTIL	2014-08-06	09	4	1084
BKUPWEEKLY	2014-08-10	02	1	201
BKUPWEEKLY	2014-08-10	03	280	89320
QCMD	2014-08-12	15	10830	2740292
QDBSTSrv	2014-08-12	15	2	666
CSSRVPGM	2014-08-14	10	1	159
QCMD	2014-08-14	11	137946	34828420
QCMD	2014-08-14	12	219650	55443316
CSSRVPGM	2014-08-14	13	1	159
QCMD	2014-08-14	13	254645	64421525
QCMD	2014-08-14	14	260692	66554801

DMANZJJOB

The job report lists the total numbers and sizes of journal entries by job, date, and hour.

08/14/14 14:39:35

**iCluster Journal Analysis Report
Jobs**

Number	User	Name	Date	Hour	# of transactions	Size of transactions
443990	XICLUSTER	@GWDBMARK	2014-07-24	17	125	47563
444033	XICLUSTER	@GWDBMARK	2014-07-24	17	125	47563
494652	MARKW_X	GWDBRDMUPD	2014-08-12	15	10830	2740292
484383	QSYS	QDBFSTCCOL	2014-08-12	15	2	666
498283	BARRIEB_X	DMANZJRN	2014-08-14	10	1	159
498469	MARKW_X	GWDBRDMUPD	2014-08-14	11	137946	34828420
498469	MARKW_X	GWDBRDMUPD	2014-08-14	12	41804	10530302
498550	MARKW_X	GWDBRDMUPD	2014-08-14	12	177846	44913014
498841	MARKW_X	DMANZJRN	2014-08-14	13	1	159
498832	MARKW_X	GWDBRDMUPD	2014-08-14	13	211958	53624879
498864	MARKW_X	GWDBRDMUPD	2014-08-14	13	42687	10796646
498832	MARKW_X	GWDBRDMUPD	2014-08-14	14	61473	15657034
498864	MARKW_X	GWDBRDMUPD	2014-08-14	14	199219	50897767

DMANZJNENT

The native object journal entry report lists the total numbers and sizes of journal entries by native object, entry code, and entry type.

08/14/14 10:14:24

**iCluster Journal Analysis Report
Journal Entries for Native Objects**

Object library	Object name	Object Type	Journal Code	Entry Type	# of Transactions	Size of Transactions
AAAJRNLIB	WTR0000127	*JRNRCV	J	RD	1	159
AAAJRNLIB	WTR0000128	*JRNRCV	J	JR	1	159
AAAJRNLIB	WTR0000128	*JRNRCV	J	RS	4	804
AAAJRNLIB	WTR0000128	*JRNRCV	J	RD	1	159
AAAJRNLIB	WTR0000129	*JRNRCV	J	JR	1	159
AAAJRNLIB	WTR0000129	*JRNRCV	J	RS	2	402
GWDB	AGENCY	*FILE	R	UB	785	175863
GWDB	AGENCY	*FILE	R	UP	785	175863
GWDB	AGENCY	*FILE	R	PX	358	88426
GWDB	AGENCY	*FILE	R	DL	359	88673
GWDB	AGENCY	*FILE	F	CB	14	4662
GWDB	AGENCY	*FILE	R	PT	3	741
GWDB	AGENCY	*FILE	F	MS	6	1914

DMANZJBENT

The IFS object journal entry report lists the total numbers and sizes of journal entries by IFS object, entry code, and entry type.

DMANZJJRN

The journal report lists the total numbers and sizes of journal entries by journal, date, and hour.

08/14/14 10:14:24

**iCluster Journal Analysis Report
Journals**

Journal library	Journal name	Date	Hour	# of transactions	Size of transactions
AAAJRNLIB	WTRJRN	2014-06-11	14	1	159
AAAJRNLIB	WTRJRN	2014-06-14	17	1	159
AAAJRNLIB	WTRJRN	2014-06-17	09	21996	5524288
AAAJRNLIB	WTRJRN	2014-06-22	03	1	201
AAAJRNLIB	WTRJRN	2014-06-22	04	278	88682
AAAJRNLIB	WTRJRN	2014-06-24	18	1	333
AAAJRNLIB	WTRJRN	2014-06-24	20	1	232
AAAJRNLIB	WTRJRN	2014-06-24	21	3	696
AAAJRNLIB	WTRJRN	2014-06-24	22	242	56144
AAAJRNLIB	WTRJRN	2014-06-24	23	242	56144
AAAJRNLIB	WTRJRN	2014-06-26	15	177	61967
AAAJRNLIB	WTRJRN	2014-06-29	07	279	88883
AAAJRNLIB	WTRJRN	2014-06-30	13	121	28072
AAAJRNLIB	WTRJRN	2014-07-07	10	37400	9436692

Related information

[DMANZJRN \(Analyze Journals command\)](#)

Cross-Library Logical File Report (DMRPTXLF command)

Use the DMRPTXLF (Report Cross-library LFs) command to generate a report that lists the cross-library logical files and physical files.

The cross-library logical files and physical files report is useful during initial configuration to ensure that physical files and their associated logical files are in the same replication group, regardless of the library in which they reside.

When logical files and their associated physical files reside in different replication groups, the **Delete dependent non-group logical files** system value must be *YES for the DMSETSVAL command. The *YES value specifies to delete the dependent logical files on the backup node when the associated physical files are deleted.

Cross-library logical files have based-on physical files in libraries that are different from the libraries of the logical files. Cross-library physical files have logical files in libraries that are different from the libraries of the physical files. The results of the report are sent to the DMRPTXLF spooled file, and optionally displayed on the console. Depending on what libraries you specify, the report search can be extensive. The recommended way to produce the report is to submit a batch job that runs the DMRPTXLF command.

The physical files with cross-library logical files are listed first, and show the based-on physical file library, the based-on physical file name, the number of cross-library logical files, the logical file library, and the logical file name. The logical files with cross-library based-on physical files are listed next, and show the logical file library, the logical file name, the logical file member, the based-on physical file library, the based-on physical file name, and the based-on physical file member.

Depending on what library or libraries are specified, the report search can be extensive. The recommended way to produce the report is to submit a batch job that runs the DMRPTXLF command.

Example: Cross Library Logical File Report sample

The following command produces this sample report for all user libraries:

```
DMRPTXLF LIB(*ALLUSR)
```

The following image shows the report sample of the physical files with cross-library logical files.

08/12/14 09:38:45 icluster Cross-library Logical Files Report

page 1

System ICDEV54A
Libraries *ALLUSR

Physical Files with Cross-library Logical Files

Based on PF Library	Based on PF Name	# of Cross-lib LFs	LF Library	LF Name
ACMSCTL	APPDEF	6	ACMIVCTL	USRAUT0001
ACMSCTL	APPDEF	6	ACMIVCTL	USRAUT0007
ACMSCTL	APPDEF	6	ACMIVCTL	USRAUT0003
ACMSCTL	APPDEF	6	ACMIVCTL	USRPER0001
ACMSCTL	APPDEF	6	ACMIVCTL	UCPERMAPP4
ACMSCTL	APPDEF	6	ACMIVCTL	UCPERMAPP2
ACMSCTL	APPLOG	5	ACMIVCTL	ATVREL
ACMSCTL	APPLOG	5	ACMIVCTL	ATVTSK
ACMSCTL	APPLOG	5	ACMIVCTL	ATVSUMMATV

Related information

[DMRPTXLF \(Report Cross-library LFs command\)](#)

Object Counts Report (DMOBJCNT command)

Use the DMOBJCNT (Report Object Counts) command to generate a report that lists the object counts and sizes for the specified library or libraries. Object counts are useful in the initial configuration to identify the objects to replicate.

The object counts report is produced in the DMOBJCNT spooled file, and is optionally displayed.

Depending on what library or libraries are specified, the report search can be extensive. The recommended way to produce the report is to submit a batch job that runs the DMOBJCNT command.

Example: Object Counts Report sample

The following command produces this sample report for all user libraries:

```
DMOBJCNT *ALLUSR
```

DMOBJCNT										Page/Line	2/8			
										Columns	1 - 130			
Find	*1.....2.....3.....4.....5.....6.....7.....8.....9.....0.....1.....2.....3.....
Library	Physical File Count	Physical File Size-Kbytes	Other Object Count	Other Object Size-Kbytes	Total Object Count	Total Object Size-kbytes	Inaccess. Object Count	Total Object Count	Total Object Size-kbytes	Member Count	Library Descripti			
AKTRIGGER	0	0	0	0	0	0	0	0	0	0	0			
ALBERTLIB	0	0	2	12012	2	12012	0	0	0	0	0			
APPLIB	2	96	1	76	3	172	0	0	0	2				
BBB	1	40	0	0	1	40	0	0	0	1				
BUSINESS	1	40	0	0	1	40	0	0	0	1	Library c			
CARLIB	6	1276	14	1144	20	2420	0	0	0	12	SEAGULL C			
CARLIBSRC	5	784	0	0	5	784	0	0	0	21	SEAGULL C			
CBSDATA	1	228	0	0	1	228	0	0	0	1				
CFRANZ_X	3	112	4	1076	7	1188	0	0	0	3				
CHGPFWA	1	64	16	2692	17	2756	0	0	0	2				
CHRISF	0	0	0	0	0	0	0	0	0	0	0			
DANSAVE	0	0	2	356628	2	356628	0	0	0	0	0			
DANTAR	6	336	2	88	8	424	0	0	0	9				
DANTARD	14	1248	17	868	31	2116	0	0	0	51				
DEMO	2	432	0	0	2	432	0	0	0	2				

Related information

[DMOBJCNT \(Report Object Counts command\)](#)

iCluster Suspended Objects Report (DMSUSOBRPT command)

Use the DMSUSOBRPT (Suspended Object Report) command to generate a report which lists the objects that are currently in suspended state in your iCluster replication configuration.

The suspended objects list will also be available in a database file whose name and library can be specified in the RESULTFILE and RESULTLIB parameters.

This command can be issued on any node in the cluster.

Pathnames longer than 35 characters cannot be listed directly in the report. For these path objects, the report will give a truncated version of the pathname, along with a numerical identifier for the full pathname. The full pathnames of truncated pathnames are listed at the end of the report in a mapping table.

The columns listed in the report for native (library) objects are:

- Node - the primary node from which the object is replicated
- IASP dev - the name of the IASP device where the object resides on the primary node
- Library - the library of the object on the primary node
- Object - the object's name
- ObjType - the object's type
- RC - the suspension reason code. The meanings of the suspension reason codes are provided in the *iCluster User Guide*.
- Group - the name of the group that is replicating the object
- JrnLib - the library of the journal that contains the journal entry related to the object's suspension
- Journal - the journal that contains the journal entry related to the object's suspension
- JrnRcvrLib - the library of the journal receiver that contains the journal entry related to the object's suspension
- JrnRcvr - the journal receiver that contains the journal entry related to the object's suspension
- Journal position - the sequence number for the journal entry related to the object's suspension

The columns listed in the report for path and DLO objects are:

- Node - the primary node from which the object is replicated
- IASP dev - the name of the IASP device where the object resides on the primary node
- Pathname - the object's full pathname, or its partial pathname with an ID giving the ID of the full pathname in the pathname mapping table at the end of the report
- ObjType - the object's type
- RC - the suspension reason code. The meanings of the suspension reason codes are provided in the *iCluster User Guide*.
- Group - the name of the group that is replicating the object
- JrnLib - the library of the journal that contains the journal entry related to the object's suspension
- Journal - the journal that contains the journal entry related to the object's suspension
- JrnrcvrLib - the library of the journal receiver that contains the journal entry related to the object's suspension
- JrnRcvr - the journal receiver that contains the journal entry related to the object's suspension
- Journal position - the sequence number for the journal entry related to the object's suspension

Example: iCluster Suspended Objects Report sample

The following image shows an example of the suspended objects report showing both suspended native objects and suspended path objects.

icluster suspended objects Report											Page 1	16/06/21 12:47:02
icluster Suspended Native Objects Report												
Node	IASP dev	Library	Object	objtype	RC	Group	JrnLib	Journal	JrnRcvrLib	JrnRcvr	Journal position	
ICTST61A	*SYSBAS	IVCOMP	D_CONTROL	*DTAARA	JRN	IVGRP	ICLUSTER	HADJRN	QSYS	QAUDJR9646	494989	
ICTST61A	*SYSBAS	IVCOMP	DMX	*SRVPGM	SBU	IVGRP	QSYS	QAUDJRN	QSYS	QAUDJR9697	9317	
ICTST61A	*SYSBAS	IVCOMP	TSTQ	*DTAQ	JRN	IVGRP	ICLUSTER	HADJRN	QSYS	QAUDJR9646	494989	

icluster Suspended Path Objects Report												
Node	IASP dev	Pathname (pathname ID)			objType	RC	Group	JrnLib	Journal	JrnRcvrLib	JrnRcvr	JrnPos
ICTST61A	*SYSBAS	/home/ivcomp/mirror/logfile			*STMF	SBU	IVREFBSF	QSYS	QAUDJRN	QSYS	QAUDJR9698	93188

Related information

[DMSUSOBRPT \(Suspended Object Report command\)](#)

iCluster commands

Refer to the command descriptions, which include iCluster concepts, general configuration information, and command examples.

A detailed description and the following information is provided for each command:

Input Parameters

Describes each parameter in the command and identifies the valid values.

Examples

Provides one or more examples of the command use.

Restrictions

Identifies limitations that determine where and when you can invoke the command.

Minimum Authority Level

Where applicable, identifies the minimum iCluster authority level (administrator, operator, or user) that is required to invoke the command.

Menu Access

Identifies the options to invoke the command from iCluster menus.

Some parameters accept wildcards to select or filter on a subset that matches a combination of characters (mask). To specify a generic name, use a character string that contains one or more characters followed by an asterisk (*). To include all results with names that start with the same characters as the generic name, up to the asterisk, specify a generic name. For example, specify AB* to generate results for all names that start with AB.

Node commands

Use the node commands to add, change, remove, display, work with nodes, change node IP address, and initialize node for recovery.

The following are the node commands with their usage:

- **DMADDNODE (Add Node command)**: Use this command to add a node to the cluster and automatically activates that node.
- **DMCHGNODE (Change Node command)**: Use this command to change one or more attributes of a node in the cluster.
- **DMRMVNODENODE (Remove Node command)**: Use this command to remove a node that was previously defined in the cluster. Cluster operations at the node are stopped.
- **DMRNMNODENODE (Rename Node command)**: Use this command on any node in the cluster, except the node to rename.
- **DMDSPNODENODE (Display Node command)**: Use this command to display the current settings for the node.
- **DMWRKNODENODE (Work with Nodes command)**: Use this command to list the nodes in the cluster. Filter the list with the **BY** and **GROUP** parameters.
- **DMCHGNODIP (Change Node IP Address command)**: Use this command to change the primary IP address of a node in the cluster when the network address of a node changes.
- **DMINZNODENODE (Initialize Node for Recovery command)**: Use this command to recover an iCluster node if a machine has a hard failure or unexpected interruption in normal operations, and group operations cannot be restarted.

Related information

- [DMADDNODE \(Add Node command\)](#)
- [DMCHGNODE \(Change Node command\)](#)
- [DMRMVNODER \(Remove Node command\)](#)
- [DMRNMMNODE \(Rename Node command\)](#)
- [DMDSPNODE \(Display Node command\)](#)
- [DMWRKNODE \(Work with Nodes command\)](#)
- [DMCHGNODIP \(Change Node IP Address command\)](#)
- [DMINZNODE \(Initialize Node for Recovery command\)](#)

DMADDNODE (Add Node command)

Adds a node to the cluster and automatically activates that node.

```
DMADDNODE NODE() IPADDR() IPADDR2() PORT() PRODLIB() DESC() REPLJOBD()
USRPRFSTS() CFGSRCHLD() STGSTORESZ() STGSTORLIB() SWITCHRES() CHKPRIMLNK()
LNKCHKFRQ() LNKCHKRTO() LNKCHKTRY() MSGQUEUE() MSGUSREXIT() NUMMSGACT()
MSGACTWAIT() JOBCCSID() COMMRCVRY() MINPORT() MAXPORT()
```

If a node has not been added to the cluster, then issue this command on the first server to define in the cluster.

Input Parameters

NODE

The name that identifies the node in the cluster.

IPADDR

The primary IP address of the node that is being added to the cluster. The Internet protocols IPv4 and IPv6 are supported. All node IP addresses in the same cluster must use the same protocol.

You can specify the IP address of the node in dotted quad notation (for example, 125.4.3.55) or in domain name form (for example, sys1a.abccorp.com). The domain name format resolves to the dotted quad format address that is used for communications.

IPADDR2

The secondary IP address of the node that is being added to the cluster. Specify the IP address of the node in dotted quad notation (for example, 125.4.3.56) or in domain name form (for example, sys1b.abccorp.com). The Internet protocols IPv4 and IPv6 are supported. All node IP addresses in the same cluster must use the same protocol. The secondary IP address is used only by Cluster Resource Services to determine if a node in the cluster is not operational or the communication link that connects the node failed. The secondary IP address is not used by iCluster for replication operations or by the SwitchOver System.

PORT

The TCP/IP port number on the node that is reserved for iCluster communications.

This port number was specified when the dmcluster service was defined to the TCP/IP service table.

The default value is 4545.

PRODLIB

The library on the node where iCluster is installed.

The default value is ICLUSTER.

DESC

A short description that identifies this node. You can use up to 50 characters.

REPLJOB

The name of the job description to associate with jobs that handle replication activity on the node.

The possible values are:

- <name>—Specifies the job description name.
- *CLUSTER—Uses the system value that is assigned in the DMSETSVAL—Set Cluster System Values command. This is the default value.

The library where the job description resides must be identified if you do not specify *CLUSTER. Prefix the job description with the name of the library where the job description is located. For example:

LIB1/RJD

USRPRFSTS

The status to assign to user profiles that are replicated to the node that you are adding to the cluster.

The possible values are:

- *CLUSTER—Uses the system value that is assigned in the DMSETSVAL—Set Cluster System Values command. This is the default value.
- *PRIMARY—Sets the user profile status to the same status that is currently assigned to the corresponding user profile on the primary node.
- *DISABLED—Sets all replicated user profiles to a status of *DISABLED on the backup node.
- *NOCHG—Preserves the current status of each replicated user profile on this node. Newly created user profiles are set to a status of *DISABLED on this node.

CFGSRCHLD

Indicates whether to automatically create configuration objects immediately after they are received on the node or hold the commands for creating them in specific source physical files so that they can be created later.

The possible values are:

- *CLUSTER—Uses the system value that is assigned in the DMSETSVAL—Set Cluster System Values command. This is the default value.
- *YES—Holds commands to create configuration objects in specific physical files so that they can be created later with the CRTCFCGOBJ (Create Configuration Objects) command.
- *NO—Creates configuration objects automatically as soon as they are received on the node you are adding to the cluster.

If a configuration object that is being replicated exists on the node that is being added to the cluster, set this value to *YES to prevent iCluster from trying to create the object when it is in use.

STGSTORESZ

Indicates the size (in megabytes) to allocate for the staging store. The size of the store changes dynamically, but the size cannot exceed the size that is specified through this parameter. The size of the staging store can range from a minimum of 512 MB to a maximum of 1,048,576 MB.

All objects and data that is replicated between nodes travels through the staging stores on the backup nodes. Assuming the backup node update process is active, the journal entries are ready to be applied. If the update process is not active, the staging store holds all journal entries until the apply process is restarted.

The default value is 512.

STGSTORLIB

The library where the staging store is located.

If you specify a library that does not exist, the library is automatically created. The default value is DMSTORE.

SWITCHRES

Indicates whether to use switchable resources on the current node.

The possible values are:

- *YES—Enables the node to use switchable resources. IBM i option 41 HA Switchable Resources must be installed on the node with a valid license key.
- *NO—Does not enable the node for switchable resources. By default, this value is used.

CHKPRIMLNK

Specifies to test the primary IP address for communication failures between the primary and backup nodes.

The possible values are:

- *YES—Tests the primary IP address for communication failures. By default, this value is used.
- *NO—Does not test the primary IP address for communication failures.

This parameter has no effect in clusters that use Cluster Resource Services.

CHKALTLNK

This parameter is no longer used by iCluster.

LNKCHKFRQ

How often, in seconds, to poll the primary and alternate links to this node for communication failures. The frequency can range from 1 - 300 seconds.

The default value is 60.

This parameter has no effect in clusters that use Cluster Resource Services.

LNKCHKRTO

The number of seconds (1 - 300) to wait for a response when polling links to this node for failures.

The default value is 30.

This parameter has no effect in clusters that use Cluster Resource Services.

LNKCHKTRY

The number of consecutive link check failures to allow before the node is considered failed. For example, if this parameter is 3, then after the fourth consecutive failure the status of the node changes to *FAILED.

The default value is 5.

This parameter has no effect in clusters that use Cluster Resource Services.

MSGQUEUE

The name and library of the message queue to receive messages after the number of consecutive link check failures, which are specified with the **LNKCHKTRY** parameter, is exceeded.

The possible values are:

- <name>—Specifies the message queue name.
- *NONE—Specifies that no message queues are used. By default, this value is used.

This parameter has no effect in clusters that use Cluster Resource Services.

MSGUSREXIT

The name and library of the user exit program to run after the number of consecutive link check failures, which are specified with the **LNKCHKTRY** parameter, is exceeded.

The possible values are:

- <name>—Calls this user exit program name.
- *NONE—Does not call a user exit program. By default, this value is used.

This user exit is run once per message that is sent to the message queue.

This parameter has no effect in clusters that use Cluster Resource Services.

NUMMSGACT

The maximum number of messages to send through the message queue, which is specified with the **MSGQUEUE** parameter, after a node failure is detected. The system waits for the time that is specified in the **MSGACTWAIT** parameter between sending each message.

The default value is 0.

This parameter has no effect in clusters that use Cluster Resource Services.

- ! **Note:** Both **MSGACTWAIT** and **NUMMSGACT** must be set to non-zero values for messages to be sent after a node failure.

MSGACTWAIT

The number of minutes to wait before sending messages to the message queue, which is specified with the **MSGQUEUE** parameter, after a node failure is detected.

The default value is 0.

This parameter has no effect in clusters that use Cluster Resource Services.

- ! **Note:** Both **MSGACTWAIT** and **NUMMSGACT** must be set to non-zero values for messages to be sent after a node failure.

JOBCCSID

Specifies the coded character set identifier (CCSID) to use for replication jobs. The replication job CCSID controls character data conversion.

The possible values are:

- *CLUSTER—Uses the system value that is assigned in the **DMSETSVAL**—Set Cluster System Values command. This is the default value.
- <value>—Specifies a valid CCSID value between 1 - 65534.
- 65535—Specifies no character data conversion and uses the country or region identifier (QCNTRYID) operating system value.

Replication errors might occur if the coded character set identifier (QCCSID) operating system value is different from the value that is associated with the country or region identifier (QCNTRYID) operating system value. Byte stream files with path names that contain characters other than English might be incorrectly processed. To avoid replication errors, set the **JOBCCSID** parameter to the same value as the coded character set identifier (QCCSID) operating system value.

COMMRCVRY

Specifies whether to use the iCluster communications auto-recovery feature in the case of a brief communication failure between nodes.

A brief communications failure is one that lasts less than the total timeout defined by the LNKCHKFRQ, LNKCHKRTO, and LNKCHKTRY parameters for the node that detects the failure.

The possible values are:

- *CLUSTER—Uses the system value that is assigned in the DMSETSVAL—Set Cluster System Values command. This is the default value.
- *YES—Specifies that the communications auto-recovery feature will be used in case of a brief communications failure.
- *NO—Specifies that the communications auto-recovery feature will not be used in case of a brief communications failure.

This parameter has no effect if the Check Primary Link (CHKPRIMLNK) parameter is set to *NO or in clusters that use IBM i Cluster Services. In those cases, the iCluster communications auto-recovery feature is not used.

MINPORT

Minimum port number of the range of TCP/IP ports that have been reserved for iCluster communications, unless ports are allowed to be dynamically assigned.

The possible values are:

- *DYNAMIC—iCluster will use dynamically assigned ports for its communications. By default, this value is used.
- port-number—Specifies an integer port number whose value can range from 1027 to 65505.

MAXPORT

Maximum port number of the range of TCP/IP ports that have been reserved for iCluster communications. iCluster needs at least 31 ports for its operation. However, to accommodate a greater number of nodes and replication groups you should reserve 10 ports for each node in the cluster and 5 ports for each group having this node in its recovery domain.

This parameter is ignored if the value of the MINPORT parameter is set to *DYNAMIC.

The possible values are:

- port-number—Specifies an integer port number whose value can range from 1057 to 65535. The MAXPORT value should be greater than the MINPORT value by at least 30.

Example 1

```
DMADDNODE NODE(NODE1) IPADDR('155.4.5.21') PORT(4141)
DESC('New York') REPLJOBD(LIB1/RJD) USRPRFSTS(*DISABLED)
CFGSRCHLD(*YES) STGSTORESZ(1024) SWITCHRES(*NO)
MSGQUEUE(QUEUES/NYQ) MSGACTWAIT(10) NUMMSGACT(6) JOBCCSID(*CLUSTER)
```

Adds the node named **NODE1** to a cluster.

NODE1 has a primary IP address (expressed in dotted quad notation) of 155.4.5.21 and will use port number 4141 for iCluster replication within the cluster.

The description indicates that the node is located in New York.

The job description associated with replication jobs on **NODE1** is **RJD** in library **LIB1**.

User profiles that are replicated to **NODE1** will all be set to a status of *DISABLED.

Commands to create configuration objects are held in specific source physical files so that they can be created later.

The maximum size of the staging store on **NODE1** is 1,024 MB.

Switchable resources will not be enabled on the node.

This node will send messages regarding primary node failures to the **NYQ** queue in the **QUEUES** library.

This node will send messages to **QUEUES/NYQ** every ten minutes if it detects a failure.

This node will send up to 6 messages to **QUEUES/NYQ**.

Uses the CCSID for replication jobs as defined by the **JOBCCSID** parameter in the **DMSETSVAL** (Set Cluster System Values) command.

This node will use the iCluster communications auto-recovery feature in the event of a brief communication failure.

Example 2

```
DMADDNODE NODE(NODE2) IPADDR('sys1a.abc123corp.com') PORT(3333)
DESC('Los Angeles') REPLJOB(LIB1/RJD) USRPRFSTS(*PRIMARY)
CFGSRCHLD(*NO) STGSTORESZ(2048) SWITCHRES(*NO) COMMRCVRY(*NO)
```

Adds the node named **NODE2** to the cluster.

NODE2 has a primary IP address (expressed in domain name form) of sys1a.abc123corp.com and will use port number 3333 for iCluster replication within the cluster.

The description indicates that the node is located in Los Angeles.

The job description associated with replication jobs on **NODE2** is **RJD** in library **LIB1**.

User profiles that are replicated to **NODE2** are set to the same status that is currently assigned to the corresponding user profile on the primary node.

Configuration objects that are replicated to **NODE2** are automatically created as soon as they are received.

The maximum size of the staging store on **NODE2** is 2048 MB.

Switchable resources will not be enabled on the node.

This node will not use the iCluster communications auto-recovery feature in the event of a brief communication failure.

Restrictions

If at least one node has been defined in the cluster, this command must be invoked from an active node. You can define a maximum of 128 nodes in the cluster.

Minimum Authority Level

*ADMIN

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **1**

Work With Nodes panel - Option **F6**

DMCHGNODE (Change Node command)

Changes one or more attributes of a node in the cluster.

```
DMCHGNODE NODE() DESC() REPLJOBD() USRPRFSTS()  
CFGSRCHLD() STGSTORESZ() SWITCHRES() CHKPRIMLNK()  
LNKCHKFRQ() LNKCHKRTO() LNKCHKTRY() MSGQUEUE() MSGUSREXIT()  
NUMMSGACT() MSGACTWAIT() JOBCCSID() CHGSTATUS() COMMRCVRY() MINPORT() MAXPORT()
```

The node that you change must be active in the cluster, with the following exception: If you are using the Cluster Resource Services failover mechanism, the node must be *INACTIVE when changing its IP addresses for iCluster.

If replication is active and you are using the Cluster Resource Services failover mechanism, the node IP addresses and job description will not be changed. You can change all other node parameters when replication is active on the node. However, some parameters used by replication jobs will only come into effect when new replication jobs are started. Existing replication jobs continue to use the previous parameters, except for the staging store size. The staging store size comes into use by all backup replication jobs as soon as it is changed.

Input Parameters

NODE

The node in the cluster that will have one or more attributes changed by this command.

The node must be in the cluster. If you specify a node that does not exist in the cluster, this command fails. Therefore, this parameter is used to reference an existing node in the cluster so that other attributes of the node can be changed.

MINPORT

Minimum port number of the range of TCP/IP ports that have been reserved for iCluster communications, unless ports are allowed to be dynamically assigned.

The possible values are:

- *SAME—Keeps the current setting for this parameter.
- *DYNAMIC—iCluster will use dynamically assigned ports for its communications.
- port-number—Specifies an integer port number whose value can range from 1027 to 65505.

MAXPORT

Maximum port number of the range of TCP/IP ports that have been reserved for iCluster communications. iCluster needs at least 31 ports for its operation. However, to accommodate a greater number of nodes and replication groups you should reserve 10 ports for each node in the cluster and 5 ports for each group having this node in its recovery domain.

This parameter is ignored if the value of the MINPORT parameter is set to *DYNAMIC.

The possible values are:

- *SAME—Keeps the current setting for this parameter.
- port-number—Specifies an integer port number whose value can range from 1057 to 65535. The MAXPORT value should be greater than the MINPORT value by at least 30.

DESC

A short description that identifies this node.

The possible values are:

- <description>—Specifies the description text. You can use up to 50 characters.
- *SAME—Keeps the current setting for this parameter.

REPLJOB

The name of the job description to associate with jobs that handle replication activity on the node.

The possible values are:

- <name>—Specifies the job description name
- *SAME—Keeps the current setting for this parameter.
- *CLUSTER—Uses the system value that is assigned in the DMSETVAL (Set Cluster System Values) command.

The library where the job description resides must be identified if you do not specify *CLUSTER or *SAME. Prefix the job description with the name of the library where the job description is located. For example:

LIB1/RJD

USRPRFSTS

The status to assign to user profiles that are replicated to the node you are changing in the cluster.

The possible values are:

- *SAME—Keeps the current setting for this parameter.
- *PRIMARY—Sets the user profile status to the same status that is currently assigned to the corresponding user profile on the primary node.
- *DISABLED—Sets all replicated user profiles to a status of *DISABLED on the backup node.
- *NOCHG—Preserves the current status of each replicated user profile on this node. Newly created user profiles are set to a status of *DISABLED on this node.
- *CLUSTER—Uses the system value that is assigned in the DMSETVAL (Set Cluster System Values) command.

CFGSRCHLD

Indicates whether to automatically create configuration objects immediately after they are received on the node or hold the commands for creating them in specific source physical files so that they can be created later.

The possible values are:

- *SAME—Keeps the current setting for this parameter.
- *YES—Holds command to create configuration objects in specific source physical files so that they can be created later with the CRTCFGOBJ (Create Configuration Objects) command.
- *NO—Creates configuration objects as soon as they are received on the node you are changing in the cluster.
- *CLUSTER—Uses the system value that is assigned in the DMSETVAL (Set Cluster System Values) command.

If a configuration object that is being replicated exists on the node that is being changed in the cluster, set this value to *YES to prevent iCluster from trying to create the object when it is in use.

STGSTORESZ

Indicates the size (in megabytes) to allocate for the staging store. The size of the store changes dynamically but it cannot exceed the size that is specified through this parameter.

The staging store is a nonvolatile storage mechanism for backup nodes that is managed on a node-by-node basis. The size of the staging store must be set for each backup node in the cluster.

All objects and data that are replicated between nodes travel through the staging stores on the backup nodes. Assuming the update process is active, the journal entries are ready to be applied. If the update process is not active, the staging store holds all journal entries until the apply process is restarted.

The possible values are:

- <size>—Specifies the size of the staging store in MB, from 512 - 1048576.
- *SAME—Keeps the current setting for this parameter.

! **Note:** Unlike the DMADDNODE (Add Node) command, you cannot modify the staging store library through this command. To change the existing staging store library for the node, the node must be removed and then added again to the cluster.

SWITCHRES

Indicates whether to use switchable resources on the node.

The possible values are:

- *SAME—Keeps the current setting for this parameter.
- *YES—Enables the node to use switchable resources. IBM i option 41 HA Switchable Resources must be installed on the node with a valid license key.
- *NO—Specifies that the node is not enabled to use switchable resources.

CHKPRIMLNK

This parameter specifies whether to test the primary IP address for communication failures between the primary and backup nodes.

The possible values are:

- *SAME—Keeps the current setting for this parameter.
- *YES—Tests the primary IP address for communication failures.
- *NO—Does not test the primary IP address for communication failures.

This parameter has no effect in clusters that use Cluster Resource Services.

CHKALTLNK

This parameter is no longer used by iCluster.

LNKCHKFRQ

Set this parameter to how often, in seconds, to poll the primary link for communication failures.

The possible values are:

- <time>—Specifies a polling time in seconds. This value can range from 1 - 300.
- *SAME—Keeps the current setting for this parameter.

This parameter has no effect in clusters that use Cluster Resource Services.

LNKCHKRTO

The number of seconds to wait for a response when polling links to this node for failures.

The possible values are:

- <time>—Specifies a wait time in seconds. This value can be from 1 - 300.
- *SAME—Keeps the current setting for this parameter.

This parameter has no effect in clusters that use Cluster Resource Services.

LNKCHKTRY

The number of consecutive link check failures to allow before the node is considered failed. For example, if this parameter is set to 3, then after the fourth consecutive failure, the status of the primary node changes to *FAILED.

The possible values are:

- <number>—The number of consecutive failures.
- *SAME—Keeps the current setting for this parameter.

This parameter has no effect in clusters that use Cluster Resource Services.

MSGQUEUE

The name and library of the message queue that will receive messages after the Cluster Resource Services number of consecutive link check failures, which are specified with the **LNKCHKTRY** parameter, is exceeded.

The possible values are:

- *SAME—Keeps the current setting for this parameter.
- <name>—The name and library of the user exit.
- *NONE—Does not call a user exit program.

This parameter has no effect in clusters that use Cluster Resource Services.

MSGUSREXIT

The name and library of the user exit program to run after the Cluster Resource Services number of consecutive link check failures, which are specified with the **LNKCHKTRY** parameter, is exceeded. This user exit is run once per message.

The possible values are:

- *SAME—Keeps the current setting for this parameter.
- <name>—The name and library of the user exit.
- *NONE—Does not call a user exit program.

This parameter has no effect in clusters that use Cluster Resource Services.

NUMMSGACT

The maximum number of messages to send through the message queue, which is specified with the **MSGQUEUE** parameter, after detecting a node failure. The system waits for the time that is specified in the **MSGACTWAIT** parameter between sending each message.

The possible values are:

- <number>—A number of messages to send.
- *SAME—Keeps the current setting for this parameter.

Note: Both **MSGACTWAIT** and **NUMMSGACT** must be set to non-zero values for messages to be sent after a node failure.

This parameter has no effect in clusters that use Cluster Resource Services.

MSGACTWAIT

The number of minutes to wait before sending messages to the message queue, which is specified with the

MSGQUEUE parameter, after detecting a node failure.

The possible values are:

- <time>—A number of minutes to wait.
- *SAME—Keeps the current setting for this parameter.

! **Note:** Both **MSGACTWAIT** and **NUMMSGACT** must be set to non-zero values for messages to be sent after a node failure.

This parameter has no effect in clusters that use Cluster Resource Services.

JOBCCSID

Specifies the coded character set identifier (CCSID) to use for replication jobs. The replication job CCSID controls character data conversion.

The possible values are:

- *SAME—Keeps the current setting for this parameter.
- *CLUSTER—Uses the system value that is assigned in the DMSETVAL—Set Cluster System Values command. This is the default value.
- <value>—A valid CCSID value between 1–65534.
- 65535—Character data conversion does not occur during replication.

Replication errors might occur if the coded character set identifier (QCCSID) operating system value is different from the country or region identifier (QCNTRYID) operating system value. Byte stream files with path names that contain characters other than English might be incorrectly processed. To avoid replication errors, set the **JOBCCSID** parameter to the same value as the coded character set identifier (QCCSID) operating system value.

CHGSTATUS

Indicates whether the status of the node should be changed.

The possible values are:

- *SAME—Keeps the current setting for this parameter.
- *FAILED—Identifies that the node failed. Use this value when a node failed, but its status in the cluster is *PARTITION. After the status of the node changes to *FAILED, the node can be removed from the cluster or restarted, if the node is capable of rejoining the cluster.

COMMRCVRY

Specifies whether to use the iCluster communications auto-recovery feature in the case of a brief communication failure between nodes.

A brief communications failure is one that lasts less than the total timeout defined by the LNKCHKFRQ, LNKCHKRTO, and LNKCHKTRY parameters for the node that detects the failure.

The possible values are:

- *SAME—Keeps the current setting for this parameter.
- *CLUSTER—Uses the cluster system value for COMMRCVRY assigned in the DMSETVAL command.
- *YES—Specifies that the communications auto-recovery feature will be used in case of a brief communications failure.
- *NO—Specifies that the communications auto-recovery feature will not be used in case of a brief communications failure.

This parameter has no effect if the Check Primary Link (CHKPRIMLNK) parameter is set to *NO or in clusters that

use IBM i Cluster Services. In those cases, the iCluster communications auto-recovery feature is not used.

Example 1

```
DMCHGNODE NODE(NODE1) DESC(Chicago)
REPLJOB(LIB1/RJD) USRPRFSTS(*DISABLED) CFGSRCHLD(*YES) STGSTORESZ(4096)
SWITCHRES(*NO) JOBCCSID(*CLUSTER) CHGSTATUS(*SAME)
```

Changes the attributes of **NODE1** in the cluster.

Changes the primary IP address of **NODE1** to 155.4.5.44 (expressed in dotted quad notation).

Changes the description that is associated with the node to indicate that the system is located in Chicago.

Changes the job description that is associated with replication jobs on **NODE1** to **RJD** in library **LIB1**.

User profiles that are replicated to **NODE1** are set to a status of *DISABLED.

Commands to create configuration objects are held in specific source physical files so that they can be created later.

Changes the maximum size of the staging store on **NODE1** to 4,096 MB.

Switchable resources are not be enabled on the node, and the status of the node remains the same.

Uses the CCSID for replication jobs as defined by the **JOBCCSID** parameter in the DMSETSVAL (Set Cluster System Values) command.

This node will use the iCluster communications auto-recovery feature in the event of a brief communication failure.

Example 2

```
DMCHGNODE NODE(NODE2) DESC(London)
REPLJOB(LIB1/RJD) USRPRFSTS(*PRIMARY) CFGSRCHLD(*NO)
STGSTORESZ(8192) SWITCHRES(*NO) CHGSTATUS(*SAME) COMMRCVRY(*NO)
```

Changes the attributes of **NODE2** in the cluster.

Changes the primary IP address of **NODE2** to sys2a.abc123corp.com (expressed in domain name form).

Changes the description associated with the node to indicate that the system is located in London.

Changes the job description associated with replication jobs on **NODE2** to **RJD** in library **LIB1**.

User profiles that are replicated to **NODE2** are set to the same status that is currently assigned to the corresponding user profile on the primary node.

Configuration objects that are replicated to **NODE2** are automatically created as soon as they are received.

Changes the maximum size of the staging store on **NODE2** to 8,192 MB.

Switchable resources will not be enabled on the node, and the status of the node will remain the same.

This node will not use the iCluster communications auto-recovery feature in the event of a brief communication failure.

Example 3

```
DMCHGNODE NODE(NODE3) LNKCHKTRY(3)
```

Changes the attributes of **NODE3** in the cluster.

Sets the SwitchOver System to allow three link check failures before changing the status of the node to *FAILED.

Restrictions

You must issue this command from an active node.

Minimum Authority Level

*ADMIN

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **2**

Work With Nodes panel - Option **2**

DMRMVNODER (Remove Node command)

Removes a node that was previously defined in the cluster. Cluster operations at the node are stopped.

```
DMCHGNODE NODE( )
```

You cannot make a reference to the node after it is removed from the cluster. If you remove the only node in the cluster, the entire cluster is deleted. In this case, the removal of the node is equivalent to the operations performed when the DMDLTCLSTR (Delete Cluster) command is invoked.

You can use this command to remove the backup node for replication groups. You cannot remove the primary node of a group. If you want to remove a primary node from the cluster, you can take one of two actions:

- Remove all groups with this node as their primary node.
- End cluster operations at the node before removing it. After ending cluster operations, ensure that you remove the cluster definitions from the inactive node by issuing the DMDLTCLSTR (Delete Cluster) command at that node.

Input Parameters

NODE

The name of the existing node to remove from the cluster.

Example

```
DMRMVNODER NODE(NODE1)
```

Removes the NODE1 node from the cluster.

Restrictions

You must issue this command from an active node.

If you issue this command on a node other than the specified node to be removed, the specified node must be active. If the specified node is inactive, the node is removed only from the active part of the cluster. A later attempt to add the same node to the cluster will fail unless you issue the DMDLTCLSTR (Delete Cluster) command on the specified node before you attempt to add it to the cluster again.

Minimum Authority Level

*ADMIN

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **3**

Work With Nodes screen - Option **6**

DMRNMNODE (Rename Node command)

Use this command on any node in the cluster, except the node to rename.

```
DMRNMNODE NODE( ) NEWNODE( )
```

All nodes in the cluster must be active. The node that is being renamed cannot be the node where the command is run. Groups that have the node that is being renamed in their recovery domain must be inactive, and have another node in their recovery domain. The groups must have a current backup node. The groups cannot be local loopback groups.

After the command is run, restart the groups that have the renamed node in their recovery domain. Because all replication information is lost during the rename, restart the groups as if the groups are new groups. Use the DMMRKPOS command, or do a refresh, depending on how much data is being replicated.

Input Parameters

NODE

The name of the node to rename.

The node must be active in the cluster.

NEWNODE

The new name of the node.

Example

```
DMRNMNODE NODE(MYNODE) NEWNODE(MYNODEB)
```

Renames the node named **MYNODE** to **MYNODEB**.

Removes the **MYNODE** node from the cluster and adds the **MYNODEB** to the cluster, with the same parameters.

Historical replication information is no longer available.

Restrictions

The node being renamed must be active in the cluster. Groups that have the node that is being renamed in their recovery domain must be inactive, and have another node in their recovery domain. The groups must have a current backup node. The groups that have the node that is being renamed in their recovery domain cannot be local loopback groups.

The groups that have the renamed node in their recovery domain will have their last applied journal positions reset. The groups do not have to be restarted as if they are newly-created groups. However, the replication information for these groups will be rebuilt when the groups are restarted. In the case of groups replicating a large number of objects, the group startup after renaming a node in the group's recovery domain will take longer because of this replication information rebuild.

Minimum Authority Level

*ADMIN

Menu Access

None.

DMDSPNODE (Display Node command)

Displays the current settings for the node.

```
DMDSPNODE NODE( )
```

Input Parameters

NODE

The name of the node to examine.

Example

```
DMDSPNODE NODE(NODE1)
```

Displays the current settings for the node **NODE1** when prompted with F4.

Restrictions

You can issue this command on an active node in the cluster.

Minimum Authority Level

*USER

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option 4

Work With Nodes screen - Option 5

DMWRKNODE (Work with Nodes command)

Lists the nodes in the cluster. Filter the list with the **BY** and **GROUP** parameters.

```
DMWRKNODE BY( ) GROUP( )
```

The status of each node is indicated on the displayed screen, and can be one of the following:

- *ACTIVE—Indicates that cluster operations are running on the node.
- *ACT_PEND—Indicates that cluster operations are in the process of being started on the node.
- *INACTIVE—Indicates that cluster operations are not running on the node.
- *INACT_PND—Indicates that cluster operations are in the process of being stopped on the node.
- *NEW—Indicates that the node has been defined in the cluster, but cluster operations have not been started on the node.
- *RMV_PEND—Indicates that the node is in the process of being removed from the cluster.
- *FAILED—Indicates that a system failure has occurred on the node, or other nodes in the cluster can no longer communicate with this node.
- *PARTITION—Indicates that a system failure has occurred on the node or communications with the node have been lost. If a primary node in an active group is in this state, the group will assume an *INACTIVE status and no role change will occur. If a backup node in an active group is in this state, the group will remain in an *ACTIVE state and no role change will occur.
- *IN_ERROR—Indicates that an internal iCluster error may have occurred. If this status persists, contact Technical Support.
- *UNKNOWN—Indicates that the node is not recognized in iCluster, or the cluster is not recognized on the system, or the current node is inactive in the cluster.

Input Parameters

BY

The set of nodes that are listed by this command.

The possible values are:

- *NONE—Specifies all defined nodes. By default, this value is used.
- *GROUP—Specifies nodes that are included in the recovery domain for the group specified through the **GROUP** parameter.

GROUP

The name of an existing group. Nodes included in the recovery domain for the named group are listed.

This parameter applies only when the **BY** parameter is set to *GROUP.

Example 1

```
DMWRKNODE BY(*NONE)
```

Lists all defined nodes.

Example 2

```
DMWRKNODE BY(*GROUP) GROUP(GRP1)
```

Lists all nodes in the recovery domain for group **GRP1**.

Restrictions

You must issue this command on an active node in the cluster.

Minimum Authority Level

*USER

Menu Access

Main Menu - Option 1

iCluster Commands menu - F22 (Shift + F10) - Option 5

DMCHGNODIP (Change Node IP Address command)

Changes the primary IP address of a node in the cluster when the network address of a node changes.

```
DMCHGNODIP NODE() IPADDR()
```

For example, the IP address might need to be changed when the backup node is initially configured in the same physical location as the primary node, but is later moved to a remote location on a different network.

The node whose primary IP address is changing must be inactive in the cluster, but it must be accessible using the specified IP address.

Input Parameters

NODE

The existing node in the cluster that will have the new IP address.

IPADDR

The new primary IP address of the node. The Internet protocols IPv4 and IPv6 are supported. All node IP addresses in the same cluster must use the same protocol. Specify the IP address of the node in dotted quad notation (for example, 125.4.3.55) or in domain name form (for example, 'sys1a.abccorp.com').

Example

```
DMCHGNODIP NODE(NODE1) IPADDR('155.4.5.44')
```

Changes the primary IP address of **NODE1** to 155.4.5.44.

Restrictions

You must issue this command from an active node that can communicate with the inactive node whose IP address is being changed.

Minimum Authority Level

*ADMIN

Menu Access

This command is not available from the iCluster menus.

DMINZNODE (Initialize Node for Recovery command)

Recovers an iCluster node if a machine has a hard failure or unexpected interruption in normal operations, and group operations cannot be restarted.

```
DMINZNODE CLRSTGSTOR( ) CLRUSRSPCQ( )
```

This command performs the required recovery commands and optionally clears user spaces and the staging store. You must issue this command on every node in the cluster before restarting replication.

Use this command only when iCluster operations cannot be restarted normally. The command performs the DMSETPOS (Set Journal Start Position) command for all groups and journals that use this node as their primary node, so restarting the groups and journals could take much longer than restarting replication normally.

The work for DMINZNODE (Initialize Node for Recovery) is performed by a separate job that runs under the **QDFTJOBDB** job description. Ensure that the job queue for **QDFTJOBDB** is available for this job to run before you call the command.

If there is no definition of the current node in the iCluster metadata, the SETHAREG (Restore iCluster Communications Registry) command is invoked to allow the creation of a new cluster.

Input Parameters

CLRSTGSTOR

Specifies whether to clear the staging store libraries that are used on the current node. The staging stores are recreated when operations are restarted.

The possible values are:

- *YES—Clears the staging store libraries of all groups that use the current node as their backup node. By default, this value is used.
- *NO—Does not clear any staging store libraries.

CLRUSRSPCQ

Specifies whether to delete *USRQ and *USRSPC objects that were created on this node for iCluster internal use. The user spaces and user queues are recreated when operations are restarted.

The possible values are:

- *YES—Deletes all the user queues and user spaces that were created on this node for iCluster internal use. By default, this value is used.
- *NO—Does not delete the user queues and user spaces.

Example

```
DMINZNODE CLRSTGSTOR(*YES) CLRUSRSPCQ(*NO)
```

Performs the recovery commands for the current node and clears the staging store.

Restrictions

The job queue for QDFTJOBDB must be available for this job to run.

Minimum Authority Level

*ADMIN

Menu Access

This command is not available from the iCluster menus.

Group commands

Use the group commands to add, change, remove, display, work with groups, and add or remove backup nodes.

The following are the group commands and their usage:

- **DMADDGRP (Add Group command)**: Use this command to add a group that consists of one primary node and possibly one backup node.
- **DMCHGGRP (Change Group command)**: Use this command to change one or more attributes of an existing replication group or, in certain cases, an existing refresh group.
- **DMRMVGRP (Remove Group command)**: Use this command to remove a group that was previously defined. A reference cannot be made to the group after it has been removed.
- **DMDSPGRP (Display Group command)**: Use this command to display the current settings for a replication group on an active node in the cluster.
- **DMWRKGRP (Work with Groups command)**: Use this command to list the groups in a cluster. Use the **BY, NODE**, and **STATUS** parameters to filter the list.
- **DMADDBACK (Add Backup Node to Recovery Domain command)**: Use this command to add a backup node to the recovery domain for an existing group.
- **DMRMVBACK (Remove Backup Node from Recovery Domain command)**: Use this command to remove a backup node to the recovery domain from an existing group.

Related information

- [**DMADDGRP \(Add Group command\)**](#)
- [**DMCHGGRP \(Change Group command\)**](#)
- [**DMRMVGRP \(Remove Group command\)**](#)
- [**DMADDBACK \(Add Backup Node to Recovery Domain command\)**](#)
- [**DMRMVBACK \(Remove Backup Node from Recovery Domain command\)**](#)
- [**DMDSPGRP \(Display Group command\)**](#)

[**DMWRKGRP \(Work with Groups command\)**](#)

DMADDGRP (Add Group command)

Adds a group that consists of one primary node and possibly one backup node.

```
DMADDGRP GROUP() GRPTYPE() DMNSRC() PRIMNODE() PRIMIASP() BACKUPS() BACKIASP()
TGTLIB() STGSTORLIB() MQVERSION() QMNAME() MSGQUEUE() ROLESWITCH() CHKJRN()
BSWUSREXIT() ASWUSREXIT() EXITDATA() POLLINT() SAVACT() DFTDBJRN() JRNLOC()
JRNBACKUP() CMTLVL() JRNBA() PFRFSHMTD() OPTIMZAPY() DFTBSFJRN() RCVRYEXP() JRNKEY()
LCKBACKUP() CFGSRCHLD() SWDEV() ONLINE() DESC() CONTSYNC() DELAY() SCTYPE()
AUTREPAIR() JRNREPAIR() CBUREPAIR() CRDREPAIR() ACT() MAXRLE() HADELAY() SAVRETRY()
LOCKCHKS() LOCKCHKDLY() APYDELAY() ALWOOS() ALWSUS() ALRTMON() CHKLATNCY()
AUTOSTART() RUNPTY()
```

After creating a group, use the DMSELOBJ (Select Objects to Group) or DMADDBSF (Add Path Specifier to Group) commands to specify the objects that the group replicates.

Input Parameters

GROUP

The name of the group to define.

The group name must be unique in the cluster.

GRPTYPE

Specifies the type of group to define.

The possible values are:

- *REPL—Specifies a group for continuous object replication between the primary and backup nodes.
- *RFSH—Specifies a refresh-only group.
- *SWDEV—Specifies a group for the switchable disk storage devices on the primary and backup nodes.
- *MQSERIES—Specifies a group to replicate a WebSphere MQ environment (a queue manager and its messages).
- *STDBY—Specifies a stand-by group that can be converted to a replication or refresh-only group at a later time; for example, in a disaster recovery situation.

Note: A group with type *STDBY will always have a status of *NONE. It cannot:

- be started (and consequently cannot be ended).
- have marked positions or set positions
- be seen in the Status Monitor
- be considered as a group for RTVHAPOS, DSPHAPOS, journal management, and other group operations related to replication.
- have a role switch

When the group must be used, run the DMCHGGRP command to change the group type to *REPL, do whatever pre-start processing must be done (DMSETPOS or DMMRKPOS), and then start the group.

When a group is changed from *REPL to *STDBY, the group's metadata (other than the



group itself and its object specifiers) will be cleared, the same as when a group undergoes a roleswitch.

- *HADR—Specifies a replication group intended for both high availability (HA) and disaster recovery (DR). An *HADR group has one primary node and two backup nodes: one for high availability and one for disaster recovery. When a roleswitch is done with an *HADR group, the HA backup node becomes the new primary node and the previous primary node becomes the new HA backup node. The role of the DR backup node is not changed. Replication in an *HADR group is from the primary node to both of the backup nodes.



Note: A group of type *HADR cannot be a local loopback group and it cannot have target library redirection. The group must conform to the requirements for roleswitch eligibility. This is enforced when backup nodes are defined for an *HADR group and when object or BSF specifiers are selected to an *HADR group.

DMNSRC

The name of an existing group to use to define the recovery domain of the group you are adding.

This parameter allows you to add a group with the same set of primary and backup nodes as the recovery domain for an existing group. Defining two or more groups with the same set of nodes is useful when the goal is to replicate objects between the same nodes, but to define different replication group behaviors for different sets of objects.

The possible values are:

- <name>—Specifies the group name.
- *LIST—Specifies the explicit identification of the primary and backup nodes in the group instead of selecting an existing group. By default, this value is used.

PRIMNODE

The name of a node that will be the primary node in the new group.

The node must have been added to the cluster using the DMADDNODE (Add Node) command.

This parameter applies only if the DMNSRC parameter is set to *LIST.

For *SWDEV groups, the primary and backup nodes must have the switchable disk storage devices enabled (**SWITCHRES** parameter).

PRIMIASP

The name of an independent auxiliary storage pool (IASP) on the primary node from which you want to replicate.

The possible values are:

- <name>—Specifies the IASP name.
- *SYSBAS—Uses a system ASP rather than an IASP. This is the default value.

BACKUPS

The name of a node that will be the backup node in the new group.

Only one backup node can be specified for a group.

The node must have been added to the cluster using the DMADDNODE (Add Node) command.

This parameter applies only if the DMNSRC parameter is set to *LIST.

BACKIASP

The name of an independent auxiliary storage pool (IASP) on the backup node to which you want to replicate.

The possible values are:

- <name>—Specifies the IASP name.
- *SYSBAS—Uses a system ASP rather than an IASP. This is the default value.

TGTLIB

The name of the library on the backup node that receives the replicated objects.

The possible values are:

- <name>—The library name.
- *PRIMARY—Sets the backup node library to be identical to the primary node library where the object resides. If the primary and backup environments reside on the same physical server (local loopback replication), the library that you specify cannot be the same library where a selected object in the group currently resides. By default, this value is used.

STGSTORLIB

The name of the staging store library for the group.

The staging store is a nonvolatile storage mechanism for the changes that are sent from the primary node to be applied to the backup node. The staging store holds these changes before they are applied to the backup node to allow the backup apply processes to be shut down temporarily without shutting down the journal scraper processes. Specify a staging store library for each group to improve performance and minimize overall latency.

The possible values are:

- *NODE—Sets the staging store library to the library that is set for the node. By default, this value is used.
- <name>—A unique staging store library name that is not used for any other staging store library. It is recommended to specify a library that does not already exist on your server. The library is created if it does not already exist.

MQVERSION

For groups of type *MQSERIES, specify the product version of WebSphere MQ that you are using.

The possible values are:

- V5R2M0
- V5R3M0
- V6R0M0
- V7R0M0
- V7R0M1
- V7R1M0. This is the default value.

QMNAME

For groups of type *MQSERIES, specify the name of the WebSphere MQ queue manager that must be mirrored. Multiple queue managers are not supported. To mirror multiple queue managers, use the DMADDGRP (Add Group) command to define a group for each of them.

Enter specific names. Do not use *DFT or generic names.

MSGQUEUE

The name of the message queue on the new primary node that receives messages that are generated when a failover occurs. This message queue can be monitored by third-party software or procedures outside of iCluster. This parameter applies only to replication groups and MQSeries groups.

The possible values are:

- <name>—Specifies the message queue name. The library where the message queue resides must be identified if you do not specify *NONE. Prefix the message queue with the name of the library where the queue is located. For example:

LIB1/MSGQ1

- *NONE—Specifies that message queues are not used. By default, this value is used.

ROLESWITCH

Specifies to change the role of the backup node automatically so that it becomes the primary node in the group when a failover occurs.

This parameter applies only to replication groups that have the value of the **Target library** field set to *PRIMARY, and to WebSphere MQ groups.

The functions of the primary node are moved to the backup node if this parameter is set to *YES when a failover occurs.

The possible values are:

- *YES—Changes the role of the backup node in the group automatically to the primary node when a failover occurs. In this case, user exit programs that are specified through the **BSWUSREXIT** and **ASWUSREXIT** parameters are called. Only groups that have a status of *ACTIVE are switched.
- *NO—Does not change the role of the backup node in the group automatically to the primary node when a failover occurs. This option allows you to continue with a role switch, or continue with your current configuration. By default, this value is used.

With this setting, a failure of the group's primary node is declared by the failover mechanism, although the backup node is not prepared to take over as the primary node. Replication does not start automatically after the failed primary node is again active in the cluster.

CHKJRN

When a group role switch is being performed, it specifies whether to check that the objects for the group are being correctly journaled on the new primary node, and starts journaling them if necessary. Certain types of objects must be journaled before replication starts. However, because journaling objects can be time-consuming, this option allows you to specify whether to start journaling for objects during a switchover, depending on your business requirements.

Depending on the value of the group **CHKJRN** parameter, processing that is equivalent to the DMMRKPOS (Mark Journal Positions) command might be performed when a role switch occurs.

If the value of the group's **CHKJRN** parameter is *NO, iCluster does not perform DMMRKPOS (Mark Journal Positions) processing and the replication metadata for the new primary node is generated based on the existing replication metadata on the current backup node. In case of an error in processing the backup replication metadata, or nonexistent backup replication metadata, iCluster performs DMMRKPOS (Mark Journal Positions) processing regardless of the **CHKJRN** parameter's value.

If the value of the group's **CHKJRN** parameter is *YES, iCluster always performs DMMRKPOS (Mark Journal Positions) processing during a role switch, and the backup replication metadata is not used for setting up the replication metadata for the new primary node. The DMMRKPOS (Mark Journal Positions) command establishes the set of mirrored objects based on the actual objects that match specifiers on the backup system, regardless if they were previously mirrored or not, and starts journaling for the mirrored objects that should be journaled.

The possible values are:

- *YES—Checks if all mirrored objects that should be journaled are being journaled on the new primary

node. If not, the objects are journaled. By default, this value is used.

- *NO—Does not check if all mirrored objects that should be journaled are journaled on the new primary node. You must make sure that all objects that need to be journaled are being journaled to the correct journals, and that the objects have been journaled before replication starts on the new primary node.

BSWUSREXIT

The name of the user exit program to call immediately before the role change is performed.

The user exit program is called on both nodes of the group for a switchover, but only on the new primary node for a failover.

This parameter applies only to replication groups with the value of the **Target library** field set to *PRIMARY.

The possible values are:

- <name>—Specifies the user exit program name. The **ROLESWITCH** parameter must be set to *YES to call the program when a failover occurs. The user exit program will always be called (regardless of the **ROLESWITCH** parameter setting) when a switchover is initiated.
- *NONE—Does not call a user exit program. By default, this value is used.

The library where the user exit program resides must be identified when *NONE is not specified. Prefix the name of the user exit program name with the name of the library where the program is located (for example, LIB1/USREXIT1).

ASWUSREXIT

The name of the user exit program to call immediately after the role change is performed. After a role change occurs, the user exit program is called on both nodes of the group.

This parameter applies only to replication groups with the value of the **Target library** field set to *PRIMARY.

The possible values are:

- <name>—Specifies the user exit program name. The **ROLESWITCH** parameter must be set to *YES to call the program when a failover occurs. The user exit program is always invoked (regardless of the **ROLESWITCH** parameter setting) when a switchover is initiated.
- *NONE—Does not call a user exit program. By default, this value is used.

The library where the user exit program resides must be identified when *NONE is not specified. Prefix the name of the user exit program name with the name of the library where the program is located (for example, LIB1/USREXIT1).

EXITDATA

Identifies the user-defined data to pass to the user exit programs specified through the **BSWUSREXIT** and **ASWUSREXIT** parameters.

This parameter applies only to replication groups with the value of the **Target library** field set to *PRIMARY.

If you specify two different user exit programs (one program before the role switch, and a different program after the role switch), the same user-defined data is passed to both programs. A maximum of 256 bytes of data can be passed to the user exit programs. If your data contains spaces or non-alphanumeric characters (commas, periods, and so on), you must enclose your data in single quotes.

POLLINT

The time interval that determines how often to check for content changes to user spaces, source physical files, and Document Library Objects (DLO).

This parameter applies only to replication groups.

You set the interval by specifying the number of hours (first two digits), minutes (middle two digits), and seconds (last two digits) between consecutive polls. The polling interval applies only when pollable objects (user spaces, source physical files, and (DLO) Document Library Objects) are selected to the group through the DMSELOBJ

(Select Objects to Group) command.

The possible values are:

- <time>—A time value from 000010 - 235959 expressed in HHMMSS format.
- *CLUSTER—Uses the system value that is assigned in the DMSETVAL (Set Cluster System Values) command.
- *NONE—Does not poll objects that are selected to the group.

! **Note:** If *NONE is selected as the polling interval for an object specifier but the group's polling interval is not *NONE, polling occurs for all pollable objects that are selected to the group that do not match the object specifier that has a polling interval of *NONE.

SAVACT

Indicates if an object can be updated at the same time it is being saved for refresh to the backup node. This parameter does not apply to physical files because physical objects cannot be modified while they are being saved.

This parameter applies only to replication groups.

The possible values are:

- *YES—Allows objects in use by another job to be saved by the product. Objects may reach checkpoints at different times and may not be in a consistent state in relationship to each other.
- *NO—Does not allow objects in use to be saved and updated at the same time. Objects cannot be updated while being saved.
- *SYNC—Allows objects in use by another job to be saved and updated. All of the objects reach a checkpoint together and are saved in a consistent state in relationship to each other.
- *CLUSTER—Uses the system value that is assigned in the DMSETVAL (Set Cluster System Values) command. This is the default value.

DFTDBJRN

The name of the database journal to use as the default.

This parameter applies only to replication groups.

The journal that you specify is used for files, data areas, and data queues that are to be mirrored but are not yet journaled. iCluster starts journaling automatically to this journal.

The possible values are:

- <name>—Specifies the database journal name. The library where the journal resides must be identified if you do not specify *CLUSTER. Prefix the journal with the name of the library where the journal is located. For example:

LIB1/JRN1

- *CLUSTER—Uses the system value that is assigned in the DMSETVAL (Set Cluster System Values) command. This is the default value.

! **Note:** If this group replicates data from an IASP device, then the journal must exist on the IASP device before the group can start replicating objects.

JRNLOC

The location of the database journal where scraping occurs.

The possible values are:

- *LOCAL—Scrapes the database journals on the primary node during replication. By default, this value is used.
- *REMOTE—Scrapes the remote database journals on the backup node during replication.

! **Note:** When a remote journal group starts, iCluster activates the remote journal automatically (if it is not already active) using the default value for the **DELIVERY** parameter in the **CHGRMTJRN** command.

After changing the journal location with this parameter, you must set the position of the local journal to the position of the last applied entry by using the **DMSETPOS** (Set Journal Start Position) command with the **JRNPOS(*LASTAPY)** parameter.

JRNBACKUP

Indicates whether to journal the replicated physical files, data areas, data queues and BSF files on backup nodes.

! **Note:** This parameter applies only to *REPL and *MQSERIES groups.

The possible values are:

- *CLUSTER—Uses the system value that is assigned in the **DMSETVAL**—Set Cluster System Values command. This is the default value.
- *YES—Journals replicated physical files, data areas, data queues and BSF files on backup nodes.
- *NO—Does not journal the physical files, data areas, data queues and BSF files on backup nodes.

! **Note:** For PFDTA files that are replicated with **MIRRCNTS(*NO)**, iCluster does not journal them on backup nodes even if the **JRNBACKUP** parameter is set to *YES.

CMTLVL

The level of commitment control to use when replicating *FILE objects.

This parameter applies only to replication groups.

Commitment control stages database transactions so that they are assembled before being applied, or additionally, opened in a commitment control environment to ensure that all updates are received. It also makes sure that the changes are applied in the correct sequence.

The possible values are:

- *NONE—Commitment control staging will not be performed during the update process on the backup node.
- *LEVEL1—Assembles all updates that comprise a transaction before being applied on the backup node.
- *LEVEL2—Opens all updates in a transaction in a commitment control environment to ensure that all updates are received. This option provides true commitment control. If you select *LEVEL2, but a file on the backup node cannot be opened under commitment control, *LEVEL1 commitment control is used for that file.
- *CLUSTER—Uses the system value that is assigned in the **DMSETVAL** (Set Cluster System Values) command. This is the default value.

JRNBA

Indicates if default journaling includes both before and after images.

This parameter applies only to replication groups.

Before images are required for commitment control *LEVEL 1 and *LEVEL2 in the case of a rollback of transactions under commitment control. Also, if you specified unique key update method for a file, then it must be journaled with both before and after images. If a file is replicated using relative record number, it may be journaled with after images only.

Both before and after images must be journaled if the **CMTLVL** parameter is set to *LEVEL2.

The possible values are:

- *BOTH—Journals both before and after images.
- *AFTER—Journals only the after image.
- *CLUSTER—Uses the system value that is assigned in the DMSETVAL (Set Cluster System Values) command. This is the default value.

PFRFSHMTD

Specifies the method of physical file refresh when a physical file is refreshed as part of a group refresh.

This parameter only applies to groups of type *REPL or *RFSH.

The possible values are:

- *CLUSTER—Uses the system value that is assigned in the DMSETVAL (Set Cluster System Values) command. This is the default value.
- *RBR—Record by record refresh.
- *SAVRST—Refresh by save/restore. Physical files are saved with the data and restored on the backup node. When the *SAVRST method is used to refresh files, the "Created by user" attribute of a file is retained on the backup node.



Note: The *SAVRST method of physical file refresh will not succeed if there are any other update or exclusive locks on the file being refreshed at the time of refresh, or if the file is under commitment control. A file's refresh by *SAVRST will also fail if the group's "Save active" parameter evaluates to *NO and the file is open or locked for update by another application. In these cases, you should specify the *RBR refresh method.

OPTIMZAPY

Indicates whether to optimize database apply entries.

Optimization can increase the apply performance for large files that are significantly larger than the shared memory pool, and have a large number of random updates applied to them.

The possible values are:

- *NO—Does not enable optimization for database apply updates. By default, this value is used.
- *YES—Enables optimization for database apply updates.

DFTBSFJRN

The name of the journal to use as the default for journaling BSF files that are replicated by this group.

This parameter applies only to replication groups.

The journal is used for BSF files that are to be mirrored and journaled.

The possible values are:

- <name>—Specifies the journal name. If this group replicates data from an IASP device, then the journal must exist on the IASP device before the group can start replicating objects.
- *NONE—Journaling is not required for BSF replication.
- *CLUSTER—Uses the system value that is assigned in the DMSETVAL (Set Cluster System Values) command. This is the default value.

The library where the journal resides must be identified if you do not specify *NONE or *CLUSTER. Prefix the journal with the name of the library where the journal is located. For example:

LIB2/JRN2

RCVRYEXP

Specifies the number of journal entries that are applied between writing recovery checkpoints. Specifying a lower number decreases the performance of the apply, but provides a greater safeguard in recovery situations. Specifying a higher number lessens the impact of recovery checkpoints on the apply.

This parameter applies only to replication groups.

The possible values are:

- <number>—Specifies the number of journal entries from 0-2147483647.
- *DISABLED—Does not generate recovery checkpoints. By default, this value is used.

JRNKEY

Specifies the key, or the identifier, to assign to a recovery checkpoint for the group. You must also specify this value in the RTVRCVPT (Retrieve Recovery Checkpoint) or RTVRCVPT (Retrieve Recovery Checkpoint (CL Program)) command.

This parameter applies only to replication groups.

LCKBACKUP

Indicates whether journaled files should be locked on the backup node. When locked, no other users can modify these files on the backup node.

The possible values are:

- *CLUSTER—Uses the system value that is assigned in the DMSETVAL (Set Cluster System Values) command. This is the default value.
- *YES—Locks files so that they cannot be modified on the backup node by other processes when the group is active.
- *NO—Does not lock files on the backup node when the group is active.

CFGSRCHLD

Indicates whether to automatically create configuration objects immediately after they are received on a backup node or hold the commands for creating them in specific source physical files so that they can be created later with the CRTCFOBJ (Create Configuration Objects) command.

If configuration objects that are being replicated already exist on backup nodes, this parameter should be set to *YES in order to prevent iCluster from trying to create objects when they are in use.

The possible values are:

- *BACKUP—Uses the **Hold configuration object source** setting on the backup node of the group that is specified through the DMADDNODE (Add Node) command. By default, this value is used.
- *CLUSTER—Uses the system value that is assigned in the DMSETVAL (Set Cluster System Values) command.

- *YES—Holds configuration objects in specific physical files so that they can be created later with the CRTCFCGOBJ (Create Configuration Objects) command.
- *NO—Creates configuration objects as soon as they are received on the node.

CFGSCOPTS - Configuration object sync check options

Indicates whether a configuration objects should be reported as out of sync if the CL source member for creating the object exists on the backup node, even when the object itself does not exist on the backup node.

This parameter is used only when the value of **CFGSRCHLD** is set to *YES.

The possible values are:

- *CLUSTER—Uses the cluster system value that was assigned in the DMSETSVAL (Set Cluster System Values) command. By default, this value is used.
- *RPTCFG—Configuration objects are reported as out of sync with the CFG reason code if the objects do not exist on the backup node but the CL source members for creating the objects exist on the backup node.
- *NORPTCFG—Configuration objects are not reported as out of sync if the CL source members for creating the objects exist on the backup node, even when the objects themselves do not exist on the backup node.

SWDEV

The name of a switchable disk storage device that is associated with this group.



Note: This parameter applies only when the **GRPTYPE** parameter is set to *SWDEV.

ONLINE

Indicates whether the device associated with the switchable resource group is varied on or varied off when the group is switched over from one node to another or when it is failed over to another node.

This parameter applies only to groups of type *SWDEV.

The possible values are:

- *YES—Specifies that the device associated with the group is varied on when the group is switched over from one node to another or when it is failed over to another node. By default, this value is used.
- *NO—Specifies that the device associated with the group is not varied on when the group is switched over from one node to another or when it is failed over to another node.

DESC

A short description that allows you to identify this group. You can use up to 50 characters.

CONTSYNC

Specifies that sync checks are run continuously, as long as the latencies (source receive and backup apply) of the group being sync checked are below their respective latency thresholds and the group is active. If a sync check is not started because of excess latency, a message is issued the first time this happens (after a successful sync check start) and the latency check interval is used as the interval between sync check attempts.

The possible values are:

- *CLUSTER—Uses the system value that is assigned in the DMSETSVAL (Set Cluster System Values) command. This is the default value.
- *YES—Enables continuous sync checks.
- *NO—Does not enable a continuous sync check.

DELAY

Specifies the delay time between object checks during sync checks.

You can minimize the interference from sync check processing on regular object replication when the sync

check is done through replication jobs. The benefit of doing sync check through the replication jobs is a more accurate sync check, since the sync check is synchronous with object replication. However, sync check can be a CPU-intensive process and might cause unnecessary delays in the regular replication processing if there is not a delay between the attribute checking of individual objects.

The possible values are:

- *CLUSTER—Uses the system value that is assigned in the DMSETSVAL (Set Cluster System Values) command. This is the default value.
- <time>—A number of milliseconds between 0 - 10000.

SCTYPE

Specifies the type of continuous synchronization check to be effected.

The possible values are:

- *CLUSTER—Uses the system value that is assigned in the DMSETSVAL (Set Cluster System Values) command. This is the default value.
- *FILEATTR—Checks whether the *FILE objects on the primary node exist on the backup node. It also checks that the attributes of the primary node *FILE objects match the attributes of the backup node *FILE objects.
- *OBJATTR—Checks whether objects exist on the backup node. Also checks their authorities and journaling attributes (for journaled objects). BSF and native objects are checked.
- *FULL—Performs an *OBJATTR sync check on all objects that are replicated by the group and a *FILEATTR sync check on database file objects. Also performs a *CHKOBSLETE sync check to identify objects that exist on the backup node but do not exist on the primary node.

AUTREPAIR

Specifies whether to refresh out-of-sync private authorities when the private authorities, ownership and primary group of objects, whose authorities (including ownership and primary group) are found to be mismatched by a full (*FULL) or object attribute (*OBJATTR) sync check. This parameter applies only when the value of the **Enable continuous sync check** (CONT) parameter is *YES.

The possible values are:

- *CLUSTER—Uses the system value that is assigned in the DMSETSVAL (Set Cluster System Values) command. This is the default value.
- *YES—Refresh out-of-sync private authorities.
- *NO—Do not refresh out-of-sync private authorities.

JRNREPAIR

Specifies whether to repair mismatched journaling of objects on the backup node as part of the sync check processing when the objects on the backup node are found to be out-of-sync by a full (*FULL) or object attribute (*OBJATTR) sync check. This parameter applies only when the value of the **Enable continuous sync check** (CONT) parameter is *YES.

The possible values are:

- *CLUSTER—Uses the system value that is assigned in the DMSETSVAL (Set Cluster System Values) command. This is the default value.
- *YES—Repair mismatched journaling so that the objects on the backup node are journaled to the same journal as the objects on the primary node.
- *NO—Do not repair mismatched journaling.

CBUREPAIR

Specifies that database *FILE objects whose "Created by user" attribute is found to be mismatched by a *FULL or *OBJATTR sync check, are refreshed as part of the sync check processing. This parameter is only used if the **CONTSYNC** parameter (Enable continuous sync check) is *YES.

The possible values are:

- *CLUSTER—Uses the cluster system value assigned in the **DMSETSVAL** command.
- *NO—Does not refresh out-of-sync created by user for database *FILE objects.
- *YES—Refreshes out-of-sync created by user for database *FILE objects as part of continuous sync check.

CRDREPAIR

Specifies that library (native) objects whose "Creation date and time" attribute is found to be mismatched by a *FULL or *OBJATTR sync check, are refreshed as part of the sync check processing. This parameter is only used if the **CONTSYNC** parameter (Enable continuous sync check) evaluates to *YES.

The possible values are:

- *CLUSTER—Uses the cluster system value assigned in the **DMSETSVAL** (Set Cluster System Values) command.
- *NO—Does not repair mismatched creation date and time for library (native) objects.
- *YES—Repairs mismatched creation date and time for library (native) objects as part of continuous sync check.

ACT

Settings for automatic reactivation of suspended objects that are replicated by a group. The **Automatic reactivation** system value must be set to *YES. The following parameters combine to define the automatic reactivation settings at the group level.

Automatic reactivation—Specifies to try to reactivate suspended objects that are replicated by the object specifier. The possible values for **Automatic reactivation** are:

- *CLUSTER—Uses the product system value. By default, this value is used.
- *NO—Does not attempt to reactivate suspended objects.

Max. reactivation attempts specifies the number of automatic retries before ending reactivation attempts.

The possible values for **Max. reactivation attempts** are:

- *CLUSTER—Uses the product system value. By default, this value is used.
- *NOMAX—Specifies no maximum reactivation attempts.
- <number>—Specifies the maximum number of retries. Valid entries range between 1 - 32767.

Max. reactivation size specifies the maximum size of an object to include in the reactivation. Large object size impacts network performance and can introduce mirroring latency. The possible values for **Max. reactivation size** are:

- *CLUSTER—Uses the product system value. By default, this value is used.
- *NOMAX—Includes objects of all sizes in the reactivation. This value might cause performance issues on the primary node if very large objects are frequently locked by other jobs.
- <size>—Specifies the maximum size (in bytes) of an object.

Activate files in ACTOOS state specifies to reactivate files that are replicated by this group and are in out-of-sync (ACTOOS) state. Files that are in ACTOOS state are not suspended, but encountered record levels errors on the backup node during replication. The Maximum record level errors value for these files is set to *FIX. When activated, only the records in error are refreshed to the backup node, not the entire file. Files whose contents are currently being read for a *CHECKSUM sync check are not reactivated even if they are in ACTOOS state. (This restriction applies only to the files on the primary node.) The possible values for **Activate files in ACTOOS state** are:

- *CLUSTER—Uses the product system value. By default, this value is used.
- *NO—Does not attempt to refresh records in error for files in ACTOOS state.

-
- *YES—Specifies to reactivate files that are replicated by this group and are in out-of-sync (ACTOOS) state. This value is available only if automatic reactivation is enabled for the group.

MAXRLE

Specifies the maximum number of record level errors (RLE) that are allowed before suspending a file on the backup node. Record level errors can occur during the apply process.

The possible values are:

- *CLUSTER—Uses the product system value. By default, this value is used.
- *NOMAX—Specifies to continue replication of physical files for any number of record-level errors.
- *FIX—Specifies to continue replication of physical files when record-level errors occur during the backup apply process, but changes the state of the file to allow it to be fixed in a subsequent activation using the DMACTOBJ (Activate Object) command. File repair at activation is supported for files that are suspended with RLE. For more information about this value, refer to the ***FIXREC** option on the DMACTOBJ (Activate Object) command.
- <number>—Specifies the maximum number of record level errors.

HADELAY

Specifies the following:

- The delay in seconds to wait before processing the create, rename, move, or activate journal entries for journaled objects (database files, data areas, data queues) and IFS files.
- The delay that iCluster will wait before refreshing a savefile when an object is saved into the savefile (resulting in a T ZC journal entry for the savefile).
- The delay in saving a spooled file when the audit journal scraper processes a T SF journal entry with the action code C (create).

This delay is only done on the primary node. The maximum delay is 1800 seconds (30 minutes).

The possible values are:

- *CLUSTER—Uses the product system value.
- <time>—A number of seconds from 1 - 1800.

SAVRETRY

Specifies the number of times to retry a save operation, if the save operation is not initially successful. The retries are performed only on the primary node. The wait time between each retry is approximately 1 second.

The possible values are:

- *CLUSTER—Uses the product system value. By default, this value is used.
- <number>—Specifies the maximum number of retries. Valid entries range between 1 - 100.

LOCKCHKS

Specifies the number of times to check if an object is locked by another process on the backup node. Locked objects prevent replication operations from being performed on the objects. The wait time between checks is defined by the **Delay between lock checks (LOCKCHKDLY)** parameter. The locked object checks end when the number of locked object checks is reached, or until the object is no longer locked.

The possible values are:

- *CLUSTER—Uses the product system value. By default, this value is used.
- *NOMAX—Specifies to keep checking until the object is no longer locked.
- <number>—Specifies the maximum number of checks. The default value is 5.

LOCKCHKDLY

Specifies the delay in seconds between checks of a locked object. This delay for checks of locked objects is performed only on the backup node.

The possible values are:

- *CLUSTER—Uses the product system value. By default, this value is used.
- <time>—A number of seconds from 1 - 60.

APYDELAY

Specifies the delay in minutes between the time that a journal entry was originally deposited in the source journal and the time that the same journal entry is applied on the backup node. It is assumed that the UTC offsets and system times are the same on the primary and the backup nodes.

The apply delay is not used in the following situations:

- When staging store is drained and the apply processes are started with DMSTRAPY FRCDRN(*YES).
- When the staging store is drained before the roles are changed in a group roleswitch.

When replication is ended with DMENDGRP OPTION(*CTRLD), the apply processes continue to run until all journal entries are applied.

When replication is ended with DMENDAPY OPTION(*CTRLD), the apply processes end on the next transaction boundary, regardless of the apply delay setting.

To ensure that journal entries in the staging store are not fully applied, and to use the apply delay setting, use the DMENDAPY OPTION(*CTRLD) before you end replication.

The possible values are:

- *CLUSTER—Uses the product system value. By default, this value is used.
- <time>—A number of minutes from 0 - 3000.

ALWOOS

Specifies whether to allow the group to have a non-zero out-of-sync object count when determining the role switch readiness to display in the status monitors.

The possible values are:

- *NO—Do not allow out-of-sync objects. If out-of-sync objects are detected, the group is not ready for a switchover and the role switch indicator is blank. By default, this value is used.
- *YES—Allow out-of-sync objects. The role switch indicator can be Y or blank.

ALWSUS

Specifies whether to allow the group to have a non-zero suspended object count when determining the role switch readiness to display in the status monitors.

The possible values are:

- *NO—Do not allow suspended objects. If suspended objects are detected, the group is not ready for a switchover and the role switch indicator is blank. By default, this value is used.
- *YES—Allow suspended objects. The role switch indicator can be Y or blank.

ALRTMON

Specifies whether to monitor the group as part of the replication monitoring process that is performed by the DMMONHA job.

The possible values are:

- *YES—Monitor the group as part of the replication monitoring process. By default, this value is used.
- *NO—Do not monitor the group as part of the replication monitoring process. Monitoring excludes this group and does not send alerts when changes in the group status occur or when objects are suspended.

CHKLATNCY

Specifies whether to check the group latency as part of the latency check process.

The possible values are:

- *YES—Check the group latency as part of the latency check process. By default, this value is used.
- *NO—Do not check the group latency as part of the latency check process. Latency checking excludes this group and does not send alerts. The latency columns in the status monitors are blank. The group latency is not recorded in the product history log.

AUTOSTART

Specifies whether to automatically start the group when iCluster processes are started with the DMSTRCST (Start iCluster Processes) command.

The possible values are:

- *YES—Automatically start the group when iCluster processes are started with the DMSTRCST (Start iCluster Processes) command. By default, this value is used.
- *NO—Do not automatically start the group when iCluster processes are started with the DMSTRCST (Start iCluster Processes) command.

RUNPNTY

Specifies the run priority of the jobs for the group. The run priority ranges from 1 (highest) to 99 (lowest). The jobs running the following programs use the run priority: OMGROUJOB, HADDAS, HADSFP, HADDJS, DMSQJS, OMRCV, OMTARGET, HADDASR, HADSFPR, HADDJSR, HADTUP, RCVMQMJRN, THADDAS, SC_SOURCE, SC_TARGET, DM_PRIMPRE, and DM_BACKPRE.

The possible values are:

- *PRDDFT—Keeps the product default value setting for this parameter. The default value is the run priority from the DMCLUSTER object of type *CLS in the product library. By default, this value is used.
- nn—Specifies the running priority value, where "nn" represents the run priority from 1 (highest) to 99 (lowest)

Example 1

```
DMADDGRP GROUP(GRP1) GRPTYPE(*REPL) DMNSRC(*LIST) PRIMNODE(NODE1)
PRIMIASP(DMC_IASP1) BACKUPS(NODE2) BACKIASP(*SYSBAS) TGTLIB(*PRIMARY)
MSGQUEUE(LIB1/MSGQ1) ROLESWITCH(*YES) CHKJRN(*YES)
BSWUSREXIT(LIB1/BFEXIT) ASWUSREXIT(LIB1/AFEXIT)
EXITDATA(ARJ123908KPJ230982) POLLINT(003000) SAVACT(*NO)
DFTDBJRN(LIB1/JRN1) JRNLOC(*LOCAL) CMTLVL(*LEVEL2) JRNBA(*BOTH)
OPTIMZAPY(*YES) DFTBSFJRN(LIB2/JRN2) DESC('NY/LA') AUTREPAIR(*YES)
JRNREPAIR(*YES) CBUREPAIR(*YES) CRDREPAIR(*YES) ALWOOS(*YES) ALWSUS(*YES) ALRTMON(*NO) CHKLATNC
Y(*NO) AUTOSTART(*NO)
```

Adds the replication group **GRP1**. The primary and backup nodes in the group are explicitly identified through the **PRIMNODE** and **BACKUP** parameters.

The primary node in the group is **NODE1**, and the backup node is **NODE2**. The group replicates objects from the independent auxiliary storage pool called **DMC_IASP1** on **NODE1** into ***SYSBAS** on **Node2**.

The backup library that is specified with the **TGTLIB** parameter is set so that it is the same as the primary node library where the object resides.

Messages that are generated when a failover occurs are added to the message queue **MSGQ1** in library **LIB1**. The backup node in the group becomes the primary node if a failover occurs.

In the event of a failover or switchover, iCluster checks if all mirrored objects that should be journaled are being journaled on the new primary node.

User exit programs that are specified through the **BSWUSREXIT** and **ASWUSREXIT** parameters are called when failover or switchover occurs.

The user exit program **BFEXIT** in library **LIB1** is called immediately before a role change is performed.

The user exit program **AFEXIT** in library **LIB1** is called immediately after a role change is performed.

User-defined data consisting of a sequence of characters is passed to the user exit programs.

The polling interval is 30 minutes for pollable objects that are selected for replication within the group.

Objects that are in use cannot be saved by the product at the same time.

The default database journal is **JRN1** in library **LIB1**. Database file changes are applied with *LEVEL2 commitment control.

The journal location is set to *LOCAL. Database journal scraping occurs on the group primary node.

Both before and after images are journaled for changes to database files.

Database apply entries are optimized.

The default BSF journal is **JRN2** in library **LIB2**.

The description indicates that the nodes in the group are located in New York and Los Angeles.

The out-of-sync private authorities are refreshed as part of the full (*FULL) or object attribute (*OBJATTR) sync check process when the private authorities, ownership and primary group of objects are found to be mismatched.

The mismatched journaling is repaired as part of the full (*FULL) or object attribute (*OBJATTR) sync check process so that the objects on the backup node are journaled to the same journal as the objects on the primary node.

The group allows out-of-sync objects and suspended objects when determining switch readiness.

The group is not monitored as part of the replication monitoring process.

The group latency is not checked as part of the latency check process.

The group does not automatically start the group when iCluster processes are started with the DMSTRCST (Start iCluster Processes) command.

Example 2

```
DMADDGRP GROUP(GRP1) GRPTYPE(*MQSERIES) DMNSRC(*LIST) PRIMNODE(NODE1)
BACKUPS(NODE2) MQVERSION(V5R2M0) QMNAME(QMANAGER) MSGQUEUE(LIB1/MSGQ1)
ROLESWITCH(*YES) BSWUSREXIT(LIB1/BFEXIT) ASWUSREXIT(LIB1/AFEXIT)
EXITDATA(ARJ123908KPJ230982) DESC('NY/LA') RUNPTY(99)
```

Adds the group **GRP1** for WebSphere MQ objects. The primary and backup nodes in the group are explicitly identified through the **PRIMNODE** and **BACKUP** parameters.

The primary node in the group is **NODE1**, and the backup node is **NODE2**.

The version of WebSphere MQ that is installed on the primary and backup nodes is Version V5R2M0.

The queue manager name for WebSphere MQ is **QMANAGER**.

Messages that are generated when a failover occurs are added to the message queue **MSGQ1** in library **LIB1**.

The backup node in the group is configured to become the primary node if a failover occurs.

User exit programs that are specified through the **BSWUSREXIT** and **ASWUSREXIT** parameters are called if a failover or switchover occurs.

The description indicates that the nodes in the group are located in New York and Los Angeles.

The run priority is set to be the lowest priority possible.

Example 3

```
DMADDGRP GROUP(GRP1) GRPTYPE(*REPL) DMNSRC(*LIST) PRIMNODE(NODE1)
PRIMIASP(DMC_IASP1) BACKUPS(NODE2) BACKIASP(*SYSBAS) TGTLIB(*PRIMARY)
MSGQUEUE(LIB1/MSGQ1) ROLESWITCH(*YES) CHKJRN(*YES) POLLINT(003000)
SAVACT(*NO) DFTDBJRN(LIB1/JRN1) JRNLOC(*LOCAL) CMTLVL(*LEVEL2)
JRNBA(*BOTH) PFRFSHMTD(*SAVRST) OPTIMZAPY(*YES) DFTBSFJRN(LIB2/JRN2) DESC('NY/LA')
SAVTRY(5) LOCKCHKS(6) LOCKCHKDLY(3) APYDELAY(5)
```

Adds the replication group **GRP1**. The primary and backup nodes in the group are explicitly identified through the **PRIMNODE** and **BACKUP** parameters.

The primary node in the group is **NODE1**, and the backup node is **NODE2**. The switchable disk storage device on **NODE1** is **DMC_IASP1** and the group will replicate from and **IASP** to ***SYSBAS** on **Node2**.

The backup library that is specified with the **TGTLIB** parameter is set so that it is the same as the primary node library where the object resides.

Messages that are generated when a failover occurs are added to the message queue **MSGQ1** in library **LIB1**.

The backup node in the group becomes the primary node if a failover occurs.

In the event of a failover or switchover, iCluster checks if all mirrored objects that should be journaled are being journaled on the new primary node.

The polling interval is 30 minutes for pollable objects that are selected for replication within the group.

Objects that are in use cannot be saved by the product at the same time.

The default database journal is **JRN1** in library **LIB1**. Changes are applied with ***LEVEL2** commitment control.

The journal location is set to ***LOCAL**. Database journal scraping occurs on the group's primary node.

Both before and after images are journaled for changes to database files.

Physical database files are refreshed.

Database apply entries are optimized.

The default BSF journal is **JRN2** in library **LIB2**.

Description indicates that the nodes in the group are located in New York and Los Angeles.

Save operations that are not initially successful are retried up to 5 times.

Locked objects are checked up to 6 times.

Specifies a 3-second delay between checks of a locked object.

Specifies a 5-minute delay between the time that a journal entry is deposited on the source journal and the time that the same journal entry is applied on the backup node.

Restrictions

You must issue this command from an active node. All nodes in the added group must be active.

Minimum Authority Level

*ADMIN

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **6**

DMCHGGRP (Change Group command)

Changes one or more attributes of an existing replication group or, in certain cases, an existing refresh group.

```
DMCHGGRP GROUP() GRPTYPE() TGTLIB() STGSTORLIB() MSGQUEUE() ROLESWITCH()
CHKJRN() BSWUSREXIT() ASWUSREXIT() EXITDATA() POLLINT() SAVACT()
DFTDBJRN() JRNLOC() JRNBACKUP() CMTLVL() JRNBA() PFRFSHMTD() OPTIMZAPY() DFTBSFJRN()
RCVRYEXP() JRNKEY() LCKBACKUP() CFGSRCHLD() DESC() CONTSYNC() DELAY()
SCTYPE() AUTREPAIR() JRNREPAIR() CBUREPAIR() CRDREPAIR() ACT() MAXRLE() HADELAY()
SAVTRY() LOCKCHKS() LOCKCHKDLY() APYDELAY() ALWOOS() ALWSUS() ALRTMON()
CHKLATNCY() AUTOSTART()
```

This command does not change which primary and backup nodes are in the replication group. To add and remove backup nodes, use the DMADDBACK (Add Backup Node to Recovery Domain) and DMRMVBACK (Remove Backup Node from Recovery Domain) commands.

Furthermore, the following parameters cannot be changed if the group has type *RFSH:

- *CHKJRN (Check journaling at roleswitch)
- *POLLINT (Group polling interval)
- *DFTDBJRN (Default database journal)
- *JRNLOC (Journal location)
- *JRNBACKUP (Journal objects on backup)
- *CMTLVL (Commitment control level)
- *JRNBA (PF journal images)
- *DFTNSFJRN (Default BSF journal)

- *RCVYEXP (Recovery exposure)
- *JRNKEY (Recovery journal key)
- *CONTSYNC (Enable continuous sync check)
- *DELAY (Delay between objects)
- *SCTYPE (Sync check type)
- *AUTREPAIR (Auto-repair authority)
- *JRNREPAIR (Auto-repair journaling)
- *CBUREPAIR (Auto-repair created by user)
- *CRDREPAIR (Auto-repair creation date and time)
- *ACT (Automatic reactivation)
- *MAXRLE (Max. record level errors)
- *HADELAY (Delay for object processing)
- *APYDELAY (Apply delay)
- *ALWOOS (Allow OOS for switch readiness)
- *ALWSUS (Allow SUS for switch readiness)
- *ALRTMON (Enable replication monitoring)
- *CHKLATNCY (Check group's latency)
- *AUTOSTART (Auto-start group)

If one of these parameters is changed for a group of a type *RFSH, the change is ignored.

Input Parameters

GROUP

The name of the replication group to change.

GRPTYPE

Specifies the type of group to define. The group type can only be changed from the types *REPL, *RFSH, and *STDBY, and can only be changed to one of these types, or to the type *HADR. An *HADR group can only be changed to a *STDBY group.

Note: This parameter does not apply to Switchable Device groups and WebSphere MQ groups.

The possible values are:

- *SAME—Keeps the current setting for this parameter.
- *REPL—Specifies a group for continuous object replication between the primary and backup nodes.
- *RFSH—Specifies a refresh-only group.
- *STDBY—Specifies a stand-by group that can be converted to a replication or refresh-only group at a later time; for example, in a disaster recovery situation.

Note: A group with type *STDBY will always have a status of *NONE. It cannot:

- be started (and consequently cannot be ended)
- have marked positions or set positions
- be seen in the Status Monitor
- be considered as a group for RTVHAPOS, DSPHAPOS, journal management, and other group operations related to replication
- have a role switch



When the group must be used, run the DMCHGGRP command to change the group type to *REPL, do whatever pre-start processing must be done (DMSETPOS or DMMRKPOS), and then start the group.

When a group is changed from *REPL to *STDBY, the group's metadata (other than the group itself and its object specifiers) will be cleared, the same as when a group undergoes a roleswitch.

- *HADR—Specifies a replication group intended for both high availability (HA) and disaster recovery (DR). An *HADR group has one primary node and two backup nodes: one for high availability and one for disaster recovery. When a roleswitch is done with an *HADR group, the HA backup node becomes the new primary node and the previous primary node becomes the new HA backup node. The role of the DR backup node is not changed. Replication in an *HADR group is from the primary node to both of the backup nodes.

TGTLIB

The name of the library on the backup node that receives the replicated objects.

The possible values are:

- *SAME—Keeps the current setting for this parameter.
- <name>—The library name.
- *PRIMARY—Sets the backup node library to be identical to the primary node library where the object resides. If the primary and backup environments reside on the same physical system (local loopback replication), the target library that you specify cannot be the same library where a selected object in the group currently resides.

STGSTORLIB

The name of the staging store library for the group.

The staging store is a nonvolatile storage mechanism for the changes that are sent from the primary node to be applied to the backup node. The staging store holds these changes before they are applied to the backup node to allow the backup apply processes to be shut down temporarily without shutting down the journal scraper processes. Specify a staging store library for each group to improve performance and minimize overall latency.

The possible values are:

- *SAME—Keeps the current setting for this parameter.
- *NODE—Sets the staging store library to the library that is set for the node.
- <name>—A unique staging store library name that is not used for another group. It is recommended to specify a library that does not exist on your server. The library is created if it does not exist. After you change the name of the staging store library, restart the group with a refresh or **Use marked positions** option after DMMRKPOS, or after the DMSETPOS command is run for all of the group's journals.

MSGQUEUE

The name of the message queue on the new primary node that receives messages that are generated when a failover occurs.

This parameter applies only to replication groups that have the value of the **Target library** field set to *PRIMARY, and to WebSphere MQ groups.

This parameter applies only to replication groups.

The possible values are:

- <name>—Specifies the message queue name.
- *SAME—Keeps the current setting for this parameter.
- *NONE—Specifies that no message queues are used.

The library where the message queue resides must be identified if you do not specify *NONE. Prefix the message queue with the name of the library where the queue is located. For example:

LIB1/MSGQ1

ROLESWITCH

Indicates if the role of the backup node changes automatically so that it becomes the primary node in the group when a failover occurs.

This parameter applies only to replication groups that have the value of the **Target library** field set to *PRIMARY, and to WebSphere MQ groups.

The functions of the primary node is moved to the backup node if this parameter is set to *YES when a failover occurs.

The possible values are:

- *SAME—Keeps the current setting for this parameter.
- *YES—Changes the role of the backup node in the group automatically to the primary node when a failover occurs. Groups with a status of *ACTIVE are switched over if the primary node fails. The user exit programs that are specified through the **BSWUSREXIT** and **ASWUSREXIT** parameters are called.
- *NO—Does not change the role of the backup node in the group automatically to the primary node when a failover occurs. This option specifies whether to continue with a role switch, or continue with the current configuration.

With this setting, a failure of the group primary node is declared by the failover mechanism, although iCluster does not prepare the backup node to take over as the primary node. Replication does not start automatically after the failed primary node is again active in the cluster.

! **Note:** The following parameters cannot be changed if the group has type *RFSH and it is not roleswitch-enabled:

- *MSGQUEUE (Failover message queue name and library)
- *ROLESWITCH (Do roleswitch at failover)
- *BSWUSREXIT (User exit name and library before roleswitch)
- *ASWUSREXIT (User exit name and library after roleswitch)
- *EXITDATA (Exit program data)

CHKJRN

When switching to the new primary node, specifies to check if objects are being properly journaled on the node and start journaling them. Certain types of objects must be journaled before replication starts. However, because journaling objects can be time consuming, this option specifies whether journaling should be started for objects during a switchover, depending on your business requirements.

Depending on the value of the group **CHKJRN** parameter, processing that is equivalent to the DMMRKPOS (Mark Journal Positions) command might occur during a role switch procedure.

If the value of the group **CHKJRN** parameter is *NO, specifies that no DMMRKPOS (Mark Journal Positions) processing occurs. The replication metadata for the new primary node is generated based on the existing replication metadata on the current backup node. In case of an error in processing the backup replication metadata, or nonexistent backup replication metadata, DMMRKPOS (Mark Journal Positions) processing occurs

regardless of the **CHKJRN** parameter value.

The DMMRKPOS (Mark Journal Positions) command establishes the set of mirrored objects that are based on the actual objects that match specifiers on the backup system, regardless if they were previously mirrored or not, and starts journaling for the mirrored objects that should be journaled.

If the value of the group **CHKJRN** parameter is *YES, DMMRKPOS (Mark Journal Positions) processing occurs during a role switch, and the backup replication metadata is not used for setting up the replication metadata for the new primary node.

The possible values are:

- *SAME—Keeps the current setting for this parameter.
- *YES—Checks if all mirrored objects that should be journaled are being journaled on the new primary node. If not, the objects are journaled.
- *NO—Does not check if all mirrored objects that should be journaled are journaled on the new primary node. You must make sure that all objects that must be journaled are being journaled to the correct journals, and that the objects were journaled before replication starts on the new primary node.

BSWUSREXIT

The name of the user exit program to call immediately before the role change is performed.

This parameter applies only to replication groups with the value of the **Target library** field set to *PRIMARY.

The possible values are:

- <name>—Specifies the user exit program name. The **ROLESWITCH** parameter must be set to *YES to call the program when a failover occurs. The user exit program will always be called (regardless of the **ROLESWITCH** parameter setting) when a switchover is initiated.
- *SAME—Keeps the current setting for this parameter.
- *NONE—Does not call a user exit program.

Identify the library where the user exit program resides when *SAME or *NONE is not specified. Prefix the name of the user exit program name with the name of the library where the program is located. For example:

LIB1/USREXIT1

ASWUSREXIT

The name of the user exit program to call immediately after the role change is performed.

This parameter applies only to replication groups with the value of the **Target library** field set to *PRIMARY.

The possible values are:

- <name>—Specifies the user exit program name. The **ROLESWITCH** parameter must be set to *YES to call the program when a failover occurs. The user exit program will always be invoked (regardless of the **ROLESWITCH** parameter setting) when a switchover is initiated.
- *SAME—Keeps the current setting for this parameter.
- *NONE—Does not call a user exit program.

Identify the library where the user exit program resides when *SAME or *NONE is not specified. Prefix the name of the user exit program name with the name of the library where the program is located. For example:

LIB1/USREXIT1

EXITDATA

Identifies the user-defined data to pass to the user exit programs that are specified through the **BSWUSREXIT**

and **ASWUSREXIT** parameters.

This parameter applies only to replication groups with the value of the **Target library** field set to *PRIMARY.

If you specify two different user exit programs (one program before the role switch, and a different program after the role switch), the same user-defined data is passed to both programs. A maximum of 256 bytes of data can be passed to the user exit programs. If your data contains spaces or non-alphanumeric characters (commas, periods, and so on), you must enclose your data in single quotes.

The possible values are:

- <data>—Specifies up to 256 bytes of data.
- *SAME—Keeps the current setting for this parameter.

POLLINT

The time interval that determines how often iCluster should check for content changes to user spaces, source physical files, and Document Library Objects (DLO).

This parameter applies only to replication groups.

You set the interval by specifying the number of hours (first two digits), minutes (middle two digits), and seconds (last two digits) between consecutive polls. The polling interval applies only when pollable objects (user spaces, source physical files, and DLO) are selected to the group through the DMSELOBJ (Select Objects to Group) command.

The possible values are:

- <time>—A time value from 000010 - 235959 expressed in *HHMMSS* format.
- *SAME—Keeps the current setting for this parameter.
- *CLUSTER—Uses the system value that is assigned in the DMSETVAL (Set Cluster System Values) command.
- *NONE—Does not poll objects for content changes.



Note: If *NONE is selected as the polling interval for an object specifier but the group polling interval is not *NONE, polling occurs for all pollable objects that are selected to the group that do not match the object specifier that has a polling interval of *NONE.

SAVACT

Indicates if an object can be updated at the same time it is being saved for refresh to the backup node.

This parameter does not apply to physical files because they cannot be modified while they are being saved.

This parameter applies only to replication groups.

The possible values are:

- *SAME—Keeps the current setting for this parameter.
- *YES—Allows objects in use by another job to be saved by the product. Objects may reach checkpoints at different times and may not be in a consistent state in relationship to each other.
- *NO—Does not allow objects in use by another job to be saved by the product.
- *SYNC—Allows objects in use by another job to be saved by the product. All of the objects reach a checkpoint together and are saved in a consistent state in relationship to each other.
- *CLUSTER—Uses the system value that is assigned in the DMSETVAL (Set Cluster System Values) command.

DFTDBJRN

The name of the database journal to use as the default.

This parameter applies only to replication groups.

The journal that you specify is used for files, data areas, and data queues that are to be mirrored but are not yet journaled. iCluster starts journaling automatically to this journal.

The possible values are:

- <name>—Specifies the database journal name.
- *SAME—Keeps the current setting for this parameter.
- *CLUSTER—Uses the system value that is assigned in the DMSETVAL (Set Cluster System Values) command.

The library where the journal resides must be identified if you do not specify *CLUSTER. Prefix the journal with the name of the library where the journal is located. For example:

LIB1/JRN1

- ! **Note:** If this group replicates data from an IASP device, then the journal must exist on the IASP device before the group can start replicating objects.

JRNLOC

The location of the database journal where scraping occurs.

The possible values are:

- *LOCAL—Scrapes the database journals on the primary node during replication.
- *REMOTE—Scrapes the remote database journals on the backup node during replication.
- *SAME—Keeps the current setting for this parameter.

After changing the journal location with this parameter, you must set the position of the local journal to the position of the last applied entry by using the DMSETPOS (Set Journal Start Position) command with the **JRNPOS(*LASTAPY)** parameter.

JRNBACKUP

Indicates whether to journal replicated physical files, data areas, data queues, and journaled BSF files on backup nodes.

This parameter applies only to *REPL and *MQSERIES groups.

The possible values are:

- *SAME—Keeps the current setting for this parameter.
- *CLUSTER—Uses the system value that is assigned in the DMSETVAL (Set Cluster System Values) command.
- *YES—Journals the replicated physical files, data areas, data queues, and journaled BSF files on backup nodes.
- *NO—Does not journal the physical files, data areas, data queues, and journaled BSF files on backup nodes.

- ! **Note:** For PFDTA files that are replicated with **MIRRCNTS(*NO)**, iCluster does not journal them on backup nodes even if the **JRNBACKUP** parameter is set to *YES.

CMTLVL

The level of commitment control to use when replicating *FILE objects.

This parameter applies only to replication groups.

Commitment control stages database transactions so that they are assembled before being applied, or additionally, opened in a commitment control environment to ensure that all updates are received. It also makes sure that the changes are applied in the correct sequence.

The possible values are:

- *SAME—Keeps the current setting for this parameter.
- *NONE—Commitment control staging will not be performed during the update process on the backup node.
- *LEVEL1—Assembles all updates that comprise a transaction before being applied on the backup node.
- *LEVEL2—Opens all updates in a transaction in a commitment control environment to ensure that all updates are received. This option provides true commitment control.
- *CLUSTER—Uses the system value that is assigned in the DMSETVAL (Set Cluster System Values) command.

If you select *LEVEL2, but a file on the backup node cannot be opened under commitment control, *LEVEL1 commitment control is used for that file.

JRNBA

Indicates if default journaling includes both before and after images.

This parameter applies only to replication groups.

Before images are necessary to remove applied or keyed replication updates. Also, if you specified the unique key update method for a file, then it must be journaled with both before and after images. If a file is replicated using relative record number, it may be journaled with after images only.

Both before and after images must be journaled if the **CMTLVL** parameter is set to *LEVEL2.

The possible values are:

- *SAME—Keeps the current setting for this parameter.
- *BOTH—Journals both before and after images.
- *AFTER—Journals only the after image.
- *CLUSTER—Uses the system value that is assigned in the DMSETVAL (Set Cluster System Values) command.

PFRFSHMTD

Specifies the method of physical file refresh when a physical file is refreshed as part of a group refresh.

This parameter applies only to replication groups.

This parameter only applies to groups of type *REPL or *RFSH.

The possible values are:

- *SAME—Keeps the current setting for this parameter.
- *CLUSTER—Uses the system value that is assigned in the DMSETVAL (Set Cluster System Values) command.
- *RBR—Record by record refresh.
- *SAVRST—Refresh by save/restore. Physical files are saved with the data and restored on the backup node. When the *SAVRST method is used to refresh files, the "Created by user" attribute of a file is retained on the backup node.



Note: The *SAVRST method of physical file refresh will not succeed if there are any other update or exclusive locks on the file being refreshed at the time of refresh, or if the file is under commitment control. A file's refresh by *SAVRST will also fail if the group's "Save active" parameter evaluates to *NO and the file is open or locked for update by another

! application. In these cases, you should specify the *RBR refresh method.

OPTIMZAPY

Indicates whether to optimize database apply entries.

Optimization can increase the apply performance for large files that are significantly larger than the shared memory pool, and have a large number of random updates applied to them.

The possible values are:

- *SAME—Keeps the current setting for this parameter.
- *NO—Does not enable optimization for database apply updates. By default, this value is used.
- *YES—Enables optimization for database apply updates.

DFTBSFJRN

The name of the journal to use as the default for journaling BSF files that are replicated by this group.

This parameter applies only to replication groups.

! **Note:** If this group replicates data from an IASP device, then the journal must exist on the IASP device before the group can start replicating objects.

The journal is used for BSF files that are to be mirrored and journaled.

The possible values are:

- <name>—Specifies the journal name.
- *SAME—Keeps the current setting for this parameter.
- *NONE—Journaling is not required for BSF replication.
- *CLUSTER—Uses the system value that is assigned in the DMSETVAL (Set Cluster System Values) command.

The library where the journal resides must be identified if you do not specify *NONE or *CLUSTER. Prefix the journal with the name of the library where the journal is located. For example:

LIB2/JRN2

RCVRYEXP

Specifies the number of journal entries that are applied between writing recovery checkpoints. Specifying a lower number decreases the performance of the apply, but provides a greater safeguard in recovery situations. Specifying a higher number lessens the impact of recovery checkpoints on the apply.

This parameter applies only to replication groups.

The possible values are:

- <number>—Specifies the number of journal entries from 0-2147483647.
- *SAME—Keeps the current setting for this parameter.
- *DISABLED—Does not generate recovery checkpoints.

JRNKEY

Specifies the key, or the identifier, to assign to a recovery checkpoint for the group combination. You must also specify this value in the RTVRCVPT (Retrieve Recovery Checkpoint) or RTVRCVPTR (Retrieve Recovery Checkpoint (CL Program)) command.

This parameter applies only to replication groups.

The possible values are:

- <key>—Specifies the key or identifier.
- *SAME—Keeps the current setting for this parameter.

LCKBACKUP

Indicates whether journaled files should be locked on the backup node. When locked, no other users can modify these files on the backup node.

The possible values are:

- *SAME—Keeps the current setting for this parameter.
- *CLUSTER—Uses the system value that is assigned in the DMSETSVAL (Set Cluster System Values) command.
- *YES—Locks file so that they cannot be modified on the backup node when the group is active.
- *NO—Does not lock files on the backup node when the group is active.

CFGSRCHLD

Indicates whether to automatically create configuration objects immediately after they are received on a backup node or hold the commands for creating them in specific source physical files so that they can be created later with the CRTCFOBJ (Create Configuration Objects) command.

If configuration objects that are being replicated already exist on backup nodes, this parameter should be set to *YES in order to prevent iCluster from trying to create objects when they are in use.

The possible values are:

- *SAME—Keeps the current setting for this parameter.
- *BACKUP—Uses the Hold configuration object source setting on the backup node of the group that is specified through the DMADDNODE (Add Node) command.
- *CLUSTER—Uses the system value that is assigned in the DMSETSVAL (Set Cluster System Values) command.
- *YES—Holds configuration objects in specific source physical files so that they can be created later with the CRTCFOBJ (Create Configuration Objects) command.
- *NO—Creates configuration objects as soon as they are received on the node.

CFGSCOPTS - Configuration object sync check options

Indicates whether a configuration objects should be reported as out of sync if the CL source member for creating the object exists on the backup node, even when the object itself does not exist on the backup node.

This parameter is used only when the value of **CFGSRCHLD** is set to *YES.

The possible values are:

- *SAME—Keeps the current setting for this parameter. By default, this value is used.
- *CLUSTER—Uses the cluster system value that was assigned in the DMSETSVAL (Set Cluster System Values) command.
- *RPTCFG—Configuration objects are reported as out of sync with the CFG reason code if the objects do not exist on the backup node but the CL source members for creating the objects exist on the backup node.
- *NORPTCFG—Configuration objects are not reported as out of sync if the CL source members for creating the objects exist on the backup node, even when the objects themselves do not exist on the backup node.

DESC

A short description to identify this group.

The possible values are:

- *SAME—Keeps the current setting for this parameter.
- <description>—Specifies the description text. You can use up to 50 characters.

CONTSYNC

Specifies that sync checks are continuously run, when the latencies (source receive and target apply) of the group that is being sync checked are below their respective latency thresholds and the group is active. If a sync check is not started because of excess latency, a message is issued the first time this happens (after a successful sync check start) and the latency check interval is used as the interval between sync check attempts.

The possible values are:

- *SAME—Keeps the current setting for this parameter.
- *CLUSTER—Uses the system value that is assigned in the DMSETVAL (Set Cluster System Values) command.
- *YES—Enables continuous sync checks.
- *NO—Does not enable a continuous sync check.

DELAY

Specifies the delay time, in milliseconds, between object checks during sync checks.

You can minimize the interference from sync check processing on regular object replication when the sync check is done through replication jobs. The benefit of doing sync check through the replication jobs is a more accurate sync check, since the sync check is thus truly synchronous with object replication. However, sync check can be a CPU-intensive process and might cause unnecessary delays in the regular replication processing if there is not a delay between the attribute checking of individual objects.

The possible values are:

- *SAME—Keeps the current setting for this parameter.
- <time>—A time value expressed in milliseconds (0 - 10000).
- *CLUSTER—Uses the system value that is assigned in the DMSETVAL (Set Cluster System Values) command.

SCTYPE

Specifies the type of continuous synchronization check to be effected.

The possible values are:

- *SAME—Keeps the current setting for this parameter.
- *FILEATTR—Checks whether the *FILE objects on the primary node exist on the backup node. It also checks that the attributes of the primary node *FILE objects match the attributes of the backup node *FILE objects.
- *OBJATTR—Checks whether objects exist on the backup node. Also checks their authorities and journaling attributes (for journaled objects). BSF and native objects are checked.
- *FULL—Performs an *OBJATTR sync check on all objects replicated by the group and a *FILEATTR sync check on database file objects. Also performs a *CHKOBSELETE sync check to identify objects that exist on the backup node but do not exist on the primary node.
- *CLUSTER—Uses the system value that is assigned in the DMSETVAL (Set Cluster System Values) command.

AUTREPAIR

Specifies whether to refresh out-of-sync private authorities when the private authorities, ownership and primary group of objects, whose authorities (including ownership and primary group) are found to be

mismatched by a full (*FULL) or object attribute (*OBJATTR) sync check. This parameter applies only when the value of the **Enable continuous sync check** (CONT) parameter is *YES.

The possible values are:

- *SAME—Keeps the current setting for this parameter.
- *YES—Refresh out-of-sync private authorities.
- *NO—Do not refresh out-of-sync private authorities.
- *CLUSTER—Uses the system value that is assigned in the DMSETVAL—Set Cluster System Values command.

JRNREPAIR

Specifies whether to repair mismatched journaling of objects on the backup node as part of the sync check processing when the objects on the backup node are found to be out-of-sync by a full (*FULL) or object attribute (*OBJATTR) sync check. This parameter applies only when the value of the **Enable continuous sync check** (CONT) parameter is *YES.

The possible values are:

- *SAME—Keeps the current setting for this parameter.
- *YES—Repair mismatched journaling so that the objects on the backup node are journaled to the same journal as the objects on the primary node.
- *NO—Do not repair mismatched journaling.
- *CLUSTER—Uses the system value that is assigned in the DMSETVAL (Set Cluster System Values) command.

CBUREPAIR

Specifies that database *FILE objects whose "Created by user" attribute is found to be mismatched by a *FULL or *OBJATTR sync check, are refreshed as part of the sync check processing. This parameter is only used if the **CONTSYNC** parameter (Enable continuous sync check) evaluates to *YES.

The possible values are:

- *CLUSTER—Uses the cluster system value assigned in the DMSETVAL (Set Cluster System Values) command.
- *NO—Does not refresh out-of-sync created by user for database *FILE objects.
- *YES—Refreshes out-of-sync created by user for database *FILE objects as part of continuous sync check.

CRDREPAIR

Specifies that library (native) objects whose "Creation date and time" attribute is found to be mismatched by a *FULL or *OBJATTR sync check, are refreshed as part of the sync check processing. This parameter is only used if the **CONTSYNC** parameter (Enable continuous sync check) evaluates to *YES.

The possible values are:

- *CLUSTER—Uses the cluster system value assigned in the DMSETVAL (Set Cluster System Values) command.
- *NO—Does not repair mismatched creation date and time for library (native) objects.
- *YES—Repairs mismatched creation date and time for library (native) objects as part of continuous sync check.

ACT

Settings for automatic reactivation of suspended objects that are replicated by a group. The **Automatic reactivation** system value must be set to *YES. The following parameters combine to define the automatic reactivation settings at the group level.

Automatic reactivation—Specifies to try to reactivate suspended objects that are replicated by the object specifier. The possible values for **Automatic reactivation** are:

- *SAME—Keeps the current setting for this parameter.
- *CLUSTER—Uses the system value that is assigned in the DMSETVAL (Set Cluster System Values) command.
- *NO—Does not attempt to reactivate suspended objects.

Max. reactivation attempts—Specifies the number of automatic retries before ending reactivation attempts. The possible values for **Max. reactivation attempts** are:

- *SAME—Keeps the current setting for this parameter.
- *CLUSTER—Uses the system value that is assigned in the DMSETVAL (Set Cluster System Values) command.
- *NOMAX—Specifies no maximum reactivation attempts.
- <number>—Specifies the maximum number of retries. Valid entries range between 1-32767.

Max. reactivation size—Specifies the maximum size of an object to include in the reactivation. Large object size impacts network performance and can introduce mirroring latency. The possible values for **Max. reactivation size** are:

- *SAME—Keeps the current setting for this parameter.
- *CLUSTER—Uses the system value that is assigned in the DMSETVAL (Set Cluster System Values) command.
- *NOMAX—Includes objects of all sizes in the reactivation. This value might cause performance issues on the primary node if very large objects are frequently locked by other jobs.
- <size>—Specifies the maximum size (in bytes) of an object.

Activate files in ACTOOS state specifies to reactivate files that are replicated by this group and are in out-of-sync (ACTOOS) state. The possible values for **Activate files in ACTOOS state** are:

- *SAME—Keeps the current setting for this parameter.
- *CLUSTER—Uses the system value that is assigned in the DMSETVAL (Set Cluster System Values) command.
- *NO—Does not attempt to refresh records in error for files in ACTOOS state.
- *YES—Specifies to reactivate files that are replicated by this group and are in out-of-sync (ACTOOS) state. This value is available only if automatic reactivation is enabled for the group.

MAXRLE

Specifies the maximum number of record level errors (RLE) that are allowed before suspending a file on the backup node. Record level errors can occur during the apply process.

The possible values are:

- *SAME—Keeps the current setting for this parameter.
- *CLUSTER—Uses the system value that is assigned in the DMSETVAL (Set Cluster System Values) command.
- *NOMAX—Specifies to continue replication of physical files for any number of record-level errors.
- *FIX—Specifies to continue replication of physical files when record-level errors occur during the backup process. File repair at activation is supported for files that are suspended with RLE. For more information about this value, refer to the ***FIXREC** option on the DMACTOBJ (Activate Object) command.
- <number>—Specifies the maximum number of record level errors.

HADELAY

Specifies the following:

- The delay in seconds to wait before processing the create, rename, move, or activate journal entries for journaled objects (database files, data areas, data queues) and IFS files.
- The delay that iCluster will wait before refreshing a savefile when an object is saved into the savefile (resulting in a T ZC journal entry for the savefile).
- The delay in saving a spooled file when the audit journal scraper processes a T SF journal entry with the action code C (create).

This delay is only done on the primary node. The maximum delay is 1800 seconds (30 minutes).

Specifies the delay in seconds to wait before processing the create, rename, move, or activate journal entries for journaled objects (database files, data areas, data queues) and IFS files. Also specifies the delay that iCluster will wait before refreshing a savefile when an object is saved into the savefile (resulting in a T ZC journal entry for the savefile). This delay is only done on the primary node. The maximum delay is 1800 seconds (30 minutes).

The possible values are:

- *SAME—Keeps the current setting for this parameter.
- *CLUSTER—Uses the system value that is assigned in the DMSETVAL (Set Cluster System Values) command.
- <time>—A number of seconds from 1 - 1800. The default value is 0.

SAVRETRY

Specifies the number of times to retry a save operation, if the save operation is not initially successful. The retries are performed only on the primary node. The wait time between each retry is approximately 1 second.

The possible values are:

- *SAME—Keeps the current setting for this parameter. This is the default value.
- *CLUSTER—Uses the system value that is assigned in the DMSETVAL (Set Cluster System Values) command.
- <number>—Specifies the maximum number of retries. Valid entries range between 1 - 100.

LOCKCHKS

Specifies the number of times to check if an object is locked by another process on the backup node. Locked objects prevent replication operations from being performed on the objects. The wait time between checks is defined by the **Delay between lock checks (LOCKCHKDLY)** parameter. The locked object checks end when the number of locked object checks is reached, or until the object is no longer locked.

The possible values are:

- *SAME—Keeps the current setting for this parameter.
- *CLUSTER—Uses the system value that is assigned in the DMSETVAL (Set Cluster System Values) command.
- *NOMAX—Specifies to keep checking the locked object until it is no longer locked.
- <number>—Specifies the maximum number of checks. The default value is 5.

LOCKCHKDLY

Specifies the delay in seconds between checks of a locked object. This delay for checks of locked objects is performed only on the backup node.

The possible values are:

- *SAME—Keeps the current setting for this parameter.
- *CLUSTER—Uses the system value that is assigned in the DMSETVAL (Set Cluster System Values) command.

-
- <time>—A number of seconds from 1 - 60.

APYDELAY

Specifies the delay in minutes between the time that a journal entry was originally deposited in the source journal and the time that the same journal entry is applied on the backup node. It is assumed that the UTC offsets and system times are the same on the primary and the backup nodes.

The apply delay is not used in the following situations:

- When staging store is drained and the apply processes are started with DMSTRAPY FRCDRN(*YES).
- When the staging store is drained before the roles are changed in a group roleswitch.

When replication is ended with DMENDGRP OPTION(*CTRLD), the apply processes continue to run until all journal entries are applied.

When replication is ended with DMENDAPY OPTION(*CTRLD), the apply processes end on the next transaction boundary, regardless of the apply delay setting.

To ensure that journal entries in the staging store are not fully applied, and to use the apply delay setting, use the DMENDAPY OPTION(*CTRLD) before you end replication.

The possible values are:

- *SAME—Keeps the current setting for this parameter.
- *CLUSTER—Uses the system value that is assigned in the DMSETSVAL (Set Cluster System Values) command.
- <time>—A number of minutes from 0 - 3000.

ALWOOS

Specifies whether to allow the group to have a non-zero out-of-sync object count when determining the role switch readiness to display in the status monitors.

The possible values are:

- *SAME—Keeps the current setting for this parameter.
- *NO—Do not allow out-of-sync objects. If out-of-sync objects are detected, the group is not ready for a switchover and the role switch indicator is blank.
- *YES—Allow out-of-sync objects. The role switch indicator can by Y or blank.

ALWSUS

Specifies whether to allow the group to have a non-zero suspended object count when determining the role switch readiness to display in the status monitors.

The possible values are:

- *SAME—Keeps the current setting for this parameter.
- *NO—Do not allow suspended objects. If suspended objects are detected, the group is not ready for a switchover and the role switch indicator is blank.
- *YES—Allow suspended objects. The role switch indicator can by Y or blank.

ALRTMON

Specifies whether to monitor the group as part of the replication monitoring process.

The possible values are:

- *SAME—Keeps the current setting for this parameter.
- *YES—Monitor the group as part of the replication monitoring process.

- *NO—Do not monitor the group as part of the replication monitoring process. Monitoring excludes this group and does not send alerts when changes in the group status occur or when objects are suspended.

CHKLATNCY

Specifies whether to check the group latency as part of the latency check process.

The possible values are:

- *SAME—Keeps the current setting for this parameter.
- *YES—Check the group latency as part of the latency check process.
- *NO—Do not check the group latency as part of the latency check process. Latency checking excludes this group and does not send alerts. The latency columns in the status monitors are blank. The group latency is not recorded in the product history log.

AUTOSTART

Specifies whether to automatically start the group when iCluster processes are started with the DMSTRCST (Start iCluster Processes) command.

The possible values are:

- *SAME—Keeps the current setting for this parameter.
- *YES—Automatically start the group when iCluster processes are started with the DMSTRCST (Start iCluster Processes) command.
- *NO—Do not automatically start the group when iCluster processes are started with the DMSTRCST (Start iCluster Processes) command.

RUNPTY

Specifies the run priority of the jobs for the group. The run priority ranges from 1 (highest) to 99 (lowest). The jobs running the following programs use the run priority: OMGROUPJOB, HADDAS, HADSFP, HADDJS, DMSQJS, OMRCV, OMTARGET, HADDASR, HADSFPR, HADDJSR, HADTUP, RCVMQMJRN, THADDAS, SC_SOURCE, SC_TARGET, DM_PRIMPRE, and DM_BACKPRE.

The possible values are:

- *SAME—Keeps the current setting for this parameter. By default, this value is used.
- *PRDDFT—Keeps the product default value setting for this parameter. The default value is the run priority from the DMCLUSTER object of type *CLS in the product library.
- nn—Specifies the running priority value, where "nn" represents from run priority from 1 (highest) to 99 (lowest)

Example 1

```
DMCHGGRP GROUP(GRP1) TGTLIB(TGT1) MSGQUEUE(LIB1/MSGQ1) ROLESWITCH(*YES)
BSWUSREXIT(*NONE) ASWUSREXIT(LIB1/AFEXIT) EXITDATA('ARJ 123908 KPJ 230982')
POLLINT(*SAME) SAVACT(*SYNC) DFTDBJRN(LIB1/JRN1) JRNLOC(*LOCAL) CMTLVL(*NONE)
JRNBA(*AFTER) OPTIMZAPY(*YES) RCVRYEXP(5000) JRNKEY(MYGROUP) DESC('LON/PAR')
AUTREPAIR(*YES) JRNREPAIR(*YES) CBUREPAIR(*YES) CRDREPAIR(*YES) ALWOOS(*YES) ALWSUS(*YES) ALRTMO
N(*NO)
CHKLATNCY(*NO) AUTOSTART(*NO) RUNPTY(*PRDDFT)
```

Changes are made to one or more attributes of the group **GRP1**.

TGT1 is the name of the library on the backup system that will receive the replicated objects.

Messages that are generated when a failover occurs are added to the message queue **MSGQ1** in library **LIB1**.

The backup node in the group will become the primary node if a failover occurs. The user exit program specified through the **ASWUSREXIT** parameter is called.

No user exit program is called immediately before a role change is performed.

The user exit program **AFEXIT** in library **LIB1** is called immediately after a role change is performed.

User-defined data that consists of a sequence of characters is passed to the user exit program. Note that single quotes are required to enclose data that includes spaces and other non-alphanumeric characters.

The polling interval for pollable objects that are selected for replication within the group is the current setting.

Objects can be updated and saved when they are being replicated within the group. All of the objects will reach a checkpoint together and are saved in a consistent state in relationship to each other.

The default database journal is **JRN1** in library **LIB1**.

The journal location is set to ***LOCAL**. Journal scraping will occur on the group primary node.

Changes are applied with no commitment control.

Only after images are journaled.

Database apply entries are optimized.

5000 journal entries are applied between recovery checkpoints.

The group **MYGROUP** is assigned to a recovery checkpoint for the group combination.

The description associated with the group indicates that the nodes are located in London and Paris.

The out-of-sync private authorities are refreshed as part of the full (***FULL**) or object attribute (***OBJATTR**) sync check process when the private authorities, ownership and primary group of objects are found to be mismatched.

The mismatched journaling is repaired as part of the full (***FULL**) or object attribute (***OBJATTR**) sync check process so that the objects on the backup node are journaled to the same journal as the objects on the primary node.

The group allows out-of-sync objects and suspended objects when determining switch readiness.

The group is not monitored as part of the replication monitoring process.

The group latency is not checked as part of the latency check process.

The group does not automatically start the group when iCluster processes are started with the **DMSTRCST** (Start iCluster Processes) command.

The run priority for the jobs of the group are set to the product default value setting.

Example 2

```
DMCHGGRP GROUP(GRP2) GRPTYPE(*RFSH)
```

Changes the replication group **GRP2** to a refresh-only group.

Restrictions

You must issue this command from an active node, and the group must be inactive.

Minimum Authority Level

*ADMIN

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **7**

Work With Groups screen - Option **2**

DMRMVGRP (Remove Group command)

Removes a group that was previously defined. A reference cannot be made to the group after it has been removed.

```
DMRMVGRP GROUP( ) RMVACTIVE( )
```

If the group is active, group operations are automatically stopped before the group is removed.

Input Parameters

GROUP

The name of the existing group to remove.

RMVACTIVE

Specifies whether to remove a group if the group is active.

The possible values are:

- *YES—Removes the active group.
- *NO—Does not remove the active group. By default, this value is used.

Example

```
DMRMVGRP GROUP(GRP1)
```

Removes the replication group **GRP1**.

Restrictions

You must issue this command on an active node in the cluster.

Minimum Authority Level

*ADMIN

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **8**

Work With Groups screen - Option **6**

DMADDBACK (Add Backup Node to Recovery Domain command)

Adds a backup node to the recovery domain for an existing group.

```
DMADDBACK NAME( ) NODE( ) BACKIASP( ) BKUPTYPE( )
```

You can add only a single backup node through this command. The recovery domain for the group must already have a defined primary node. You can add a backup node to the recovery domain only for inactive groups.

You must add a backup node to the recovery domain before replication can be started for a group.

Input Parameters

NAME

The name of the group that will have a backup node added to its recovery domain.

NODE

The name of the node in the cluster to add to the recovery domain for the group.

You must enable the **SWITCHRES** parameter for an *SWDEV group. The node must have switchable disk storage devices.

BACKIASP

The name of the independent auxiliary storage pool (IASP) on the backup node to which you want to replicate.

The possible values are:

- <name>—Specifies the IASP name.
- *SYSBAS—Uses a system ASP rather than an IASP. This is the default value.

BKUPTYPE

The type of backup node being added. This parameter is used only if the type of the group is *HADR.

The possible values are:

- *ONLY—Specifies that the node being added will be the only backup node for the group. This value is not valid for an *HADR group.
- *HABACKUP—Specifies that the node being added will be the HA backup node for the group.
- *DRBACKUP—Specifies that the node being added will be the DR backup node for the group.

Example

```
DMADDBACK NAME(GRP1) NODE(NODE1) BACKIASP(DMC_3) BKUPTYPE(*ONLY)
```

Adds the backup node **NODE1** to group **GRP1**. The name of the IASP on the backup node is **DMC_3**. The type of backup node must be *ONLY for a group that is not an *HADR group.

Restrictions

You must issue this command on an active node in the cluster. The node that is specified with this command must also be active.

Minimum Authority Level

*ADMIN

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **9**

Work With Groups screen - Option **22**

DMRMVBACK (Remove Backup Node from Recovery Domain command)

Removes the backup node from the recovery domain for an existing group.

```
DMRMVBACK NAME( ) NODE( )
```

The recovery domain for the group must already have defined primary and backup nodes. You can remove backup nodes from recovery domains only for inactive groups.

Input Parameters

NAME

The name of the group that will have a backup node removed from the recovery domain.

NODE

The name of the backup node in the cluster that is removed from the recovery domain for the group.

Example

```
DMRMVBACK NAME(GRP1) NODE(NODE1)
```

Removes the backup node **NODE1** from group **GRP1**.

Restrictions

You must issue this command on an active node in the cluster.

Minimum Authority Level

*ADMIN

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **10**

Work With Groups screen - Option **23**

DMDSPGRP (Display Group command)

On an active node in the cluster, displays the current settings for a replication group.

```
DMDSPGRP GROUP()
```

Input Parameters

GROUP

The name of the replication group to examine.

Example

```
DMDSPGRP GROUP(GRP1)
```

Displays the current settings for the group **GRP1**.

Restrictions

You must issue this command on an active node in the cluster.

Minimum Authority Level

*USER

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **11**

Work With Groups screen - Option **5**

DMWRKGRP (Work with Groups command)

Lists the groups in a cluster. Use the **BY**, **NODE**, and **STATUS** parameters to filter the list.

```
DMWRKGRP BY( ) NODE( ) STATUS( ) OUTPUT( )
```

The replication status that is currently assigned to the replication, refresh-only, and WebSphere MQ groups is displayed in the **Repl. status** column, and can be one of the following statuses:

- *ACTIVE—iCluster replication is being performed within the group.
- *INACTIVE—iCluster replication is not being performed within the group.
- *INDOUBT—Some, but not all of the group replication processes are running.
- *IN_ERROR—An internal iCluster error might have occurred. If this state persists, contact Technical Support.
- *UNKNOWN—The replication status of the group cannot be determined. This status might occur when the primary node in the group is inactive or there is no backup node in the group.
- *NONE—The group does not have a backup node; no replication is possible for the group.
- *SWOSTART—iCluster is starting switchover processing.
- *BEFUEXIT—iCluster is executing the before roleswitch user exit.
- *STSDRAIN—iCluster is draining the staging store.
- *STRJRN—iCluster is starting journaling on the journaled objects in replication scope.
- *TRIGGERS—Triggers are being enabled on the new primary node.

- *CONSTR—iCluster is enabling the referential integrity constraints on the new primary node.
- *CHKIDCOL—iCluster is checking the ID columns of files on the new primary node.
- *CHGUPRF—User profiles statuses are being changed on both primary and backup nodes.

Note:

*CHGUPRF status is shown only if it matches the following condition:

1. Run **DMSTRSWO** and **DMCHGROLE** with the parameter **SWOUPRF(*YES)**
2. The group to be switched should replicate *USRPRF objects.

When status is *CHGUPRF, iCluster exchanges the status of all profiles replicated by it between the primary and backup nodes. For example, user profile USRPRF01 is *ENABLED on the old primary node, and it is *DISABLED on the old backup node. After switchover, the status becomes *ENABLED on the new primary (old backup) and *DISABLED on the new backup (old primary).

- *CHGROLES—iCluster is changing the roles of the primary and backup nodes.
- *MRKPOS—iCluster is marking the starting position for replication on the new primary node.
- *AFTUEXIT—iCluster is executing the after roleswitch user exit.
- *RSS—A roleswitch simulation was started for the group.
- *ENDAPY—The group apply jobs were ended because a roleswitch simulation is being run on the group backup node.

The apply status of each replication group is indicated in the **Apply Status** column on the displayed screen, and can be one of the following statuses:

- *ACTIVE—Apply processes are active.
- *INACTIVE—Apply processes are inactive.
- *INDOUBT—Some, but not all, apply processes are running.
- *IN_ERROR—An internal iCluster error might have occurred. If this state persists, contact Technical Support.
- *UNKNOWN—The group's apply process status cannot be determined.

A blank field indicates that the group is not a replication group or it does not have a backup node.

Input Parameters

BY

Filters the set of groups that are listed by this command.

The possible values are:

- *NONE—Specifies all groups. By default, this value is used.
- *NODE—Specifies the groups that contain the node that is specified through the **NODE** parameter.
- *STATUS—Specifies the groups whose replication status or apply process status is the one specified on the **STATUS** parameter.

NODE

Filters the groups by the name of an existing node in the cluster. Groups that contain the node are listed.

This parameter applies only when the BY parameter is set to *NODE.

STATUS

Specifies the status of groups to be listed.

After you have selected the STATUS value on which to filter, the full list of groups is displayed, but the groups having the selected status are highlighted. Pressing **F16** toggles the display to show only the groups that have the selected status. Pressing **F16** again displays the full list of groups with the groups having the selected status are highlighted.

The possible values are:

- *ALL—Specifies all statuses, that is, all groups.
- *ACTIVE—Specifies groups that are fully active. All expected replication and apply processes for the groups are active.
- *INACTIVE—Specifies groups that are fully inactive. All expected replication and apply processes for the groups are inactive.
- *INDOUBT—Specifies groups that are only partially active. Only a subset of the expected replication and apply processes for the groups are active.
- *IN_ERROR—Specifies groups for which an error is detected when retrieving the group's replication status or apply status.
- *ENDAPY—Specifies groups where the journal scrape and receive processes are fully active but the apply processes are inactive.
- *NONE—Specifies groups that are not eligible for replication or cluster activity. For example, they do not have a backup node or they are *STDBY groups.

OUTPUT

Specifies where to write the report output.

The possible values are:

- *—Display the group list. By default, this value is used.
- *PRINT—Write the output to a spooled file.

Example

```
DMWRKGRP BY(*STATUS) STATUS(*ACTIVE)
```

Filter display by status value *ACTIVE.

Restrictions

You must issue this command on an active node in the cluster.

Minimum Authority Level

*USER

Menu Access

Main Menu - Option 2

iCluster Commands menu - F22 (Shift + F10) - Option **12**

Native IBM i object commands

Use the native IBM i object commands to select and deselect objects to groups, change object selections, work with object specifiers, send objects, select and deselect directory entries, and activate and suspend directory entries.

The following are the IBM i object commands with their usage:

- **DMSELOBJ (Select Objects to Group command)**: Use this command to select an object specifier to a replication group.
- **DMDSELOBJ (Deselect Objects from Group command)**: Use this command to deselect an object specifier from a replication group. Deselecting an object specifier from a group stops referenced objects from being replicated by the group.
- **DMCHGOBJSL (Change Object Selection command)**: Use this command to change specific attributes of an object specifier that is selected to a replication group.
- **DMWRKOBJ (Work with Object Specifiers command)**: Use this command to list the object specifiers. Use the **BY** and **GROUP** parameters to filter the list.
- **DMSNDOBJ (Send Object Immediately command)**: Use this command to immediately replicate one or more objects that are not part of a replication group to a backup node.
- **DMSELDIRE (Select Directory Entry command)**: Use this command to select directory entries for replication.
- **DMDSELDIRE (Deselect Directory Entry command)**: Use this command to deselect directory entries from a replication group.
- **DMSUSDIR (Suspend Directory Entry command)**: Use this command to suspend a specified directory entry to prevent it from being replicated to a backup node.
- **DMACTDIRE (Activate Directory Entry command)**: Use this command to activate one or more specified directory entries that are currently suspended from replication.

Related information

- [DMSELOBJ \(Select Objects to Group command\)](#)
- [DMDSELOBJ \(Deselect Objects from Group command\)](#)
- [DMCHGOBJSL \(Change Object Selection command\)](#)
- [DMWRKOBJ \(Work with Object Specifiers command\)](#)
- [DMSNDOBJ \(Send Object Immediately command\)](#)
- [DMSELDIRE \(Select Directory Entry command\)](#)
- [DMDSELDIRE \(Deselect Directory Entry command\)](#)
- [DMSUSDIR \(Suspend Directory Entry command\)](#)
- [DMACTDIRE \(Activate Directory Entry command\)](#)

DMSELOBJ (Select Objects to Group command)

Selects an object specifier to a replication group.

```
DMSELOBJ GROUP( ) OBJ( ) OBJTYPE( ) OBJATTR( ) TGTLIB( ) DESC( ) INCFLG( )
POLLINT( ) NEWOBJACT( ) MIRRCNTS( ) PFRFSHMTD( ) PFUPDMTD( ) PFKEY( ) ACT( ) MAXRLE( )
OUTQOPTS( ) EXCLMBR( ) HLDJOBSCD( )
```

The objects that are referenced by selected specifiers are replicated from the primary node to the backup node in the group. Use generic names to select a number of object specifiers to a group.

You must define the group before selecting object specifiers to the group.

-
- You can use this command when you add new object specifiers to active groups.
 - To synchronize objects on the primary and backup nodes through a save file or tape transfer, select the object specifiers that reference the objects before you save the objects. This process ensures that changes to the objects are audited between the time of the save and the time when replication is started.

Use this command to create object specifiers and indicate the specific library where the objects that match the object specifier should be replicated. Local loopback replication requires a backup library at the group level that is not the same as the source library that contains the objects that are being replicated.

Input Parameters

GROUP

The name of the defined group that will have objects selected to it.

OBJ

The object name component of the specifier to select.

The possible values are:

- <name>—Specifies the object name.
- <generic*>—Specifies a generic object name to identify multiple objects in a library.
- *ALL—Specifies all objects in a library.

The library where the objects reside must be identified.

The possible values are:

- <name>—The library name.
- <generic*>—A generic library name to identify multiple libraries. For example, JDE* identifies all libraries that begin with JDE, such as JDEPROD, JDEDEV, and so on.
- *ALL—Specifies all libraries. This value can only be used with *EXCLUDE specifiers for a specific object type.

OBJTYPE

The object type component of the specifier to select.

The possible values are:

- <type>—Object type. Press F4 for a list of all values.
- *ALL—Specifies all object types. *ALL cannot be used for Objects that can reside only in the QSYS library, that is, *LIB (library), *USRPRF (user profile) and the configuration object types *CNNL, *COSD, *CTLD, *DEVD, *IPXD, *LIND, *MODD, *NTBD, *NWID, *NWSID.

OBJATTR

The attribute component of the specifier to select.

This parameter is applicable only when the OBJTYPE parameter is set to *FILE or *DEVD.

This field is free-form. Consequently, you can enter any value to describe the object, as long as the value conforms to IBM i naming conventions. However, there are values that have special meaning to iCluster. They are PFSRC (source physical file), PFDTA (data physical file), and PF (any physical file). Other values listed are standard IBM i file sub-types. If PF is used, the object specifier will match either PFSRC or PFDTA files.

The possible values are:

- <type>—Attribute type. Press F4 for a list of values.

- *ALL—Specifies all object attributes. *ALL is not a valid IBM i object attribute but allows iCluster to gather all objects regardless of their attribute. By default, this value is used.

TGTLIB

The name of the library on the backup node that will receive the replicated objects.

The possible values are:

- <name>—The library name.
- *GROUP—Uses the same library on the backup node as specified in the DMADDGRP (Add Group) or DMCHGGRP (Change Group) commands. By default, this value is used.
- *PRIMARY—Sets the backup node library to be identical to the primary node library where the object resides. If the primary and backup environments reside on the same physical server (local loopback replication), the library that you specify cannot be the same library where a selected object in the group currently resides. By default, this value is used.

DESC

A short description that identifies this object specifier selection. You can use up to 50 characters.

INCFLG

Indicates whether to replicate the objects that are referenced by the specifier.

The possible values are:

- *INCLUDE—Replicates referenced objects within the group when cluster operations are started. Required if the specifier is being added to an active group. By default, this value is used.
- *EXCLUDE—Does not replicate referenced objects within the group when cluster operations are started. Prevents the referenced objects from being replicated to a backup environment.

POLLINT

The time interval that determines how often to check for content changes to user spaces and source physical files.

Set the interval by specifying time between consecutive polls. The polling interval applies only when the object type is set to *USRSPC (user space) and for source physical files.

The possible values are:

- <time>—A time value from 000010 - 235959 expressed in HHMMSS format.
- *GROUP—Uses the group value that is specified in the DMADDGRP (Add Group) or DMCHGGRP (Change Group) commands. By default, this value is used.
- *CLUSTER—Uses the system value that is assigned in the DMSETVAL (Set Cluster System Values) command.
- *NONE—Does not poll objects for content changes.

! **Note:** If *NONE is selected as the polling interval for a pollable object specifier but the group's polling interval is not *NONE, polling will occur for all pollable objects selected to the group that do not match the object specifier whose polling interval is *NONE.

NEWOBJACT

Indicates the method by which replication is to begin for the objects that come into replication scope when an object specifier is added.

The possible values are:

- *NONE—Preserves the replication status of new, in-scope objects. This value is intended to support initial group configuration. If this value is selected, replication must be started with a refresh of the entire group, or with the DMMRKPOS (Mark Journal Positions) or DMREGPOS (Register Positions) commands. The group replication status must be *INACTIVE if you use this value. By default, this value is used.
- *CURRENT—Begins replication of journal entries for new, in-scope objects at the time the object specifier is added. Journal entries that are related to newly in-scope objects that are created after that time are replicated. Journal entries that are related to new, in-scope objects that are created before that time are not replicated. The group replication status must be *ACTIVE for this option to take effect.

No changes should occur on the new, in-scope objects until the OMI0320 event for the object specifier appears in the event log.

- *REFRESH—Begins replication of journal entries for new, in-scope objects as each object is refreshed. Newly in-scope objects are refreshed one at a time. The group replication status must be *ACTIVE for this option to take effect.

MIRRCNTS

Indicates whether the contents of the object are mirrored. This parameter allows you to mirror object-level operations such as CREATE, DELETE, RENAME, and MOVE, but not the contents of the object.

You can only use this parameter with the following type and attribute combinations:

- OBJTYPE parameter is *DTAQ
- OBJTYPE parameter is *DTAARA
- OBJTYPE parameter is *FILE and OBJATTR parameter is PFDTA

The possible values are:

- *YES—Refreshes and mirrors the contents of the object. By default, this value is used.
- *NO—Does not refresh or mirror the contents of objects. Refresh creates an empty object on the backup node, and only the object-level operations such as CREATE, DELETE, RENAME, and MOVE are mirrored.

Note: *FILEATTR and *DTAARA sync checks will not report object contents mismatches if you decide not to refresh or mirror the object contents with MIRRCNTS(*NO).

PFRFSHMTD

Specifies the method of physical file refresh when a physical file is refreshed as part of a group refresh or object activation.

This parameter applies to the refresh of *FILE objects with an attribute of PF-DTA that match this object specifier when its INCFLG parameter is set to *INCLUDE.

The possible values are:

- *CLUSTER—Uses the cluster system value assigned in the DMSETSVAL command.
- *GROUP—Keeps the group setting for this parameter. By default, this value is used.
- *RBR—Record by record refresh.
- *SAVRST—Refresh by save/restore. Physical files are saved with the data and restored on the backup node. When the *SAVRST method is used to refresh files, the "Created by user" attribute of a file is retained on the backup node.

Note: The *SAVRST method of physical file refresh will not succeed if there are any other update or exclusive locks on the file being refreshed at the time of refresh, or if the file is under commitment control. A file's refresh by *SAVRST will also fail if the group's "Save

- !** "active" parameter evaluates to *NO and the file is open or locked for update by another application. In these cases, you should specify the *RBR refresh method.

PFUPDMTD

Specifies the physical data file update method. This parameter is applicable when *FILE object types with an attribute of PFDTA are selected and the **INCFLG** parameter is set to *INCLUDE.

You must assign an update method by relative record number or by unique key.

- Relative record number specifies the location of a record in relation to the beginning of a file.
- Unique key specifies a unique index that is used to update a file. It allows you to perform reorganizations of physical files at different times on the primary node and the backup node. This option requires that both before and after images are journaled on the backup node because the before images are required to remove the applied or keyed replication updates. Also, if you are updating multimember files, the members of the unique index must have the same name as the members of the physical file.

The possible values are:

- *RRN—Updates by relative record number. By default, this value is used.
- *UKEY—Updates by unique index. If you are updating by unique index, you then need to specify the name of the unique index (logical file) in the **PFKEY** parameter. If a unique index cannot be specified, then you must choose update by relative record number.

- !** **Note:** You cannot change the replication method in an object specifier for a *FILE object once selected.

PFKEY

Indicates a physical file unique key. An object name and library defines which file to use as the physical file unique key.

This parameter is available only when the **PFUPDMTD** parameter is set to *UKEY and the **INCFLG** parameter is set to *INCLUDE. It must be specified when the update method is by unique key. The object specifier must identify a unique index for the file.

The possible values are:

- <key>—Specifies the physical file unique key.
- *AUTO—Specifies to automatically determine a unique key for every physical file that is replicated by the object specifier.

- !** **Note:** If the unique key cannot be found on the backup node, the physical file is suspended on the backup node with the UKM suspension code.

You must identify the unique key and its library if you do not specify *AUTO. Prefix the physical file key with the name of the library where the key is located. For example:

LIB1/OBJ1

ACT

Settings for automatic reactivation of suspended objects that are replicated by an object specifier. The **Automatic reactivation** system value must be set to *YES. The following parameters combine to define the

automatic reactivation settings at the object specifier level.

Automatic reactivation—Specifies to try to reactivate suspended objects that are replicated by the object specifier. The possible values for **Automatic reactivation** are:

- *GROUP—Uses the group setting. By default, this value is used.
- *NO—Does not attempt to reactivate suspended objects.

Max. reactivation attempts specifies the number of automatic retries before ending reactivation attempts. The possible values for **Max. reactivation attempts** are:

- *CLUSTER—Uses the system value that is assigned in the DMSETVAL (Set Cluster System Values) command.
- *GROUP—Uses the group setting. By default, this value is used.
- *NOMAX—Specifies no maximum reactivation attempts.
- <number>—Specifies the maximum number of retries. Valid entries range between 1–32767.

Max. reactivation size specifies the maximum size of an object to include in the reactivation. Large object size impacts network performance and can introduce mirroring latency. The possible values for **Max. reactivation size** are:

- *CLUSTER—Uses the system value that is assigned in the DMSETVAL (Set Cluster System Values) command.
- *GROUP—Uses the group setting. By default, this value is used.
- *NOMAX—Includes objects of all sizes in the reactivation. This value might cause performance issues on the primary node if very large objects are frequently locked by other jobs.
- <size>—Specifies the maximum size (in bytes) of an object.

MAXRLE

Specifies the maximum number of record level errors (RLE) that are allowed before suspending a file on the backup node. Record level errors can occur during the apply process.

The possible values are:

- *CLUSTER—Uses the system value that is assigned in the DMSETVAL (Set Cluster System Values) command.
- *GROUP—Uses the group setting. By default, this value is used.
- *NOMAX—Specifies to continue replication of physical files for any number of record-level errors.
- *FIX—Specifies to continue replication of physical files when record-level errors occur during the backup apply process. File repair at activation is supported for files that are suspended with RLE. For more information about this value, refer to the ***FIXREC** option on the DMACTOBJ (Activate Object) command.
- <number>—Specifies the maximum number of record level errors.

OUTQOPTS

The following ***OUTQ replication options** parameters define the ***OUTQ** object replication and apply only to the output queues (*OUTQ objects) that are in the scope of the object specifier:

AUTOSTRWTR option—Specifies the value of a refreshed output queue's AUTOSTRWTR option on the backup node. This parameter is ignored when **Replicate *OUTQ contents** parameter evaluates to *NO. The possible

values for **AUTOSTRWTR** option are:

- *PRIMARY—Uses the AUTOSTRWTR option value for a refreshed output queue that is set on the primary node for the backup node. By default, this value is used.
- *NONE—Sets the AUTOSTRWTR option for a refreshed output queue on the backup node to *NONE. Ignores the AUTOSTRWTR option that is set on the primary node.

Replicate *OUTQ contents—The possible values are:

- *CLUSTER—Uses the system value that is assigned in the DMSETSVAL—Set Cluster System Values command. By default, this value is used.
- *NO—Replicates only the *OUTQ objects in the scope of this object specifier, but does not replicate spooled files. The **Replicate *OUTQ contents** system value must be set to *YES.

EXCLMBR

Specifies one or more file members to exclude from replication. Applies only to physical file objects (object type *FILE) that can have more than one member.

The possible values are:

- *NONE—No members of the file are excluded. By default, this value is used.
- <name>—The name of the member to exclude.
- <generic*>—A generic member name to identify multiple members of a file to exclude from replication.

HLDJOBSCD

Specifies whether to hold all of the job schedule entries for the replicated *JOBSCD objects on the backup node.

The possible values are:

- *NO—Do not hold the job schedule entries on the backup node. The entries have the same status on the backup node as the primary node. By default, this value is used.
- *YES—Hold all of the job schedule entries for the replicated *JOBSCD objects on the backup node.

Example 1

```
DMSELOBJ GROUP(GRP1) OBJ(LIB1/OBJ1) OBJTYPE(*JOBBD)
DESC('Job Description OBJ1 in LIB1') INCFLG(*INCLUDE) NEWOBJACT(*CURRENT)
```

Selects the object **OBJ1** in library **LIB1** that has an object type of *JOBBD (job description).

Provides a description to be associated with the object selection.

The object specifier is replicated in the group **GRP1**.

Replication of journal entries for new, in-scope objects begins at the time the object specifier is added, provided that the group is currently active.

Example 2

```
DMSELOBJ GROUP(GRP2) OBJ(LIB2/OBJ*) OBJTYPE(*USRSPC) TGTLIB(*GROUP)
INCFLG(*EXCLUDE)
```

Selects the individual objects in library **LIB2** that have names which start with **OBJ** (for example, **OBJ2**, **OBJTEST**, and so on) and an object type of *USRSPC (user spaces).

The objects referenced by the specifier are selected to the group **GRP2**.

The library is the same library that is as specified in the DMADDGRP (Add Group) or DMCHGGRP (Change Group) commands.

The referenced objects are not replicated within the group when cluster operations are started.

Example 3

```
DMSELOBJ GROUP(GRP3) OBJ(LIB3/OBJ3) OBJTYPE(*FILE) OBJATTR(PFDTA) TGTLIB(*PRIMARY)  
INCFLG(*INCLUDE) PFUPDMTD(*UKEY) PFKEY(LIB3/KEY3)
```

Selects the object **OBJ3** in the library **LIB3** that has an object type of *FILE and attribute of PFDTA.

The object referenced by the specifier is replicated in the group **GRP3** when cluster operations are started.

The **Target library** is set to *PRIMARY so that it is the same as the primary node library where the object resides.

The physical file is updated by unique key using KEY3 in library **LIB3**.

Restrictions

You must issue this command on an active node in the cluster.

Minimum Authority Level

*ADMIN

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **13**

Work With Object Specifier screen - **F6**

DMDSELOBJ (Deselect Objects from Group command)

Deselects an object specifier from a replication group. Deselecting an object specifier from a group stops referenced objects from being replicated by the group.

```
DMDSELOBJ GROUP( ) OBJ( ) OBJTYPE( ) OBJATTR( )
```

Input Parameters

GROUP

The name of the defined replication group that will have an object specifier deselected from it.

OBJ

The object name component of the specifier to deselect.

The possible values are:

- *ALL—Specifies all objects in a library.
- <name>—Specifies an object name.
- <generic*>—Specifies a generic object name to identify multiple objects in a library.

You must identify the library where the objects reside. Prefix the object specification with the name of the library where the objects are located. For example:

LIB1/OBJ1

A specifier that has a generic library must be deselected with the same generic library specifier. The possible values are:

- *ALL—Specifies all libraries. This value can only be used with *EXCLUDE specifiers for a specific object type.

OBJTYPE

The object type component of the specifier to select.

The possible values are:

- <type>—Specifies the object type. Press F4 for a list of all values.
- *ALL—Specifies all object types.

*ALL cannot be used for object types that can reside only in the QSYS library, that is, *LIB (library), *USRPRF (user profile) and the configuration object types *CNNL, *COSD, *CTLSD, *DEVD, *IPXD, *LIND, *MODD, *NTBD, *NWID, *NWSID.

OBJATTR

The attribute component of the specifier you want to select.

This parameter is only applicable when the OBJTYPE parameter is set to *FILE or *DEVD.

The possible values are:

- <type>—Specifies an attribute type. Press F4 for a list of all values.
- *ALL—Specifies all object attributes. *ALL is not a valid system attribute but allows iCluster to gather all objects regardless of their attribute. By default, this value is used.

Example 1

```
DMDSELOBJ GROUP(GRP1) OBJ(LIB1/OBJ1) OBJTYPE(*MSGF)
```

Deselects the object specifier **OBJ1** in library **LIB1** that has an object type of *MSGF (message file).

The object specifier is deselected from the group **GRP1**.

Example 2

```
DMDSELOBJ GROUP(GRP2) OBJ(LIB2/OBJ*) OBJTYPE(*SRVPGM)
```

Deselects the object specifier in library **LIB2** for objects that have names which start with **OBJ** (for example, **OBJ2**, **OBJTEST**, and so on) and an object type of *SRVPGM (service program).

The individual objects that are referenced by the specifier will no longer be replicated.

Example 3

```
DMDSELOBJ GROUP(GRP3) OBJ(LIB3/OBJ3) OBJTYPE(*FILE) OBJATTR(DSPF)
```

Deselects the object **OBJ3**, which resides in library **LIB3**, from the group **GRP3**.

The object specifier is a physical file object of the IBM i standard type DSPF.

Restrictions

You must issue this command on an active node in the cluster. The group must be inactive when you issue this command.

Minimum Authority Level

*ADMIN

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option 14

Work With Object Specifiers screen - Option 4

DMCHGOBJSL (Change Object Selection command)

Changes specific attributes of an object specifier that is selected to a replication group.

```
DMCHGOBJSL GROUP( ) OBJ( ) OBJTYPE( ) OBJATTR( ) TGTLIB( ) DESC( ) INCFLG( )
```

```
POLLINT( ) MIRRCNTS( ) PFRFSHMTD( ) PFUPDMTD( ) PFKEY( ) ACT( ) MAXRLE( ) OUTQOPTS( )
```

```
EXCLMBR( ) HLDJOBSCD( )
```

Through this command, you can change the attributes that are associated with the object specifier and modify the flag that determines whether the referenced objects are replicated within the group (**INCFLG**).

Input Parameters

GROUP

The name of the defined replication group that is affected by the change to the object specifier.

OBJ

The object name component of the specifier to change.

You cannot change the object name component through this command. It must be identified in order to change the other parameters that can be modified by this command.

The possible values are:

- *ALL—Specifies all objects in a library.

- <name>—Specifies the object name.
- <generic*>—Specifies a generic object name, to identify multiple objects in a library.

The library where the objects reside must be identified. Prefix the object specifier with the name of the library where the objects are located. For example:

LIB1/OBJ1

The possible values are:

- *ALL—Specifies all libraries. This value can only be used with *EXCLUDE specifiers for a specific object type.

OBJTYPE

The object type component of the specifier to select.

The possible values are:

- <type>—Specifies the object type. Press F4 for a list of all values.
- *ALL—Specifies all object types. *ALL cannot be used for objects that can reside only in the QSYS library, that is, *LIB (library), *USRPRF (user profile) and the configuration object types *CNNL, *COSD, *CTLD, *DEVD, *IPXD, *LIND, *MODD, *NTBD, *NWID, *NWSID.

OBJATTR

The attribute component of the specifier you want to select.

This parameter is only applicable when the **OBJTYPE** parameter is set to *FILE or *DEVD.

Note that you cannot change the object attribute component through this command. It must be identified in order to change the other parameters that can be modified by this command.

The possible values are:

- <name>—Specifies the attribute type. Press F4 for a list of all values.
- *ALL—Specifies all object attributes. *ALL is not a valid system attribute but allows iCluster to gather all objects regardless of their attribute. By default, this value is used.

TGTLIB

The name of the library on the backup node that will receive the replicated objects.

The possible values are:

- <name>—The library name.
- *SAME—Keeps the current setting for this parameter.
- *PRIMARY—Specifies that the library on the backup node is the same as the library where the object resides on the primary node.

If the primary and backup environments reside on the same physical system (local loopback implementation), the library that you specify cannot be the same library where a selected object in the group currently resides.

- *GROUP—Uses the same target library that is specified in the DMADDGRP (Add Group) or DMCHGGRP (Change Group) commands.

! **Note:** Switchovers and role changes are not supported for groups that have objects selected to them that have a value in the **Target library** field other than *GROUP or *PRIMARY. Also, because

- !** a non-primary library is required for local loopback replication, switchovers and role changes are not supported for local loopback replication.

DESC

The short description to use to identify this object specifier selection.

The possible values are:

- <description>—Specifies a short description of up to 50 characters long.
- *SAME—Keeps the current setting for this parameter.

INCFLG

Indicates whether to replicate the objects that are referenced by the specifier.

The possible values are:

- *SAME—Keeps the current setting for this parameter.
- *INCLUDE—Replicates the referenced objects within the group when cluster operations are started. The referenced objects are replicated to a backup environment when object replication is started.
- *EXCLUDE—Does not replicate the referenced objects within the group when cluster operations are started. Prevents the referenced objects from being replicated to a backup environment.

- !** **Note:** If you change this parameter from *EXCLUDE to *INCLUDE for an existing object specifier, you must issue the INITHAOBJ (Initialize Objects) command for the group.

POLLINT

The time interval that determines how often to check for content changes to user spaces and source physical files.

Set the interval by specifying the time between consecutive polls. The polling interval applies only when the object type is set to one of the pollable object types (*USRSPC and source physical file).

The possible values are:

- <time>—A time value from 000010 - 235959 expressed in HHMMSS format.
- *SAME—Keeps the current setting for this parameter.
- *GROUP—Uses the group value that is specified in the DMADDGRP (Add Group) or DMCHGGRP (Change Group) commands.
- *CLUSTER—Uses the system value that is assigned in the DMSETVAL (Set Cluster System Values) command.
- *NONE—Does not poll objects for content changes.

- !** **Note:** If *NONE is selected as the polling interval for an object specifier but the group's polling interval is not *NONE, polling will occur for all pollable objects that are selected to the group that do not match the object specifier whose polling interval is *NONE.

MIRRCNTS

Indicates whether the contents of the object are mirrored. This parameter allows you to mirror object-level operations such as CREATE, DELETE, RENAME, and MOVE, but not the contents of the object.

You can only use this parameter with the following type and attribute combinations:

- OBJTYPE parameter is *DTAQ

- OBJTYPE parameter is *DTAARA
- OBJTYPE parameter is *FILE and OBJATTR parameter is PFDTA

The possible values are:

- *SAME—Keeps the current setting for this parameter.
- *YES—Refreshes and mirrors the contents of the object.
- *NO—Does not refresh or mirror the contents of objects. Refresh creates an empty object on the backup node, and only the object-level operations such as CREATE, DELETE, RENAME, and MOVE are mirrored.

*FILEATTR and *DTAARA sync checks will not report object contents mismatches if you decide not to refresh or mirror the object contents with MIRRCNTS(*NO).

PFREFSHMTD

Specifies the method of physical file refresh when a physical file is refreshed as part of a group refresh or object activation.

This parameter applies to the refresh of *FILE objects with an attribute of PF-DTA that match this object specifier when its INCFLG parameter is set to *INCLUDE.

The possible values are:

- *SAME—Keeps the current setting for this parameter. By default, this value is used.
- *CLUSTER—Uses the cluster system value assigned in the DMSETSVAL command.
- *GROUP—Keeps the group setting for this parameter.
- *RBR—Record by record refresh.
- *SAVRST—Refresh by save/restore. Physical files are saved with the data and restored on the backup node. When the *SAVRST method is used to refresh files, the "Created by user" attribute of a file is retained on the backup node.

Note: The *SAVRST method of physical file refresh will not succeed if there are any other updates or exclusive locks on the file being refreshed at the time of refresh, or if the file is under commitment control. A file's refresh by *SAVRST will also fail if the group's "Save active" parameter evaluates to *NO and the file is open or locked for update by another application. In these cases, you should specify the *RBR refresh method.

PFUPDMTD

Note: This parameter is no longer used by iCluster. After being set, the **PFUPDMTD** in the DMSELOBJ (Select Objects to Group) command cannot be changed.

PFKEY

Indicates a physical file's unique key. An object name and library defines which file to use as the unique key for the physical file.

This parameter is available only when the **PFUPDMTD** parameter is set to *UKEY and the **INCFLG** parameter is set to *INCLUDE. It must be specified when the update method is by unique key. The object specifier must identify a unique index for the file.

The possible values are:

- <key>—Specifies the unique key and its library.
- *SAME—Keeps the current setting for this parameter.

- *AUTO—Specifies that you want iCluster to automatically determine a unique key for every physical file being replicated by the object specifier.

- Note:** If the unique key cannot be found on the backup node, the physical file is suspended with the UKM suspension code.

You must identify the unique key and its library if you do not specify *AUTO. Prefix the physical file key with the name of the library where the key is located. For example:

LIB1/OBJ1

ACT

Settings for automatic reactivation of suspended objects that are replicated by an object specifier. The **Automatic reactivation** system value must be set to *YES. The following parameters combine to define the automatic reactivation settings at the object specifier level.

Automatic reactivation—Specifies to try to reactivate suspended objects that are replicated by the object specifier. The possible values for **Automatic reactivation** are:

- *SAME—Keeps the current setting for this parameter.
- *GROUP—Uses the group setting.
- *NO—Does not attempt to reactivate suspended objects.

Max. reactivation attempts specifies the number of automatic retries before ending reactivation attempts. The possible values for **Max. reactivation attempts** are:

- *SAME—Keeps the current setting for this parameter.
- *CLUSTER—Uses the system value that is assigned in the DMSETVAL (Set Cluster System Values) command.
- *GROUP—Uses the group setting.
- *NOMAX—Specifies no maximum reactivation attempts.
- <number>—Specifies the maximum number of retries. Valid entries range between 1-32767.

Max. reactivation size specifies the maximum size of an object to include in the reactivation. Large object sizes impact network performance and can introduce mirroring latency. The possible values for **Max. reactivation size** are:

- *SAME—Keeps the current setting for this parameter.
- *CLUSTER—Uses the system value that is assigned in the DMSETVAL (Set Cluster System Values) command.
- *GROUP—Uses the group setting.
- *NOMAX—Includes objects of all sizes in the reactivation. This value might cause performance issues on the primary node if very large objects are frequently locked by other jobs.
- <size>—Specifies the maximum size (in bytes) of an object.

Activate files in ACTOOS state specifies to reactivate files that are replicated by this group and are in out-of-sync (ACTOOS) state. The possible values for **Activate files in ACTOOS state** are:

- *SAME—Keeps the current setting for this parameter.
- *GROUP—Uses the group setting.
- *CLUSTER—Uses the system value that is assigned in the DMSETSVAL (Set Cluster System Values) command.
- *NO—Does not attempt to refresh records in error for files in ACTOOS state.
- *YES—Specifies to reactivate files that are replicated by this group and are in out-of-sync (ACTOOS) state. This value is available only if automatic reactivation is enabled for the group.

MAXRLE

Specifies the maximum number of record level errors (RLE) that are allowed before suspending a file on the backup node. Record level errors can occur during the apply process.

The possible values are:

- *SAME—Keeps the current setting for this parameter.
- *CLUSTER—Uses the system value that is assigned in the DMSETSVAL (Set Cluster System Values) command.
- *GROUP—Uses the group setting.
- *NOMAX—Specifies to continue replication of physical files for any number of record-level errors.
- *FIX—Specifies to continue replication of physical files when record-level errors occur during the backup apply process. File repair at activation is supported for files that are suspended with RLE. For more information about this value, refer to the ***FIXREC** option on the DMACTOBJ (Activate Object) command.
- <number>—Specifies the maximum number of record level errors.

OUTQOPTS

The following parameters define the *OUTQ object replication and apply only to the output queues (*OUTQ objects) that are in the scope of the object specifier:

AUTOSTRWTR option—Specifies the value of a refreshed output queue's AUTOSTRWTR option on the backup node. This parameter is ignored when **Replicate *OUTQ contents** parameter evaluates to *NO. The possible values for **AUTOSTRWTR option** are:

- *SAME—Keeps the current setting for this parameter.
- *PRIMARY—Uses the AUTOSTRWTR option for a refreshed output queue that is set on the primary node for the backup node.
- *NONE—Sets the AUTOSTRWTR option for a refreshed output queue on the backup node to *NONE. Ignores the AUTOSTRWTR option that is set on the primary node.

Replicate *OUTQ contents—The possible values are:

- *SAME—Keeps the current setting for this parameter.
- *CLUSTER—Uses the system value that is assigned in the DMSETSVAL (Set Cluster System Values) command.
- *NO—Replicates only the *OUTQ objects in the scope of this object specifier, but does not replicate spooled files. The **Replicate *OUTQ contents** system value must be set to *YES.

EXCLMBR

Specifies one or more file members to exclude from replication. Applies only to physical file objects (object type *FILE) that can have more than one member.

The possible values are:

- *SAME—Keeps the current setting for this parameter.
- *NONE—Specifies that no members of the file are excluded. By default, this value is used.

- <name>—Specifies name of the member to exclude.
- <generic*>—Specifies a generic member name to identify multiple members of a file to exclude from replication.

HLDJOBSCD

Specifies whether to hold all of the job schedule entries for the replicated *JOBSCD objects on the backup node.

The possible values are:

- *NO—Do not hold the job schedule entries on the backup node. The entries have the same status on the backup node as the primary node. By default, this value is used.
- *YES—Hold all of the job schedule entries for the replicated *JOBSCD objects on the backup node.

Example 1

```
DMCHGOBJSL GROUP(GRP1) OBJ(LIB1/OBJ1) OBJTYPE(*AUTL)  
DESC('Authorization list OBJ1 in LIB1')
```

Changes the description associated with the specifier for **OBJ1** in library **LIB1** that has an object type of ***AUTL** (authorization list) in group **GRP1**.

Example 2

```
DMCHGOBJSL GROUP(GRP2) OBJ(LIB2/OBJ*) OBJTYPE(*USRSPC) INCFLG(*EXCLUDE)
```

Changes the specifier for objects in library **LIB2** that have names starting with **OBJ** (for example, **OBJ2**, **OBJTEST**, and so on) and an object type of ***USRSPC** (user spaces) to group **GRP2**.

The referenced objects are not replicated within the group when cluster operations are started.

Example 3

```
DMCHGOBJSL GROUP(GRP3) OBJ(LIB3/OBJ3) OBJTYPE(*FILE) OBJATTR(PFDTA) INCFLG(*INCLUDE)  
PFRFSHMTD(*SAVRST)
```

Changes the specifier for object **OBJ3** in the library **LIB3** that has an object type of ***FILE** to group **GRP3**.

Object **OBJ3** is replicated within the group when cluster operations are started.

Refresh by save or restore.

Restrictions

You must issue this command on an active node in the cluster. The group must be inactive when you issue this command.

Minimum Authority Level

***ADMIN**

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **15**

Work With Object Specifiers screen - Option **2**

DMWRKOBJ (Work with Object Specifiers command)

Lists the object specifiers. Use the **BY** and **GROUP** parameters to filter the list.

```
DMWRKOBJ BY() GROUP()
```

Input Parameters

BY

The set of object specifiers that are listed by this command.

The possible values are:

- *NONE—Specifies all object specifiers. By default, this value is used.
- *GROUP—Specifies the object specifiers that have been selected to the group specified through the **GROUP** parameter.

GROUP

The name of an existing group. Object specifiers selected to the named group are listed.

This parameter applies only when the **BY** parameter is set to **GROUP*.

Example 1

```
DMWRKOBJ BY(*NONE)
```

Lists all object specifiers.

Example 2

```
DMWRKOBJ BY(*GROUP) GROUP(GRP1)
```

Lists all object specifiers selected to the group **GRP1**.

Restrictions

You must issue this command on an active node in the cluster.

Minimum Authority Level

*USER

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **16**

Work With Groups screen - Option **12**

DMSNDOBJ (Send Object Immediately command)

Immediately replicates one or more objects that are not part of a replication group to a backup node.

```
DMSNDOBJ SRCNODE( ) SRCIASP( ) TGTNODE( ) TGTIASP( ) SNDTYPE( ) OBJ( )
OBJTYPE( ) OBJATTR( ) TGTLIB( ) PATH( )
```

This command also allows you to send path objects to another node immediately, without having to select them in a group.

However, if replication is running and you have determined that some objects in replication scope are not found on the backup node, do not use the DMSNDOBJ (Send Object Immediately) command to send these objects to the backup node if they are journaled objects. By doing this you will prevent these objects from being properly handled by the replication process, that is, by setting up the metadata that iCluster requires for changes to these objects to be replicated and starting journaling of the objects if they are not journaled. The correct way to deal with this issue is to use the DMACTOBJ (Activate Object) command or the DMACTBSF (Activate BSF Object) command to activate and refresh the objects from the group's primary node to the backup node.

 **Note:** Journaled objects are physical database files, logical files, data areas, data queues, and journaled BSF (Byte Stream Files).

Input Parameters

SRCNODE

The name of the node where the source object is found.

The possible values are:

- <name>—Specifies the name of a node.
- *CURRENT—Specifies the current node. By default, this value is used.

SRCIASP

The name of the independent auxiliary storage pool (IASP) device on the source node from which you want to send the object(s).

Specify the name of an existing IASP device or the following value:

- *SYSBAS—Uses a system ASP rather than an independent ASP for storing the objects that you want to send. By default, this value is used.

TGTNODE

The name of the node to send the object.

TGTIASP

The name of the independent auxiliary storage pool (IASP) device on the target node to which you want to send the object(s).

Specify the name of an existing IASP device or the following value:

-
- *SYSBAS—Uses a system ASP rather than an independent ASP for storing the objects that you want to send. By default, this value is used.

SNDTYPE

The type of object to send.

The possible values are:

- *LIBRARY—Specifies a native IBM i object that resides in an IBM i library.
- *PATH—Specifies a path object.

OBJ

The object name and library component of the object to send.

The possible values are:

- <name>—Specifies an object name.
- <generic*>—Specifies a generic object name to identify multiple objects in a library.
- *ALL—Specifies all objects in a library.

The library where the objects reside must be identified. Prefix the object specification with the name of the library where the objects are located (for example, **L1B1/OBJ1**).

This parameter is used only when the **SNDTYPE** parameter is *LIBRARY.

OBJTYPE

The object type component of the object to send.

The possible values are:

- <type>—The object type. Press F4 to see a list of all values.
- *ALL—Specifies all object types. By default, this value is used.

This parameter is used only when the **SNDTYPE** parameter is *LIBRARY.

OBJATTR

The attribute component of the object you want to send.

This parameter is only applicable when the **OBJTYPE** parameter is set to *FILE or *DEVD.

This field is free-form. Consequently, you can enter any value you want to describe the object, as long as the value conforms to IBM i naming conventions. However, there are values that have special meaning to iCluster. They are PFSRC (source physical file), PFDTA (data physical file), and PF (any physical file). Other values listed are standard IBM i file subtypes. If PF is used, the object specifier will match either PFSRC or PFDTA files.

The possible values are:

- <type>—Attribute type. Press F4 for a list of values.
- *ALL—Specifies all object attributes. *ALL is not a valid IBM i object attribute but allows iCluster to gather all objects regardless of their attribute. By default, this value is used.

This parameter is used only when the **SNDTYPE** parameter is *LIBRARY.

TGTLIB

The name of the library on the backup node that will receive the replicated objects.

The possible values are:

- <name>—Library name.
- *SRC—Sets the backup node library so that it is identical to the primary node library where the object resides.

If the primary and backup environments reside on the same physical system (local loopback implementation), the library that you specify cannot be the same library where a selected object in the group currently resides. This restriction means that iCluster does not permit the special value *SRC in this situation.

This parameter is used only when the **SNDTYPE** parameter is *LIBRARY.

PATH

The path object specifier that identifies the location of the BSF objects. This parameter applies only when the **SNDTYPE** parameter is *PATH.

Path object specifiers consist of a sequence of directories that identify the location of the BSF objects in the IFS. The path that is defined is similar to the path specification under Microsoft Windows and UNIX operating systems.

Enclose the path in single quotation marks ('') and start with a forward slash (/), for example '/Dir3/Dir4/file'. The path can be 2 - 5000 characters.

Generic path names of the form '/mydir*' are supported, where the generic indicator '*' is the final character of the path name. When using generic path names, all subdirectories are included recursively.

Generic file name extensions after the asterisk (*) are supported, where the file name extension follows the period (.) and is 1 - 8 alphabetic characters.

For example, '/mydir/*.txt' and '/mydir/prod*.txt'. File name extension support does not apply to /QDLS pathnames.

Output

None.

Example 1

```
DMSNDOBJ SRCNODE(*CURRENT) TGTNODE(TGT1) SNDTYPE(*LIBRARY) OBJ(LIB1/OBJ*)
OBJTYPE(*FILE) OBJATTR(*ALL) TGTLIB(*SRC)
```

This command will cause all FILE objects in library LIB1 that match the generic specifier OBJ* to be refreshed from the current node to the node named TGT1, into the LIB1 library.

Example 2

```
DMSNDOBJ SRCNODE(SRC1) TGTNODE(TGT1) SNDTYPE(*PATH) PATH('~/home/mydir/ex*')
```

This command will cause all path objects that in the ~/home/mydir directory that match the generic specifier 'ex*' to be refreshed from the node named SRC1 to the node named TGT1.

Example 3

```
DMSNDOBJ SRCIASP(IASP1) TGTNODE(BACKUP) TGTIASP(IASP2) OBJ(PAYROLL/CUSTOMER)
```

```
OBJTYPE(*FILE)
```

This command will cause the **CUSTOMER** file in library **PAYROLL** in **IASP1** on the current node to be sent to the backup node into the same library name but in**IASP2**.

Restrictions

The nodes that are specified on the **SRCNODE** and **TGTNODE** parameters must be active in the cluster.

Minimum Authority Level

*OPERATOR

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **17**

Related information

[Supported object types for replication](#)

DMSELDIRE (Select Directory Entry command)

Selects directory entries for replication.

```
DMSELDIRE GROUP( ) USRID( ) TGTADDR( ) DESC( ) INCFLG( )
```

This command selects the directory entries specified by the USRID parameter to the replication group identified through the first parameter. The directory entries referenced by the selected specifier are replicated from the primary node to the backup node of the replication group. Through the use of generic specifiers, any number of directory entries can be selected to a replication group.

Directory entry specifiers and objects are listed in the iCluster native object panels (DMWRKOBJ, WRKHAOBJST, DMWRKOBJST, and others) with the object type *DIRE. This is not a true IBM i object type, but is simply used by iCluster to indicate directory entry specifiers and objects in the native object panels.

Input Parameters

GROUP

The name of the replication group that will have directory entries selected to it.

USRID

Specifies the user ID and address of the directory entry that you want to select.

The possible values are:

- *ALL—Specifies all directory entries with a specific address.
- *ANY—Specifies a directory entry with the special user identifier *ANY.
- <generic-user-identifier>—Specifies a generic user identifier, to identify multiple directory entries. A generic name is a character string that contains one or more characters followed by an asterisk (*).
- user-identifier—Specifies a single user identifier.

Address

Specifies the address of the directory entries being selected.

The possible values are:

- *ALL—Specifies all addresses.
- user-address—Specifies a single address.

TGTADDR

Specifies the user address, system name and network user ID that will be given to the replicated directory entries on the backup node.

- Note:** Directory entries that do not have a user profile associated with them are always replicated with the same address and system name as they have on the primary node. However, their Network user ID parameter on the backup node will be determined by the value selected for this parameter.

The possible values are:

- *PRIMARY—Specifies that the address and system name will be the same as on the primary node.
- *BACKUP—Specifies that the address and system name will be the system name of the backup node. If this option is selected, the "Network user ID" parameter for the directory entry will be the same as the user ID and address associated with the directory entry.
- backup-address—Specifies the address and system name that will be given to the replicated directory entries matching this specifier on the backup node.

DESC

Specifies a short description that allows you to identify this directory entry specifier selection. You can use up to 50 characters.

INCFLG

Indicates whether the directory entries referenced by the specifier will be replicated within the replication group.

Specifying *INCLUDE means that the referenced directory entries will be replicated to the backup node when replication is started. Specifying *EXCLUDE prevents the referenced directory entries from being replicated to the backup node.

The possible values are:

- *INCLUDE—Specifies that the referenced directory entries are to be replicated by the group when the group's cluster operations are started. By default, this value is used.
- *EXCLUDE—Specifies that the referenced directory entries will not be replicated by the group.

Example

```
DMSELDIRE GROUP(GRP3) USRID(SYSUSR* MYSYS) TGTADDR(*BACKUP)
DESC('Directory entries for SYSUSR*') INCFLG(*INCLUDE)
```

Selects directory entries referenced by the generic user ID **SYSUSR*** and address **MYSYS** to replication group **GRP3**.

Specifies that the address and system name will be the system name of the backup node.

Includes a short description to easily identify the directory entry specifier.

Specifies that the referenced directory entries are to be replicated by the group when the group's cluster operations are started.

Restrictions

You must issue this command on an active node in the cluster.

Minimum Authority Level

*ADMIN

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **210**

Work With Object Specifier screen - **F7**

DMDSELDIRE (Deselect Directory Entry command)

Deselects a directory entry from a replication group.

```
DMDSELDIRE GROUP( ) USRID( )
```

This command deselects a directory entry specifier from the replication group identified through the first parameter. Deselecting a directory entry from a replication group stops the replication of the referenced directory entries by the replication group.

Directory entry specifiers and objects are listed in the iCluster native object panels (DMWRKOBJ, WRKHAOBJST, DMWRKOBJST, and others) with the object type *DIRE. This is not a true IBM i object type, but is simply used by iCluster to indicate directory entry specifiers and objects in the native object panels.

The directory entry specifier being deselected must be currently selected to the specified replication group. This command must be invoked on an active node in the cluster. The specified replication group must be inactive when this command is invoked.

Input Parameters

GROUP

Specifies the name of the replication group from which the directory entry specifier will be deselected.

USRID

Specifies the user identifier of the specifier that you want to deselect.

The possible values are:

- *ALL—Specifies all directory entries with a specific address.
- *ANY—Specifies a directory entry with the special user identifier *ANY.
- <generic-user-identifier>—Specifies a generic user identifier, to identify multiple directory entries. A generic name is a character string that contains one or more characters followed by an asterisk (*).
- user-identifier—Specifies a single user identifier.

Address

Specifies the address of the directory entries being deselected.

The possible values are:

- *ALL—Specifies all addresses.
- user-address—a single address.

Example

```
DMDSELDIRE GROUP(GRP2) USRID(SYSUSR* MYSYS)
```

Specifies that the directory entries in group **GRP2** associated with user ID **SYSUSR* MYSYS** will be deselected.

Restrictions

This command must be invoked on an active node in the cluster. The specified replication group must be inactive when this command is invoked.

Minimum Authority Level

*ADMIN

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **212**

Work With Object Specifier screen - **Option 4**

DMSUSDIR (Suspend Directory Entry command)

Suspends a specified directory entry to prevent it from being replicated to a backup node.

```
DMSUSDIR GROUP( ) USRID( ) BACKUP( )
```

Journal entries referencing the directory entry that were generated before a directory entry was suspended will still be sent to the backup node and applied. To activate a directory entry that was suspended through this command, use the DMACTDIRE command.



Note: Any changes made to a directory entry while it is suspended will not be mirrored when it is activated unless the directory entry is fully refreshed.

This command must be invoked on an active node in the cluster.

Input Parameters

GROUP

Specifies the name of the replication group to which the directory entry is selected. This command prevents the specified directory entry from being replicated to a backup node in the specified replication group by suspending it. The directory entry will appear suspended with the SBU reason code.

USRID

Specifies the user ID and address of the directory entry that you want to suspend. Only a single user identifier

can be suspended with this command.

The possible values are:

- *ANY—Specifies a directory entry with the special user identifier *ANY.
- user-identifier—Specifies a single user identifier.

Address

Specifies the address of the directory entry being suspended.

BACKUP

Specifies the name of the backup node for which the objects will be suspended.

If the group is not an *HADR group, you can use the special value *ONLY or you can specify the name of the group's backup node. If the group is an *HADR group, you must name one of the group's backup nodes.

The possible values are:

- <name>—Specifies the name of the group's backup node.
- *ONLY—Specifies that the group's only backup node is the backup node for the object suspension. This value is not valid for *HADR groups.

Example

```
DMSUSDIRE GROUP(GRP1) USRID(MYDIRE MYSYS) BACKUP(*ONLY)
```

Specifies the replication group **GRP1** to which the directory entry being suspended by this specifier has been selected.

Specifies the user ID and address, **MYDIRE MYSYS**, of the directory entry specifier that is being suspended.

Specifies that the group's only backup node is the backup node for the object suspension.

Restrictions

You must issue this command on an active node in the cluster.

Minimum Authority Level

*ADMIN

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **215**

DMACTDIRE (Activate Directory Entry command)

Activates one or more specified directory entries that are currently suspended from replication.

```
DMACTDIRE GROUP( ) USRID( ) RFSH( ) BACKUP( )
```

This command activates one or more directory entries that are currently suspended from replication. It can also be used to refresh one or more directory entries through the iCluster replication engine. After activating a directory entry, mirroring of the specified directory entry will start if the replication group is active. If the replication group is

inactive, mirroring will start when the group is started through the DMSTRGRP command.

! **Note:** Any changes made to a directory entry while it is suspended will not be mirrored when it is activated through this command unless the object is fully refreshed. Specify *YES for the RFSH parameter to verify that the directory entry on the backup node is correctly synchronized with the directory entry on the primary node.

This command must be invoked on an active node in the cluster.

Input Parameters

GROUP

Specifies the name of the replication group to which the directory entry to be activated is selected.

USRID

Specifies the user ID and address of the directory entry that is being activated.

The possible values are:

- *ALL—Specifies all directory entries.
- *ANY—Specifies a directory entry with the special user identifier *ANY.
- <generic-user-identifier>—Specifies a generic user identifier, to identify multiple directory entries. A generic name is a character string that contains one or more characters followed by an asterisk (*).
- user-identifier—Specifies a single user identifier.

Address

Specifies the address of the directory entries being activated.

The possible values are:

- *ALL—Specifies all addresses.
- user-address—Specifies a single address.

RFSH

Indicates whether you want to refresh the directory entry to the backup node when the directory entry is activated.

The possible values are:

- *YES—Refreshes the directory entry on the backup node when the directory entry is activated. By default, this value is used.
- *NO—Does not refresh the directory entry on the backup node when the directory entry is activated. In this case, you are responsible for refreshing the directory entry and ensuring that it is synchronized at the time the activation is performed.

BACKUP

Specifies the name of the backup node for which the objects will be activated.

If the group is not an *HADR group, you can use the special value *ONLY or you can specify the name of the group's backup node. If the group is an *HADR group, you must name one of the group's backup nodes.

The possible values are:

- <name>—Specifies the name of the group's backup node.
- *ONLY—Specifies that the group's only backup node is the backup node for the object activation. This value is not valid for *HADR groups.

Example

```
DMACTDIRE GROUP(GRP1) USRID(MYDIRE MYSYS) RFSH(*YES) BACKUP(*ONLY)
```

Specifies the replication group **GRP1** to which the directory entry being activated by this specifier has been selected.

Specifies the user ID and address, **MYDIRE MYSYS**, of the directory entry specifier that is being activated.

Refreshes the directory entry to the backup node when the directory entry is activated.

Specifies that the group's only backup node is the backup node for the object activation.

Restrictions

You must issue this command on an active node in the cluster.

Minimum Authority Level

*ADMIN

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **214**

SQL object commands

Use the SQL object commands to add, change, remove, display, and work with SQL objects.

The following are the SQL object commands with their usage:

- **DMSELSQLO (Select SQL Objects command)**: Use this command to select a SQL object specifier to the replication group identified through the GROUP parameter.
- **DMDSELSQLO (Deselect SQL Objects from group command)**: Use this command to deselect a SQL object specifier from the replication group identified through the GROUP parameter.
- **DMCHGSQLSL (Change SQL Object Selection command)**: Use this command to change the specific description of a SQL object specifier that is selected to a replication group.
- **DMWRKSQLO (Work with SQL Object Specifiers command)**: Use this command to list the SQL object specifiers. Use the **BY** and **GROUP** parameters to filter the list by displaying the SQL object specifiers that have been selected to a specific group.

Related information

[DMSELSQLO \(Select SQL Objects command\)](#)

[DMDSELSQLO \(Deselect SQL Objects from Group command\)](#)

[DMCHGSQLSL \(Change SQL Object Selection command\)](#)

[DMWRKSQLO \(Work with SQL Object Specifiers command\)](#)

DMSELSQLO (Select SQL Objects command)

Selects a SQL object specifier.

```
DMSELSQLO GROUP( ) OBJ( ) OBJTYPE( ) DESC( ) INCFLG( )
```

This command selects a SQL object specifier to the replication group identified through the GROUP parameter. The SQL objects referenced by the selected specifier are replicated from the primary node to the backup node of the replication group. Use generic specifiers to select any number of SQL objects supported by iCluster to a replication group.

SQL object specifiers are listed in the iCluster SQL object panel DMWRKSQLO.

Input Parameters

GROUP

The name of the replication group that will have SQL objects selected to it.

OBJ

The SQL object name component of the specifier to select. A SQL object name can be up to 128 characters long.

The possible values are:

- <name>—Specifies the SQL object name.
- <generic*>—Specifies a generic SQL object name to identify multiple SQL objects in a library. A generic SQL name is a character string that contains one or more characters followed by an asterisk (*).
- *ALL—Specifies all SQL objects in a library.

The library where the SQL objects reside must be identified. Prefix the SQL object specification with the name of the library where the SQL objects are located (for example, LIB1/SQLOBJ1).

The possible values are:

- <name>—The library name.
- <generic*>—A generic library name to identify multiple libraries. For example, JDE* identifies all libraries that begin with JDE, such as JDEPROD, JDEDEV, and so on.
- *ALL—Specifies all libraries.

OBJTYPE

The SQL object type component of the specifier to select or *ALL.

DESC

A short description that identifies this SQL object specifier selection. You can use up to 50 characters.

INCFLG

Indicates whether to replicate the SQL objects that are referenced by the specifier.

Specifying *INCLUDE means that the referenced SQL objects will be replicated to the backup node when replication is started. Specifying *EXCLUDE prevents the referenced SQL objects from being replicated to the backup node.

The possible values are:

- *INCLUDE—Replicates referenced SQL objects within the group when the group is started. By default, this value is used.
- *EXCLUDE—Does not replicate referenced SQL objects within the group.

Example

```
DMSELSQLO GROUP(GRP1) OBJ(LIB1/SQLPROC1) OBJTYPE(*SQPPROC)
DESC('SQL Procedure1') INCFLG(*INCLUDE)
```

Selects the object **SQLPROC1** in library **LIB1** that has an object type of ***SQLPROC** (SQL procedure).

Provides a description '**SQL procedure 1**' to be associated with the SQL object selection.

The SQL object specifier is replicated in the group **GRP1**.

Restrictions

You must issue this command on an active node in the cluster. The group must be inactive when you issue this command.

A SQL object cannot be suspended, and there is no command (including DMSUSOBJ) to suspend a SQL object.

The latency of the QSQJRN scraper DMSQJS does not appear in the monitors.

Minimum Authority Level

***ADMIN**

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **23**

Work With SQL Specifier screen - **F6**

Related information

[**Replicating SQL objects**](#)

DMDSELSQLO (Deselect SQL Objects from Group command)

Deselects a SQL object specifier from a replication group.

```
DMDSELSQLO GROUP( ) OBJ( ) OBJTYPE( )
```

This command deselects a SQL object specifier from the replication group identified through the first parameter. Deselecting a SQL object specifier from a group stops referenced SQL objects from being replicated by the group.

The SQL object specifier being deselected must be currently selected to the specified replication group.

Input Parameters

GROUP

The name of the replication group that will have a SQL object specifier deselected from it.

OBJ

The SQL object name component of the specifier to deselect. The name of a SQL object can be up to 128 characters long.

The possible values are:

- *ALL—Specifies all SQL objects in a library.
- <name>—Specifies a SQL object name.
- <generic*>—Specifies a generic SQL object name to identify multiple objects in a library. A generic SQL name is a character string that contains one or more characters followed by an asterisk (*).

The library where the SQL objects reside must be identified. Prefix the object specification with the name of the library where the objects are located. For example:

`LIB1/SQLOBJ1`

A specifier that has a generic library must be deselected with the same generic library specifier. The possible values are:

- <name>—The library name.
- <generic*>—A generic library name to identify multiple libraries. For example, JDE* identifies all libraries that begin with JDE, such as JDEPROD, JDEDEV, and so on.
- *ALL—Specifies all libraries. All non-system-included libraries that contain SQL objects are listed in the iCluster SQL object panel DMWRKSQLO.

Confirm if the specifiers meet your needs after running the command with the parameter Library set to *ALL.

If any specifiers are selected that you do not need, you must use the **DMCHGSQLSL INCFLG(*EXCLUDE)** to avoid the SQL objects that you do not need to be replicated. If you use the **DMDSELSQL0** command to delete the specifier, it may be reselected when the group starts.

Not all libraries are auto-selected. A library is not selected due to the following:

- Owned by the **DMCLUSTER** user profile.
- Owned by one of the **Aldon License Manager** user profiles and is not eligible for replication for Aldon LM high availability. (ACMSCTL and ACMSUSR libraries are eligible for replication).
- Licensed program product libraries and directories.
- iCluster work libraries.
- iCluster staging store libraries.
- Libraries and directories that are iCluster target libraries have **Library created by iCluster** as the descriptive text (which means that the library was created by iCluster as a target library for replication). Earlier forms of the text are the **Library created by iCluster for i** and the **Library created by IBM iCluster**.
- Contains only *JRN and *JRNRCV.
- System-owned library.

OBJTYPE

The SQL object type component of the specifier to deselect or *ALL.

Example

```
DMDSELSQL0 GROUP(GRP1) OBJ(LIB1/SQLOBJ1) OBJTYPE(*SQLFUNC)
```

Deselects the object specifier **SQLOBJ1** in library **LIB1** that has an object type of ***SQLFUNC** (SQL function).

The SQL object specifier is deselected from the group **GRP1**.

Restrictions

You must issue this command on an active node in the cluster. The group must be inactive when you issue this command.

Minimum Authority Level

*ADMIN

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **24**

Work With SQL Specifiers screen - Option **4**

Related information

[Replicating SQL objects](#)

DMCHGSQLSL (Change SQL Object Selection command)

Changes the specific description of a SQL object specifier that is selected to a replication group.

```
DMCHGSQLSL GROUP( ) OBJ( ) OBJTYPE( ) DESC( ) INCFLG( )
```

Through this command, you can change the description that are associated with the SQL object specifier (**DESC**) and modify the flag that determines whether the referenced objects are replicated within the group (**INCFLG**).

Input Parameters

GROUP

The name of the replication group that is affected by the change to the SQL object specifier.

OBJ

The object name component of the specifier to change. The name of the SQL object can be up to 128 characters long.

You cannot change the SQL object name component through this command. It must be identified in order to change the other parameters that can be modified by this command.

The possible values are:

- *ALL—Specifies all SQL objects in a library.
- <generic*>—Specifies a generic SQL object name, to identify multiple SQL objects in a library. A generic name is a character string that contains one or more characters followed by an asterisk (*).<name>—Specifies the SQL object name.

The library where the SQL objects reside must be identified. Prefix the SQL object specifier with the name of the library where the SQL objects are located. For example:

LIB1/SQLOBJ1

The possible values are:

- <name>—The library name.

-
- <generic*>—A generic library name to identify multiple libraries. For example, JDE* identifies all libraries that begin with JDE, such as JDEPROD, JDEDEV, and so on.
 - *ALL—Specifies all libraries. All non-system-included libraries that contain SQL objects are listed in the iCluster SQL object panel, **DMWRKSQL0**.

Confirm if the specifiers meet your needs after running the command with the parameter Library set to *ALL.

If any specifiers are selected that you do not need, you must use the **DMCHGSQLSL INCFLG(*EXCLUDE)** to avoid the SQL objects that you do not need to be replicated. If you use the **DMDSELSQL0** command to delete the specifier, it may be reselected when the group starts.

Not all libraries are auto-selected. A library is not selected due to the following:

- Owned by the **DMCLUSTER** user profile.
- Owned by one of the **Aldon License Manager** user profiles and is not eligible for replication for Aldon LM high availability. (ACMSCTL and ACMSUSR libraries are eligible for replication).
- Licensed program product libraries and directories.
- iCluster work libraries.
- iCluster staging store libraries.
- Libraries and directories that are iCluster target libraries have **Library created by iCluster** as the descriptive text (which means that the library was created by iCluster as a target library for replication). Earlier forms of the text are the **Library created by iCluster for i** and the **Library created by IBM iCluster**.
- Contains only *JRN and *JRNRVC.
- System-owned library.

OBJTYPE

The SQL object type component of the specifier to deselect or *ALL.

You cannot change the SQL object type component through this command. It needs to be identified in order to change the other parameters that can be modified by this command.

DESC

The short description to use to identify this SQL object specifier selection.

The possible values are:

- <description>—Specifies a short description of up to 50 characters long.
- *SAME—Keeps the current setting for this parameter.

INCFLG

Indicates whether to replicate the SQL objects that are referenced by the specifier.

After changing this parameter, you must restart the group with a refresh, or with the "Use marked positions" option after a synchronization of objects on backup and DMMRKPOS.

The possible values are:

- *SAME—Keeps the current setting for this parameter.
- *INCLUDE—Replicates the referenced SQL objects within the group when the group is started.
- *EXCLUDE—Does not replicate the referenced SQL objects within the group.

Example 1

```
DMCHGSQLSL GROUP(GRP1) OBJ(LIB1/SQLOBJ1) OBJTYPE(*SQLPROC)
DESC('SQLOBJ1 in LIB1')
```

Changes the description associated with the specifier for **SQLOBJ1** in library **LIB1** that has an object type of ***SQLPROC** (SQL procedure) in group **GRP1**.

Example 2

```
DMCHGSQLSL GROUP(GRP2) OBJ(LIB2/SQLOBJ*) OBJTYPE(*SQLPROC) INCFLG(*EXCLUDE)
```

Changes the specifier for SQL objects in library **LIB2** that have names starting with **SQLOBJ** (for example, **SQLOBJ2**) and an object type of ***SQLPROC** (SQL procedure) in group **GRP2**.

The referenced SQL objects are not replicated within the group when the group is started.

Example 3

```
DMCHGSQLSL GROUP(GRP3) OBJ(LIB3/SQLOBJ3) OBJTYPE(*SQLPROC) INCFLG(*INCLUDE)
```

Changes the specifier for SQL object **SQLOBJ3** in the library **LIB3** that has an object type of ***SQLPROC** to group **GRP3**.

SQL object **SQLOBJ3** is replicated within the group when the group is started.

Restrictions

You must issue this command on an active node in the cluster. The group must be inactive when you issue this command.

Minimum Authority Level

***ADMIN**

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **25**

Work With SQL Specifiers screen - Option **2**

Related information

[Replicating SQL objects](#)

DMWRKSQLO (Work with SQL Object Specifiers command)

Lists the SQL object specifiers. Use the **BY** and **GROUP** parameters to filter the list by displaying the SQL object specifiers that have been selected to a specific group.

```
DMWRKSQLO BY() GROUP()
```

Input Parameters

BY

The set of SQL object specifiers that are listed by this command.

The possible values are:

- *NONE—Specifies all SQL object specifiers. By default, this value is used.
- *GROUP—Specifies the SQL object specifiers that have been selected to the group specified through the **GROUP** parameter.

GROUP

The name of an existing group. SQL object specifiers selected to the named group are listed.

This parameter applies only when the **BY** parameter is set to **GROUP*.

Example 1

```
DMWRKSQL0 BY(*NONE)
```

Lists all SQL object specifiers.

Example 2

```
DMWRKSQL0 BY(*GROUP) GROUP(GRP1)
```

Lists all SQL object specifiers selected to the group **GRP1**.

Restrictions

You must issue this command on an active node in the cluster.

Minimum Authority Level

*USER

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **26**

Work With Groups screen - Option **15**

Related information

[Replicating SQL objects](#)

Byte Stream File commands

Use the Byte Stream File (BSF) commands to add, remove, change, and work with BSF objects.

The following are the Byte Stream File commands with their usage:

- **DMADDBSF (Add Path Specifier to Group command)**: On an active node in the cluster, select one or more Byte Stream File (BSF) objects in the Integrated File System (IFS) to the replication group.
- **DMRMVBSF (Remove Path Specifier from Group command)**: Deselects a path specifier from a replication group.
- **DMCHGBSF (Change Path Specifier to Group command)**: Changes specific attributes of a path specifier that is selected to the specified group.
- **DMWRKBSF (Work with Path Object Specifiers command)**: Lists the path object specifiers that are defined in the cluster. Use the **BY** and **GROUP** parameters to filter the list.

Related information

- [**DMADDBSF \(Add Path Specifier to Group command\)**](#)
- [**DMRMVBSF \(Remove Path Specifier from Group command\)**](#)
- [**DMCHGBSF \(Change Path Specifier to Group command\)**](#)
- [**DMWRKBSF \(Work with Path Object Specifiers command\)**](#)

DMADDBSF (Add Path Specifier to Group command)

On an active node in the cluster, select one or more Byte Stream File (BSF) objects in the Integrated File System (IFS) to the replication group.

```
DMADDBSF GROUP( ) PATH( ) DESC( ) INCFLG( ) MIRDIRCNT( ) JOURNAL( )
POLLINT( ) NEWOBJACT( ) ACT( ) NJRNCNT( )
```

The referenced BSF objects are replicated from the primary node to the backup node in the group.

Input Parameters

GROUP

The name of the defined replication group that will have BSF objects selected to it. You must define the group before selecting BSF objects to the group.

PATH

The path that identifies the location of the BSF objects.

Enclose the path in single quotation marks ('') and start with a forward slash (/), for example '/Dir3/Dir4/file'. The path can be 2 - 5000 characters.

Generic path names of the form '/mydir*' are supported, where the generic indicator '*' is the final character of the path name. When using generic path names, all subdirectories are included recursively.

Generic file name extensions after the asterisk (*) are supported, where the file name extension follows the period (.) and is 1 - 8 alphabetic characters.

For example, '/mydir/*.txt' and '/mydir/prod*.txt'. File name extension support does not apply to /QDLS pathnames.

DESC

A short description that allows you to identify this path object specifier selection. You can use up to 50 characters.

INCFLG

Indicates whether the BSF objects referenced by the path object specifier are replicated within the group.

The possible values are:

- *INCLUDE—Replicates referenced BSF objects within the group when cluster operations are started. By default, this value is used.
- *EXCLUDE—Does not replicate referenced BSF objects within the group when cluster operations are started.

! **Note:** BSF include/exclude specifiers cannot have symbolic links to directories as part of their paths. This is only permitted when the symbolic link is at the end of a nongeneric specifier that points directly to the symbolic link. iCluster does not support using symbolic links to directories to make BSF specifiers shorter.

MIRDRCNT

Indicates whether all BSF objects referenced by the path object specifier or only the directory tree will be replicated within the replication group.

The possible values are:

- *YES—Replicate all referenced BSF objects within the group when cluster operations are started. By default, this value is used.
- *NO—Replicate only referenced BSF directories within the group when cluster operations are started.

JOURNAL

Indicates how the BSF objects referenced by the path object specifier are replicated with the group.

The possible values are:

- *NONE—Does not journal BSF objects matching this object specifier. Only object-level changes to objects matching this specifier are replicated. Changes to the contents of the objects matching this specifier will not be replicated unless the entire object is replaced. By default, this value is used.
- *GROUP—Journals BSF objects matching this object specifier using the default BSF journal that is specified at the group level.

Notes:

- The overlap of journaled and non-journaled path object specifiers is restricted in iCluster. For example, using both `/home/user/*` (non-journaled) and `/home/user/employees/*` (journaled) is not permitted.
- Objects within the QDLS directory cannot be journaled. iCluster will enforce the *NONE option for the **JOURNAL** parameter of any path specifier that matches `/QDLS`.
- Content changes to non-journaled BSF objects can be replicated by setting the value for the **NJRNCNT** (Non-journaled BSF content changes) parameter to *YES. However, this type of replication of BSF content changes is slower than using journaling for BSF content changes.

POLLINT

Specifies whether to poll the Document Library objects (DLO) that match this path specifier for content changes.

The possible values are:

- *GROUP—Uses the group value that is specified in the DMADDGRP (Add Group) or DMCHGGRP (Change Group) commands. By default, this value is used.
- *NONE—Does not poll Document Library Objects (DLO) for content changes.

! **Note:** It is only possible to poll BSF objects in the `/QDLS` folder, that is, Document Library Objects (DLO).

NEWOBJACT

Indicates the method by which replication is to begin for the objects that come into replication scope when an object specifier is added.

The possible values are:

- *NONE—Does not change the replication status of new, in-scope objects. This value is intended to support initial group configuration. If this value is selected, replication must be started with a refresh of the entire group, or at a user-specified position with the DMSETPOS (Set Journal Start Position), DMMRKPOS (Mark Journal Positions), or DMREGPOS (Register Positions) commands. By default, this value is used.

 **Note:** This value is not allowed if replication of the group is active.

- *CURRENT—Begins replication of journal entries for new, in-scope objects at the time the object specifier is added. Journal entries related to new, in-scope objects created after that time are replicated. Journal entries related to new, in-scope objects created before that time will not be replicated.

 **Notes:**

- This value is only permitted when the group is active.
- If this value is selected, no changes should occur on the new, in-scope objects until the OMI0320 event for the object specifier appears in the event log.

- *REFRESH—Begins replication of journal entries for new, in-scope objects for each object as it is refreshed. New in-scope objects are refreshed one at a time.

 **Notes:**

- This value is only permitted when the group is active.
- If this option is selected, all the objects that match the specifier and do not match an exclude specifier for the group are refreshed. Objects that were already in replication scope, but also match the new object specifier, are refreshed as part of the DMADDBSF (Add Path Specifier to Group) command processing.

ACT

Settings for automatic reactivation of suspended objects that are replicated by an object specifier. The **Automatic reactivation** system value must be set to *YES. The following parameters combine to define the automatic reactivation settings at the object specifier level.

Automatic reactivation—Specifies to try to reactivate suspended objects that are replicated by the path specifier. The possible values for **Automatic reactivation** are:

- *CLUSTER—Uses the system value that is assigned in the DMSETSVAL (Set Cluster System Values) command.
- *GROUP—Uses the group setting. By default, this value is used.
- *NO—Does not attempt to reactivate suspended object that are replicated by this group for this path specifier.

Max. reactivation attempts specifies the number of automatic retries before ending reactivation attempts. The possible values for **Max. reactivation attempts** are:

- *CLUSTER—Uses the product system value.
- *GROUP—Uses the group setting. By default, this value is used.

- *NOMAX—Specifies no maximum reactivation attempts. If this value is specified, reactivation attempts never end.
- <number>—Specifies the maximum number of retries. Valid entries range between 1-32767.

Max. reactivation size specifies the maximum size of an object to include in the reactivation. Large object size can impact network performance and introduce mirroring latency. The possible values for **Max. reactivation size** are:

- *CLUSTER—Uses the product system value.
- *GROUP—Uses the group setting. By default, this value is used.
- *NOMAX—Includes objects of all sizes in the reactivation. This value might cause performance issues on the primary node if very large objects are frequently locked by other jobs.
- <size>—Specifies the maximum size (in bytes) of an object.

NJRNCONT

Indicates whether to replicate all of the content changes for the BSF objects that are not journaled to a database journal. The BSF objects that are not journaled to a database journal are selected with DMADDBSF JOURNAL(*NONE).

The possible values are:

- *CLUSTER—Uses the product system value. By default, this value is used.
- *NO—Does not replicate the content changes for nonjournaled BSF objects.
- *YES—Replicates the content changes for nonjournaled BSF objects. Objects are refreshed when the audit journal entries indicate that their content changes are scraped.

Note: Use only when you do not want to journal BSF objects on the primary system. Mirroring content changes of BSF objects using this method is slower than using the DMADDBSF JOURNAL(*GROUP) method. Objects are refreshed when the audit journal entries that indicate their content changes are scraped.

Example 1

```
DMADDBSF GROUP(GRP1) PATH('/home/myifs/docs/*') DESC('BSF Object')
INCFLG(*INCLUDE) MIRDIRCNT(*NO) JOURNAL(*NONE) NEWOBJACT(*CURRENT)
```

Selects the generic path name `/home/myifs/docs` to group **GRP1**. All path objects in the directory `/home/myifs/docs` are included for replication when cluster operations are started.

Provides a description to be associated with the object selection.

BSF objects matching this object specifier are not journaled.

Replicates only referenced BSF directories within the group when cluster operations are started.

Replication of audit journal entries begins for new, in-scope objects at the time the object specifier is added.

Example 2

```
DMADDBSF GROUP(GRP2) PATH('/home/myifs/docs/*') INCFLG(*INCLUDE) JOURNAL(*NONE)
POLLINT(*NONE) NEWOBJACT(*REFRESH)
```

Selects the generic path name `/home/myifs/docs/*` to group **GRP2**.

All path objects in the directory `/home/myifs/docs` are included for replication when cluster operations are started.

BSF objects that match this object specifier will not be journaled.

BSF objects will not be polled.

New in-scope objects are refreshed one at a time. Replication of audit journal entries for new, in-scope objects begins for each object as it is refreshed.

Example 3

```
DMADDBSF GROUP(GRP2) PATH('/home/myifs/docs/*') INCFLG(*INCLUDE) JOURNAL(*GROUP)
POLLINT(*NONE) NEWOBJACT(*REFRESH) NJRNCNT(*NO)
```

Selects the generic path name `/home/myifs/docs/*` to group **GRP2**.

All path objects in the directory `/home/myifs/docs` are included for replication when cluster operations are started.

BSF objects that match this object specifier are journaled to the group's default BSF journal.

Content changes for BSF files that match the path specifier are replicated.

BSF objects will not be polled.

New in-scope objects are refreshed one at a time. Replication of audit and database journal entries for new, in-scope objects begins for each object as it is refreshed.

Parameter **NJRNCNT(*NO)** does not take effect because **JOURNAL(*GROUP)** is specified

Restrictions

You must issue this command on an active node in the cluster.

Minimum Authority Level

*ADMIN

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **18**

Work With Path Object Specifiers screen - F6

DMRMVBSF (Remove Path Specifier from Group command)

Deselects a path specifier from a replication group.

```
DMRMVBSF GROUP( ) PATH( )
```

The path objects that are referenced by the specifier are no longer replicated in the replication group.

Input Parameters

GROUP

The name of the defined replication group from which a path specifier is deselected.

PATH

The path object specifier that identifies the location of BSF objects to deselect from the group. The path object specifier must be currently selected to the group specified through the **GROUP** parameter.

Enclose the path in single quotation marks ('') and start with a forward slash (/), for example '/Dir3/Dir4/file'. The path can be 2 - 5000 characters.

Generic path names of the form '/mydir*' are supported, where the generic indicator '*' is the final character of the path name. When using generic path names, all sub-directories are included recursively.

Generic file name extensions after the asterisk (*) are supported, where the file name extension follows the period(.) and is 1 - 8 alphabetic characters.

For example, '/mydir/*.txt' and '/mydir/prod*.txt'. File name extension support does not apply to /QDLS pathnames.

Example 1

```
DMRMVBSF GROUP(GRP1) PATH('/DIR1/DIR2/DIR3/FILEA')
```

Deselects the BSF object FILEA in /DIR1/DIR2/DIR3 from group GRP1.

Example 2

```
DMRMVBSF GROUP(GRP1) PATH('/DIR1/DIR2/DIR3/FILEA')
```

Deselects the BSF object FILEA in /DIR1/DIR2/DIR3 from group GRP1.

Example 3

```
DMRMVBSF GROUP(GRP2) PATH('/DIR1/DIR2/*')
```

Deselects the generic path name /DIR1/DIR2/* from group GRP2.

Restrictions

You must issue this command on an active node in the cluster. The group must be inactive when you issue this command.

Minimum Authority Level

*ADMIN

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **19**

Work With Path Object Specifiers screen - Option 4

DMCHGBSF (Change Path Specifier to Group command)

Changes specific attributes of a path specifier that is selected to the specified group.

```
DMCHGBSF GROUP( ) PATH( ) DESC( ) INCFLG( ) MIRDIRCNT( ) JOURNAL( )
POLLINT( ) ACT( ) NJRNCNT( )
```

You can change one or more parameters of a path specifier that is selected to a group.

Input Parameters

GROUP

The name of the defined replication group for which a path specifier is changed. The path that is referenced by the **PATH** parameter must be currently selected to the group.

PATH

The path object specifier that identifies the location of the BSF objects that are currently selected to the group that is identified through the **GROUP** parameter.

Enclose the path in single quotation marks ('') and start with a forward slash (/), for example '/Dir3/Dir4/file'. The path can be 2 - 5000 characters.

Generic path names of the form '/mydir*' are supported, where the generic indicator '*' is the final character of the path name. When using generic path names, all sub-directories are included recursively.

Generic file name extensions after the asterisk (*) are supported, where the file name extension follows the period(.) and is 1 - 8 alphabetic characters.

For example, '/mydir/*.txt' and '/mydir/prod*.txt'. File name extension support does not apply to /QDLS pathnames.

DESC

A short description that allows you to identify this path object specifier.

The possible values are:

- <description>—Specifies the description text. You can use up to 50 characters.
- *SAME—Keeps the current setting for this parameter.

INCFLG

Indicates whether the BSF objects referenced by the path object specifier are replicated within the group.

The possible values are:

- *SAME—Keeps the current setting for this parameter.
- *INCLUDE—Replicates referenced BSF objects within the group when cluster operations are started.
- *EXCLUDE—Does not replicate referenced BSF objects within the group when cluster operations are started.

MIRDIRCNT

Indicates whether all BSF objects referenced by the path object specifier or only the directory tree will be replicated within the replication group.

The possible values are:

- *YES—Replicate all referenced BSF objects within the group when cluster operations are started. By default, this value is used.
- *NO—Replicate only referenced BSF directories within the group when cluster operations are started.

JOURNAL

Indicates how the BSFBSF objects that are referenced by the path object specifier are replicated with the group.

The possible values are:

- *SAME—Keeps the current setting for this parameter.
- *GROUP—Journals BSF objects matching this object specifier using the default BSF journal that is specified at the group level.
- *NONE—Does not journal BSF objects matching this object specifier. Only object-level changes to objects that match this specifier are replicated. Changes to the contents of the objects matching this specifier are not replicated unless the entire object is replaced.

! Notes:

- The overlap of journaled and non-journaled path object specifiers is restricted in iCluster. For example, using both /home/user/* (non-journaled) and /home/user/employees/* (journaled) is not permitted.
- Content changes to non-journaled BSF objects can be replicated by setting the value for the **NJRNCNT** (Non-journaled BSF content changes) parameter to *YES. However, this type of replication of BSF content changes is slower than using journaling for BSF content changes.

POLLINT

Indicates the current polling interval for the Document Library Objects (DLO) that are referenced by the path object specifier. Using this parameter, you can indicate whether Document Library Objects (DLO) that match this object specifier are polled for content changes.

The possible values are:

- *SAME—Keeps the current setting for this parameter.
- *GROUP—Uses the group value that is specified in the DMADDGRP (Add Group) or DMCHGGRP (Change Group) commands.
- *NONE—Does not poll Document Library Objects (DLO) for content changes.

! Note:

It is possible to poll only BSF objects, for example, Document Library Objects (DLO), in the QDLS folder.

ACT

Settings for automatic reactivation of suspended objects that are replicated by an object specifier. The **Automatic reactivation** system value must be set to *YES. The following parameters combine to define the automatic reactivation settings at the object specifier level.

Automatic reactivation—Specifies to try to reactivate suspended objects that are replicated by the path specifier. The possible values for **Automatic reactivation** are:

- *SAME—Keeps the current setting for this parameter.
- *CLUSTER—Keeps the product system value for this parameter.
- *GROUP—Uses the group setting.

- *NO—Does not attempt to reactivate suspended objects.

Max. reactivation attempts specifies the number of automatic retries before ending reactivation attempts. The possible values for **Max. reactivation attempts** are:

- *SAME—Keeps the current setting for this parameter. This is the default value.
- *CLUSTER—Uses the product system value.
- *GROUP—Uses the group setting.
- *NOMAX—Specifies no maximum reactivation attempts. If this value is specified, reactivation attempts never end.
- <number>—Specifies the maximum number of retries. Valid entries range between 1–32767.

Max. reactivation size specifies the maximum size of an object to include in the reactivation. Large object size can impact network performance and introduce mirroring latency. The possible values for **Max. reactivation size** are:

- *SAME—Keeps the current setting for this parameter.
- *CLUSTER—Uses the product system value.
- *GROUP—Uses the group setting.
- *NOMAX—Includes objects of all sizes in the reactivation. This value might cause performance issues on the primary node if very large objects are frequently locked by other jobs.
- <size>—Specifies the maximum size (in bytes) of an object.

NJRNCONT

Indicates to replicate all of the content changes for the BSF objects that are not journaled to a database journal. The BSF objects that are not journaled to a database journal are selected with DMADDBSF JOURNAL(*NONE).

The possible values are:

- *SAME—Keeps the current setting for this parameter.
- *CLUSTER—Uses the product system value.
- *NO—Does not replicate the content changes for non-journaled BSF objects.
- *YES—Replicates the content changes for non-journaled BSF objects. Objects are refreshed when the audit journal entries indicate that their content changes are scraped.

Note: Use only when you do not want to journal BSF objects on the primary system. Mirroring content changes of BSF objects using this method is slower than using the DMADDBSF JOURNAL(*GROUP) method.

Example 1

```
DMCHGBSF GROUP(GRP1) PATH('/DIR1/DIR2/DIR3/FILEA') DESC('Text File') INCFLG(*SAME)
JOURNAL(*SAME)
```

Changes the description associated with the selection of BSF object FILEA in /DIR1/DIR2/DIR3 to group **GRP1**.

The parameter setting that determines whether the BSF object is replicated within the group is not changed from its current value.

The **JOURNAL** parameter uses the settings that were set in the DMADDBSF (Add Path Specifier to Group) command.

Example 2

```
DMCHGBSF GROUP(GRP2) PATH('/DIR1/DIR2/*') DESC(*SAME) INCFLG(*INCLUDE)
JOURNAL(*NONE) POLLINT(*SAME)
```

Changes the **INCFLG** parameter setting associated with the selection of a generic path name **/DIR1/DIR2/*** to group **GRP2**. All objects in the directory **/DIR1/DIR2** are replicated when cluster operations are started.

The description associated with the BSF object selection is not changed.

BSF objects matching this object specifier will not be journaled by iCluster.

The polling interval that was set in the DMADDGRP (Add Group) command is used.

Restrictions

You must issue this command on an active node in the cluster. The specified replication group must be inactive.

Minimum Authority Level

*ADMIN

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **20**

Work With Path Object Specifiers screen - Option **2**

DMWRKBSF (Work with Path Object Specifiers command)

Lists the path object specifiers that are defined in the cluster. Use the **BY** and **GROUP** parameters to filter the list.

```
DMWRKBSF BY( ) GROUP( )
```

Input Parameters

BY

The set of path object specifiers that are listed by this command.

The possible values are:

- *NONE—Specifies all path object specifiers that are selected to all groups in the cluster. By default, this value is used.
- *GROUP—Specifies the path object specifiers that have been selected to the replication group indicated in the **GROUP** parameter.

GROUP

The name of an existing replication group. Path object specifiers selected to the named group are listed.

This parameter applies only when the **BY** parameter is set to ***GROUP**.

Example 1

```
DMWRKBSF BY(*NONE)
```

Lists all path object specifiers.

Example 2

```
DMWRKBSF BY(*GROUP) GROUP(GRP1)
```

Lists all path object specifiers selected to the group **GRP1**.

Restrictions

You must issue this command on an active node in the cluster.

Minimum Authority Level

*USER

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **21**

Work With Groups screen - Option **13**

External application commands

Use the external application commands to configure high availability and disaster recovery (HA/DR) for supported external applications.

Related information

[**CFGHAXAPP \(Configure HA for External Application command\)**](#)

[**RMVHAXAPP \(Remove HA for External Application command\)**](#)

CFGHAXAPP (Configure HA for External Application command)

Creates replication groups and object specifiers to enable high availability and disaster recovery (HA/DR) for supported external applications.

```
CFGHAXAPP APPTYPE( ) GROUP( ) PRIMARY( ) BACKUP( ) HLPGRPS( )
HLPPROD( )
```

This command automates the following tasks:

- Creates one or more replication groups, as required.
- For Rocket LegaSuite for Application Modernization, the command also creates a replication group for Integrated File System (IFS) objects. For the group name, the first 7 characters of the specified group name are concatenated with IFS (for example, if the **GROUP** parameter specifies SEAGULL for the replication group name,

-
- an IFS group named **SEAGULLIFS** is created.
- Creates object specifiers for the external application.

Input Parameters

APPTYPE

Specifies the external application type. This parameter is required.

The possible values are:

- *LMI—Configures Rocket Lifecycle Manager for IBM i.
- *MOD—Configures Rocket LegaSuite for Application Modernization.
- *HLP—Configures Help Systems Robot products.

GROUP

<name>—Specifies the name of the first iCluster replication group to create for the external application HA configuration. If the external application HA configuration requires more than one replication group, the other group names are derived from the name of the first replication group.

The GROUP parameter is not used when the value of the **APPTYPE** parameter is *HLP and the **HLPGRPS** parameter is *BYPROD. This parameter is required.

PRIMARY

Specifies the name of the primary iCluster node for the replication environment. This node is typically the production system.

The possible values are:

- <name>—The name of the primary node name.
- *CURRENT—Specifies the current node where the command is being run. By default, this value is used.

This parameter is required. To determine the name of the primary node for the replication group, run the DMWRKNODE (Work with Nodes) command on the primary node.

BACKUP

<name>—Specifies the name of the backup node for the replication group. This node is typically the disaster recovery or backup system or LPAR. This parameter is required.

To determine the name of the backup node, run the DMWRKNODE (Work with Nodes) command to get a list of nodes in the cluster.

HLPGRPS

Specifies how to configure the replication groups for Help Systems when APPTYPE(*HELP).

The possible values are:

- *ONEGRP—Configures a single replication group for all Help Systems productions. By default, this value is used.
- *BYPROD—Configures a replication group for each Help Systems product. The replication group names are the product names.

HLPPROD

Specifies which Help Systems products to configure when APPTYPE(*HELP).

The possible values are:

- *ALL—Configures all Help Systems products. By default, this value is used.

- ALERT—Configures the Help Systems Robot/Alert product.
- CONSOLE—Configures the Help Systems Robot/Console product.
- CORRAL—Configures the Help Systems Robot/Corral product.
- ENTERPRISE—Configures the Help Systems Robot/Enterprise product.
- ENTERPRONE—Configures the Help Systems Robot/Enterprone product.
- EZVIEW—Configures the Help Systems Robot Easy View product.
- REPLAY—Configures the Help Systems Robot/Replay product.
- REPORTS—Configures the Help Systems Robot/Reports product.
- SCHEDULE—Configures the Help Systems Robot/Schedule product.
- TRANSFORM—Configures the Help Systems Robot/Transform product.
- TRAPPER—Configures the Help Systems Robot/Trapper product.

Example 1: Lifecycle Manager

```
CFGHAXAPP APPTYPE(*LMI) GROUP(LMI1) PRIMARY(PROD) BACKUP(LMIBACK)
```

Run this command to configure a replication group named **LMI1** on the primary node and the backup node. The command enables replication from the primary node **PROD** to the backup node **LMIBACK**.

Example 2: LegaSuite

```
CFGHAXAPP APPTYPE(*MOD) GROUP(MOD1) PRIMARY(PROD) BACKUP(MODBACK)
```

Run this command to configure two replication groups on the primary node and the backup node:

- The replication group named **MOD1** replicates native IBM i objects.
- The replication group named **MOD1IFS** replicates IFS objects.

The command enables replication from the current node **PROD** that the command is being run on to the backup node named **MODBACK**.

Example 3: Help Systems Robot/Schedule

```
CFGHAXAPP APPTYPE(*HLP) GROUP(ROBOTGRP) BACKUP(RBTBKP) HLPGRPS(*ONEGRP)  
HLPPROD(SCHEDULE ENTERPRISE)
```

Run this command to configure a replication group named **ROBOTGRP** on the primary node and the backup node.

The current node becomes the primary node. The backup node is **RBTBKP**.

Replication is enabled in a single replication group for the Help Systems Robot/Schedule and Robot/Enterprise products.

Restrictions

Issue this command on an active node in the cluster.

Minimum Authority Level

*ADMIN

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **118**

RMVHAXAPP (Remove HA for External Application command)

Removes the groups that were created to provide a high availability or disaster recovery environment for a supported external application.

This command also removes artifacts that were created by iCluster to aid in the replication process for the external application.

```
RMVHAXAPP APPTYPE( ) BACKUP( )
```

Input Parameters

APPTYPE

Specifies the external application type. This parameter is required.

The possible values are:

- *LMI—Specifies to remove groups for Rocket Lifecycle Manager for IBM i.
- *MOD—Specifies to remove groups for Rocket LegaSuite for Application Modernization.
- *HLP—Configures Help Systems Robot products.

BACKUP

<name>—Specifies the name of the backup node for the replication group.

This parameter is required. To determine the name of the backup node, use Option 5 against the group for the external application in the **DMWRKGRP** panel.

Example 1: Lifecycle Manager

```
RMVHAXAPP APPTYPE(*LMI) BACKUP(LMIBACK)
```

Run this command to remove the replication group **LMI1** for the backup node **LMIBACK**.

Example 2: LegaSuite

```
RMVHAXAPP APPTYPE(*MOD) BACKUP(MODBACK)
```

Run this command to remove the Rocket LegaSuite replication groups on the primary node and the backup node.

Restrictions

Issue this command on an active node in the cluster.

Minimum Authority Level

*ADMIN

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **119**

Administration commands

Use the administration commands to set and view system values, system date and time, manage event logs, and perform other iCluster administration tasks.

Related information

- [**DMSETVAL \(Set Cluster System Values\) command**](#)
- [**DMCHGTIME \(Change System Date and Time command\)**](#)
- [**DMCLRLOG \(Clear Cluster Event Log command\)**](#)
- [**HAPNGTCP \(Ping Using TCP command\)**](#)
- [**STRHATCP \(Start TCP/IP Listener command\)**](#)
- [**WRKHAJOB \(Work with Active Cluster Jobs command\)**](#)
- [**DMDLTCLSTR \(Delete Cluster command\)**](#)
- [**DMSETMSTR \(Set Metadata Owner command\)**](#)

DMSETVAL (Set Cluster System Values) command

View and change cluster-wide values for the replication environment.

```
DMSETVAL OPER( ) ACT( ) OBJ( ) SPLF( ) PF( ) BSF( ) EVNTLOG( ) LATENCY( ) CLUSTER( ) PERFORM( )
FULLSCOPTS( ) SYNCHECK( ) CUSTJRNE( ) DMMONHA( )
```

Cluster system values control the different behaviors of the cluster. Some values are used when the corresponding parameters in other commands are set to *CLUSTER.

Unless otherwise stated, restart the groups for the changes to take effect.

Input Parameters

Operational values (OPER)

Specifies settings for operational parameters.

Default polling interval

The time interval determines how often iCluster should check for content changes to user spaces, source physical files, and Document Library Objects (DLO) when replicating.

The possible values are:

- <time>—A time value from 000010 - 235959 that is expressed in HHMMSS format.
- *SAME—Keeps the current setting for this parameter.
- *NONE—Specifies no polling for content changes to user spaces, source physical files, and Document Library Objects (DLO).

The default value is 000500 (5 minutes).

Communications timeout

The waiting period that has to expire before the product gives up waiting for a message on the communication connection.

During the waiting period that is specified through this parameter, attempts to establish communications between the nodes continue. If communications cannot be established during this time, a communications timeout is reported.

The possible values are:

- <time>—A time value from 000015 - 235959 that is expressed in *HHMMSS* format.
- *SAME—Keeps the current setting for this parameter.

The default value is 010000 (1 hour).

Object compression

Indicates whether to compress objects in save files that are generated by iCluster and replicated to backup nodes.

The possible values are:

- *SAME—Keeps the current setting for this parameter.
- *NO—Does not compress objects. By default, this value is used.
- *YES—Compresses all objects in save files before replication.
- *MEDIUM—Compress all objects in save files before replication. Medium software compression is used. Medium compression is usually slower than low compression but faster than high compression. The compressed data is usually smaller than if low compression is used and larger than if high compression is used.
- *HIGH—Compress all objects in save files before replication. High software compression is used. High compression is usually slower, and the compressed data is usually smaller than if low or medium compression is used.

Compressing objects consumes processor cycles, but it might lead to faster transfer times of the save files. If objects are compressed on the primary node, the objects are automatically decompressed when they are received on the backup node.

Save active objects

Indicates whether an object can be updated at the same time it is being saved for refresh to the backup node.

The possible values are:

- *SAME—Keeps the current setting for this parameter.
- *NO—Does not allow objects that are in use by another job to be saved by the product. Objects cannot be updated while they are being saved. By default, this value is used.
- *YES—Allows objects that are in use by another job to be saved by the product. Objects might not be in a consistent state in relationship to each other.
- *SYNC—Allows objects that are in use by another job to be saved and updated. All of the objects are saved in a consistent state in relationship to each other.

Save operation output

Indicates whether save and restore operations create a spooled file that identifies which objects were restored. This parameter applies only to refresh, since mirroring does not generate spooled files. Use this spooled file to indicate which objects need to be refreshed.

The possible values are:

-
- *SAME—Keeps the current setting for this parameter.
 - *ERROR—Generates a spooled file for restore operations that produce an error. All other save operations do not generate a spooled file.
 - *FULL—Generates a spooled file to identify restored and not restored objects.
 - *NONE—Specifies that a spooled file is not generated. By default, this value is used.

Default job description

The name of the default iCluster job description for replication jobs.

The possible values are:

- <name>—Specifies the job description name. The default value is CSJOB
- *SAME—Keeps the current setting for this parameter.

Library

The library where the job description resides must be identified if you do not specify *SAME. When *SAME is not specified, the value has two parts: name and library.

The possible values are:

- *PRODLIB—Specifies your iCluster installation library.
- *LIBL—Specifies the set of libraries in your library list (the libraries are searched in order for the first occurrence of the specified default job description).
- <name>—The name of the library where the job description resides. The default value is ICLUSTER.

Group job start delay (secs)

This delay is used only when multiple replication processes are started through a single command invocation. It is intended for working environments that consist of relatively small systems. Use this setting to distribute the start of replication processes over time to avoid high peaks in CPU usage that typically occur when processes are started without delay. The number of seconds of delay between the start of replication processes when cluster operations are initiated for groups.

The possible values are:

- <time>—Specifies the delay time in seconds (1 - 60). The default value is 2.
- *SAME—Keeps the current setting for this parameter.

Delay for object processing

Specifies the following:

- The delay in seconds to wait before processing the create, rename, move, or activate journal entries for journaled objects (database files, data areas, data queues) and IFS files.
- The delay that iCluster will wait before refreshing a savefile when an object is saved into the savefile (resulting in a T ZC journal entry for the savefile).
- The delay before refreshing a user space if this delay is required because of a T ZC journal entry for the user space.
- The delay in saving a spooled file when the audit journal scraper processes a T SF journal entry with the action code C (create).

This delay is only done on the primary node. The maximum delay is 1800 seconds (30 minutes).

The possible values are:

- *SAME—Keeps the current setting for this parameter.
- <time>—A time value expressed in seconds, from 0 - 1800. The default value is 0.

Number of save operation retries

Specifies the number of times to retry a save operation if the initial save operation is unsuccessful. These retries are performed only on the primary node. The wait time between each retry is approximately one second.

The possible values are:

- *SAME—Keeps the current setting for this parameter.
- <number>—Specifies a number of retries from 0 - 100 to retry a save operation that is initially unsuccessful. The default value is 3.

Number of locked object checks

Specifies the number of times to check whether an object is locked by another process on the backup node. Locked objects prevent replication operations from being performed. The wait time between each check is defined with the **Delay between lock checks** system value. The locked object checks end when the number of locked object checks is reached, or when the object is no longer locked.

The possible values are:

- *SAME—Keeps the current setting for this parameter.
- *NOMAX—Specifies to never stop checking. By default, this value is used.
- <number>—Specifies the number of times to check a locked object. The default value is 5 checks. There is no maximum number of checks.

Delay between lock checks

Specifies the delay in seconds between checks of a locked object. The delay is done only on the backup node.

The possible values are:

- *SAME—Keeps the current setting for this parameter.
- <time>—A time value expressed in seconds (1 - 60). The default value is 1.

Apply delay (min)

Specifies the delay in minutes between the time that a journal entry was deposited in the source journal and the time it is applied on the backup node. It is assumed that the UTC offsets and system times are the same on the primary and the backup nodes.

The possible values are:

- *CLUSTER—Uses the product system value. By default, this value is used.
- <time>—A time value expressed in minutes (0 - 3000).

Number of DDM reconnect attempts

Specifies the number of attempts to reopen failed remote DDM connections. Failed DDM connections prevent access to the iCluster metadata that is stored on a remote node. The wait time between each attempt is defined with the **Delay between DDM retries** system value. The reconnect attempts end when the number of DDM reconnect attempts is reached, or when the DDM connection is reopened.

The possible values are:

- *SAME—Keeps the current setting for this parameter.
- *NOMAX—Specifies to never stop attempting to restore DDM connections. By default, this value is used.
- <number>—Specifies the number of times to attempt DDM reconnections. The default value is 5 attempts. The minimum number of attempts is 0. The maximum number of attempts is 60.

Delay between DDM retries

Specifies the delay, in seconds, to wait between attempts to restore DDM connections.

The possible values are:

- *SAME—Keeps the current setting for this parameter.
- <time>—A time value expressed in seconds (1 - 60). The default value is 1.

Enable comms auto-recovery

Specifies whether or not to use the iCluster communications auto-recovery feature in the case of a brief communications failure between nodes. A brief communications failure is one that lasts less than the total timeout defined by the **LNKCHKFRQ**, **LNKCHKRTO**, and **LNKCHKTRY** parameters for the node that detects the failure.

The iCluster communications auto-recovery feature is not used in clusters that use IBM Cluster Services for i.

The possible values are:

- *SAME—Keeps the current setting for this parameter.
- *YES—Specifies that the communications auto-recovery feature will be used in case of a brief communications failure. By default, this value is used.
- *NO—Specifies that the communications auto-recovery feature will not be used when a brief communications failure occurs.

iCluster replication job CCSID values

Specify the coded character set identifier (CCSID) for replication jobs.

iCluster replication job CCSID

Specify the CCSID for replication jobs. To avoid replication errors, set the **JOBCCSID** parameter to the same value as the default CCSID as determined by the country or region identifier (QCNTRYID) operating system value.

- *SAME—Keeps the current setting for this parameter.
- <value>—A valid CCSID value between 1 - 65534.
- 65535—Specifies no iCluster character data conversion and uses the country or region identifier (QCNTRYID) operating system value. By default, this value is used.

Generate SNMP traps

Specifies whether or not SNMP traps are to be generated for the iCluster messages that are defined as alertable messages.

The alertable messages issued by iCluster are:

- HAD4024 - replication failure in HADSFPR program
- HAD4025 - replication failure in HADSFPR program (refresh-only group)
- HAD5007 - replication failure in HADTUP program
- HAD5008 - replication failure in HADDJS as source for remote scraper
- HAD5009 - replication failure in HADDJSR remote journal scraper
- HAD5010 - replication failure in HADDJSR program
- HAD5011 - replication failure in HADDJS program
- HAD8050 - start-up failure in HADDAS program
- HAD8055 - replication failure in HADDAS program

-
- HAD8058 - replication failure in HADSFP program
 - HAD8059 - replication failure in HADSFP program (refresh-only group)
 - HAS0256 - node cannot be reached
 - OMI0035 - replication failure in OMGROUPJOB program
 - OMI0308 - replication latency exceeds the threshold specified
 - OMI9016 - replication failure in OMRCV program
 - OMI9017 - replication failure in OMRCV program (refresh-only group)
 - OMI9018 - replication failure in OMTARGET program
 - OMI9019 - replication failure in OMTARGET program (refresh-only group)

The possible values are:

- *SAME—Keeps the current setting for this parameter.
- *NO—Specifies that SNMP traps will not be generated for the alertable messages issued by iCluster. By default, this value is used.
- *YES—Specifies that SNMP traps will be generated for the alertable messages issued by iCluster.

If you select *YES for the GENTRAPS parameter, and there is currently no active alert filter on your system, the alert filter in your system's network attributes will be changed to the DMALRTFTR filter in the ICLUSTER library when the iCluster node is started.

If you select *YES for the GENTRAPS parameter, and there is currently an active alert filter defined in your system's network attributes, the iCluster alert selection entries will be added to the currently active alert filter with sequence numbers in the range 9982-9999 with group ICLUSTER.

There will also be an alert action entry added to the active alert filter that specifies that SNMP traps are to be generated for iCluster alerts. These entries will be removed when the node is ended or the GENTRAPS system value is changed to *NO. If any of the sequence numbers 9982-9999 are not available in the active alert filter selection entries, the iCluster alert selection entries will not be added to the active alert filter. If you still wish to have SNMP traps generated for iCluster alerts, follow the method given in the Recovery section of message CSI1237 in the HAMSGF message file in the ICLUSTER library to add the iCluster alert selection entries to the active alert filter.

Start *RFSH groups—Specify whether groups of type *RFSH are started when the DMSTRGRP command is used with a non-specific GROUP parameter. For example, when the GROUP parameter is either *ALL or *PRIMNODE and can refer to more than one group.

The *CLUSTER option of the **STRRFSHGRP** parameter on the DMSTRGRP command uses the value that is set with this parameter of the DMSETSVAL command.

- *SAME—Keeps the current setting for this parameter.
- *YES—Specifies that *RFSH groups are started when the **GROUP** parameter on the DMSTRGRP command is either *ALL or *PRIMNODE. By default, this value is used.
- *NO—Specifies that *RFSH groups are not started when the **GROUP** parameter on the DMSTRGRP command is either *ALL or *PRIMNODE.

Allow roleswitch for *RFSH groups—Specifies whether or not roleswitch is allowed for groups of type *RFSH. Values include:

- *SAME—Keeps the current setting for this parameter.
- *YES—Specifies that roleswitch is allowed for *RFSH groups. Only the DMCHGROLE command supports roleswitch for *RFSH groups. Roleswitch is only allowed for groups whose backup node is different from their primary node, do not have target library redirection, and are inactive.
- *NO—Specifies that roleswitch is not allowed for *RFSH groups. By default, this value is used.

Automatic reactivation values (ACT)

Specifies settings for the automatic reactivation of suspended objects. The following parameters combine to define the automatic reactivation settings.

Automatic reactivation

Specifies whether to attempt reactivation of suspended objects.

The possible values are:

- *SAME—Keeps the current setting for this parameter.
- *YES—Attempts to reactivate suspended objects. By default, this value is used.
- *NO—Does not attempt to reactivate suspended objects.

Reactivation interval

The number of minutes between checks for suspended objects.

The possible values are:

- <time>—A time value expressed in minutes, from 1- 1440 (one day).
- *SAME—Keeps the current setting for this parameter.

The default value is 10.

Max. reactivation attempts

The number of automatic retries before halting activation of a suspended object.

The possible values are:

- *SAME—Keeps the current setting for this parameter.
- *NOMAX—Never gives up on automatic reactivation; no limit for automatic retries. By default, this value is used.
- <number>—Specifies the maximum number of retries. Valid entries range between 1 - 32767.

Max. reactivation size

The largest object size to be included in the reactivation. The size of an object can affect network performance and introduce mirroring latency.

The possible values are:

- *SAME—Keeps the current setting for this parameter.
- *NOMAX—Includes objects of all sizes in the reactivation. This value could have serious performance issues on the primary node if very large objects are locked frequently by other jobs. By default, this value is used.
- <size>—Specifies the maximum size (in bytes) of an object.

Activate files in ACTOOS state

Specifies to reactivate files that are in an out-of-sync (ACTOOS) state. Files that are in ACTOOS state are not suspended, but produced errors for individual records (record-level errors) when they were replicated to the backup node.

The possible values are:

- *SAME—Keeps the current setting for this parameter.
- *NO—Does not attempt to refresh records for files that are in an out-of-sync (ACTOOS) state. By default,

this value is used.

- *YES—Attempts to refresh records in error for files in an out-of-sync (ACTOOS) state.

Log reactivated objects

Specifies whether to log objects that were activated with refresh. The refresh on the primary system includes automatic reactivation, and activation by using the DMACTOBJ (Activate Object), DMACTBSF (Activate BSF Object), and DMACTSPLF—Activate Suspended Spooled Files commands. The log file is **DMAREACLOG** in the iCluster product library.

- *SAME—Keeps the current setting for this parameter.
- *YES—Specifies to log all the objects, including native objects, spooled files, and BSF files that were activated with refresh on the primary system of replication.
- *NO—Does not log the reactivated objects. By default, this value is used.

Object values (OBJ)

Specifies settings for object parameters.

Lock files on backup nodes

Indicates whether journaled files should be locked on the backup node when they are being updated. When locked, no other users can modify these files on the backup node.

The possible values are:

- *SAME—Keeps the current setting for this parameter.
- *NO—Specifies that files are not locked when a group is active. Users can modify these files on the backup node. By default, this value is used.
- *YES—Specifies that files cannot be modified on the backup node when a group is active.

Maximum refresh size

Indicates the size of the largest physical file that can be refreshed automatically over the network. The size of an object can affect network performance and introduce mirroring latency.

If a physical file is too large to be refreshed, it is marked as suspended and is not refreshed. You are responsible for ensuring that the physical file is refreshed to the backup node (copy by tape, for example) and activated for replication.

The possible values are:

- <size>—Specifies the maximum file size in kilobytes.
- *SAME—Keeps the current setting for this parameter.
- *NOMAX—Specifies no maximum refresh size. By default, this value is used.

Hold configuration object source on backup node

Specifies whether to automatically create configuration objects immediately after they are received on a backup node, or hold the commands for creating them in specific physical source files so that they can be created later with the CRTCFCGOBJ (Create Configuration Objects) command.

The possible values are:

- *SAME—Keeps the current setting for this parameter.
- *YES—Holds commands to create configuration objects in specific physical source files so that they can be created later. By default, this value is used.
- *NO—Automatically creates configuration objects as soon as they are received on the backup node.

If configuration objects that are being replicated exist on backup nodes, set this value to *YES to prevent iCluster from trying to create objects when they are in use.

Configuration object sync check options

Indicates whether a configuration objects should be reported as out of sync if the CL source member for creating the object exists on the backup node, even when the object itself does not exist on the backup node.

This parameter is used only when the value of **Hold config obj src on backup node** is set to *YES.

The possible values are:

- *RPTCFG—Configuration objects are reported as out of sync with the CFG reason code if the objects do not exist on the backup node but the CL source members for creating the objects exist on the backup node. By default, this value is used.
- *NORPTCFG—Configuration objects are not reported as out of sync if the CL source members for creating the objects exist on the backup node, even when the objects themselves do not exist on the backup node.

Replicated user profile status

The status to assign to user profiles that were replicated to a backup node.

The possible values are:

- *SAME—Keeps the current setting for this parameter.
- *DISABLED—Sets all replicated user profiles to a status of *DISABLED on the backup node. The user profile cannot be used on the system. By default, this value is used.
- *PRIMARY—Sets the user profile status to be identical to the corresponding user profile on the primary node.
- *NOCHG—Preserves the current status of each user profile. Sets the status of newly created user profiles on the backup node to *DISABLED.

User profile replication level

Indicates whether to replicate dependent user profiles when a main user profile is replicated.

Dependent user profiles include object owner profiles, group profiles, supplemental profiles, and profiles on an authorization list that is associated with the replicated profile.

If you replicate dependent user profiles, then the relationship between user profiles is preserved for an exact mirroring of user profiles. However, replicating all dependent user profiles might affect performance if the number of dependent user profiles is considerable.

The possible values are:

- *SAME—Keeps the current setting for this parameter.
- *FULL—Replicates all dependent profiles, including group and supplemental profiles, profile owners, and dependent profiles that are dependent on dependent profiles, and so on. This is the default value.
- *BASIC—Replicates only profiles that match the selected object specifier. During replication, iCluster assumes that all dependent profiles already exist on the backup node. Error messages are generated if the dependent profiles are not on the backup node, though replication continues.

Replicate user profiles for authorization lists

Indicates to replicate authorization lists with or without the user profiles.

An authorization list object consists of a list of user profiles. This parameter dictates whether authorization lists are replicated with or without these user profiles. If user profiles are replicated, it is important to recognize which user profiles are sent to the backup node, as you may want to restrict access to this node. However, if a user profile in an authorization list does not exist on the backup node, the replicated authorization list is not the same as the authorization list on the primary node if user profiles are not replicated.

The possible values are:

- *SAME—Keeps the current setting for this parameter.
- *NO—Replicates authorization lists without user profiles. By default, this value is used.
- *YES—Replicates authorization lists with user profiles.

Replicate Q* user profiles and authorization lists

Indicates to replicate user profiles and authorization lists that start with the letter Q.

Usually, all profiles and authorization lists that start with the letter Q are owned by the system (for example, QSECOFR) and should not be replicated from one node to another.

The possible values are:

- *SAME—Keeps the current setting for this parameter.
- *NO—Does not replicate user profiles and authorization lists starting with the letter Q. By default, this value is used.
- *YES—Replicates non-system-owned profiles and authorization lists that start with the letter Q. If you defined user profiles and authorization lists that start with the letter Q that are not owned by the system, setting this parameter to *YES replicates those profiles and authorization lists. Profiles and authorization lists that are owned by QSYS are not replicated even when this parameter is set to *YES.

UID/GID mismatch suspension—Indicates whether to suspend user profiles when the UID or GID values are not the same on the primary and backup nodes. Replication must be ended and restarted for changes to take effect.

- *SAME—Keeps the current setting for this parameter.
- *YES—Suspend user profiles when the UID or GID values are not the same on the primary node and backup node. Select this value when your applications require the same UID and GID values on the backup node and the primary node. The same UID and GID values ensure that switchover and failover are successful when this requirement exists. You can remove suspension of user profiles by first modifying the user profiles so that they are the same on the primary node and back node, and then using the DMACTOBJ (Activate Object) command or another activation option from the Full Cluster Status Monitor. All changes to user profiles that are suspended with the IDC suspension reason are replicated, even when the user profiles are suspended.
 - Set this system value to *NO and activate the profiles with the DMACTOBJ command or one of the activation options from the Full Cluster Status Monitor.
 - Modify all user profiles on the backup node to ensure no differences or conflicts occur with the UID and GID values of the profiles on the primary node. Then activate the suspended profiles with the DMACTOBJ command or one of the activation options from the Status Monitor.
- *NO—Do not suspend user profiles when the UID or GID values are not the same on the primary node and the backup node. By default, this value is used.

Spooled file values (SPLF)

Specifies settings for replicating spooled files.

Replicate *OUTQ contents

Indicates whether to mirror the contents of output queues.

The possible values are:

- *SAME—Keeps the current setting for this parameter.
- *YES—Mirrors spooled files that have contents to the backup node. By default, this value is used.

Note: If *SPLFDTA and *PRTDTA are not included in the QAUDLVL of the IBM i system value, iCluster adds them to the QAUDLVL when SPLF is equal to *YES.

- *NO—Does not mirror the spooled files. Replicates only the *OUTQ objects, but not the spooled files in the objects.

Physical file values (PF)

Specifies settings for replicating physical files.

Commitment control level

The level of commitment control to use when replicating *FILE objects.

Commitment control stages the database transactions so that the database transactions are assembled before being applied, or additionally, opened in a commitment control environment to ensure backup database consistency. Commitment control also ensures that the changes are applied in the correct order.

The possible values are:

- *SAME—Keeps the current setting for this parameter.
- *NONE—Does not perform commitment control staging during the update process on the backup node. By default, this value is used.
- *LEVEL1—Assembles all updates in a transaction before being applied on the backup node.
- *LEVEL2—Applies all updates in a commitment control environment to ensure backup database consistency. This option provides true commitment control.

If you select *LEVEL2, but a file on the backup node cannot be opened under commitment control, *LEVEL1 commitment control is used for that file.

Journal images

Indicates whether default journaling should include both before and after images.

Before images are required for commitment control *LEVEL1 and *LEVEL2 in the case of a rollback of transactions under commitment control. Also, if unique key update method is specified for a file, then the file must be journaled with *BOTH images.

If a file is replicated using relative record number, it may be journaled with after images only.

The possible values are:

- *SAME—Keeps the current setting for this parameter.
- *AFTER—Journaling occurs only for the after image. By default, this value is used.
- *BOTH—Journaling occurs for both before and after images.

Journal objects on backup

Indicates whether to replicate the physical files, data areas, data queues, and BSF files that are journaled on the backup node.

The possible values are:

- *SAME—Keeps the current setting for this parameter.
- *NO—Does not journal replicated physical files, data areas, data queues and BSF files on backup nodes. By default, this value is used.
- *YES—Journals replicated physical files, data areas, data queues and BSF files on backup nodes.

Note: For PFDTA files that are replicated with **MIRRCNTS(*NO)**, iCluster does not journal them on backup nodes even if the **JRNBACKUP** parameter is set to *YES.

Default database journal

The name of the database journal to use as the default database journal.

The journal is used for database files, data areas, and data queues that are to be mirrored, but are not yet journaled.

The possible values are:

- <name>—Specifies the name of the database journal.
- *SAME—Keeps the current setting for this parameter.
- HADJRN—Uses the default journal that is supplied with iCluster.

Identify the library where the journal resides when *SAME is not specified. The journal has two parts, you must specify the name and the library. The possible values are:

- *PRODLIB—Specifies the iCluster installation library.
- *LIBL—Specifies the set of libraries in the library list (the libraries are searched in order for the first occurrence of the specified default job description).
- <name>—The name of the library where the database journal resides.

The default value is:

*PRODLIB/HADJRN

Delete dependent non-group logical files

Indicates whether to delete logical files that are dependent on physical files on the backup node, but are not replicated in the same group as the associated physical files, when the physical files are refreshed.

- Note:** If some of the logical files are not in the same replication group as the physical files, you must set this system parameter to *YES to ensure that the dependent logical files and the associated physical files are deleted when the physical files are refreshed.

The possible values are:

- *SAME—Keeps the current setting for this parameter.
- *NO—Specifies not to delete the logical files on the backup node when the associated physical files are deleted. Physical files that have dependent logical files can be refreshed only if the dependent logical files are deleted first on the backup node. By default, this value is used.
- *YES—Specifies to delete the logical files on the backup node when the associated physical files are deleted. Use this value when the associated physical file is not in the same replication group as the logical file.

- Note:** All logical files are deleted on the backup node when their associated physical files are refreshed. If the dependent logical files that are deleted are not in the same replication group as their associated physical files, the dependent logical files are not recreated on the backup node. To ensure that the out-of-scope dependent logical files are in sync on the backup node, you must manually refresh the logical files. To refresh the logical files, run a sync check for the group that replicates the dependent logical files, and then run the DMACTOOS command to activate the out-of-sync logical files.

Maximum record-level errors

The maximum number of record-level errors that are allowed to occur for a particular file before the file becomes suspended.

Record-level errors occur when updating, inserting, or deleting records in a file during replication to a backup node.

The possible values are:

- <number>—Specifies the maximum number of record-level errors. The default value is 1.
- *SAME—Keeps the current setting for this parameter.
- *NOMAX—Specifies that replication of physical files continues, regardless of the number of record-level errors that are generated.
- *FIX—Specifies to continue replication of physical files when record-level errors occur during the backup apply process and to record the relative record number (RRN) for each failed record to enable reactivation later. Files that become out-of-sync with record-level errors are marked as being in ACTOOS state and are listed as both active and suspended with a reason code of RLE.

For more information about specifying record-level errors, see the ***FIXREC** option for the **RFSH** parameter of the [DMACTOBJ—Activate Object](#) command.

Keep SQL trigger program attributes

Specifies whether the attributes of the SQL trigger programs of a physical file on the backup node are kept in sync with the primary node. Currently the attributes include ownership, private authorities, and text descriptions.

Maintaining the attributes of SQL trigger programs on the backup node in sync with the primary node increases the time needed for a roleswitch because each SQL trigger program needs to be checked and set up with the correct attributes on the new primary node. If you choose *YES for the parameter, the SQL trigger programs will be recreated on the new primary node at roleswitch, and the attributes of the SQL trigger programs will be updated to be the same as the attributes on the trigger programs on the previous primary node.

If you are not concerned with having the attributes of SQL trigger programs on the backup node in sync with the attributes of the SQL trigger programs on the primary node, or if you know that these attributes will not be changed, choose value *NO for the parameter. At roleswitch, the existing trigger programs for SQL triggers will be deleted on the new primary node. If you choose *NO for the parameter, the SQL trigger programs will be recreated automatically by DB2 on the new primary node when their associated physical file is opened, but they may not have the same attributes as the trigger programs on the previous primary node.

This parameter is only applicable to SQL trigger programs. Attributes of the trigger programs for native triggers on the backup node are kept in sync with the primary node, regardless of the value of the parameter.

The possible values are:

- *SAME—Keeps the current setting for this parameter.
- *YES—Specifies that the attributes of the SQL trigger programs of a physical file on the backup node will be kept in sync with the primary node.
- *NO—Specifies that the attributes of the SQL trigger programs of a physical file on the backup node will not be kept in sync with the primary node. By default, this value is used.

Physical file refresh method

Specifies the method of physical file refresh when a physical file is refreshed as part of a group refresh.

The possible values are:

- *SAME—Keeps the current setting for this parameter.
- *RBR—Record by record refresh. By default, this value is used.
- *SAVRST—Refresh by save/restore. Physical files are saved with the data and restored on the backup node. When the *SAVRST method is used to refresh files, the "Created by user" attribute of a file is retained on the backup node.



Note: The *SAVRST method of physical file refresh will not succeed if there are any other update or exclusive locks on the file being refreshed at the time of refresh, or if the file is under commitment control. A file's refresh by *SAVRST will also fail if the group's "Save

- !** "active" parameter evaluates to *NO and the file is open or locked for update by another application. In these cases, you should specify the *RBR refresh method.

Byte stream file values (BSF)

Specifies settings for replicating Byte Stream File (BSF) objects.

Default BSF journal

The name of the journal to use as the default for the journaled BSF objects.

The possible values are:

- <name>—Specifies the journal name.
- *SAME—Keeps the current setting for this parameter.
- HABSFJRN—Uses the default BSF journal that is supplied with iCluster.

Library

The possible values are:

- *PRODLIB—Specifies the iCluster installation library.
- *LIBL—Specifies the set of libraries in the library list. The libraries are searched in order for the first occurrence of the specified default BSF journal.
- <name>—Specifies the library name.

The default value is

*PRODLIB/HABSFJRN

Mirror non-default BSF attributes—Specifies to replicate changes to the following non-default attributes of BSF: *READONLY, *HIDDEN, *PCSYSTEM, *PCARCHIVE, *CCSID, *ALWCKPWRT, *DISKSTGOPT, *MAINSTGOPT, *CRTOBJSCAN, *SCAN, *ALWSAV, *RSTDNRNMUNL, *SETUID, and *SETGID.

- *SAME—Keeps the current setting for this parameter.
- *YES—Replicates the changes to the non-default attributes of BSF. Replication performance is impacted, especially when BSF are journaled on the backup node. By default, this value is used.
- *NO—Does not replicate the changes to the non-default attributes of BSF.

Nonjournaled BSF content changes—Specifies to replicate content changes for BSF objects that are not journaled to a database journal (BSF objects that are selected with the DMADDBSF JOURNAL(*NONE) command).

- *SAME—Keeps the current setting for this parameter.
- *YES—Replicates the content changes to BSF objects that are not journaled to a database journal. Objects are fully refreshed to the backup node when the audit journal entries that indicate content changes are scraped. Replication of BSF objects is slower than using the DMADDBSF JOURNAL(*GROUP) command and should be used only when you do not want to journal BSF objects to a database journal on the primary node.
- *NO—Does not replicate the content changes to BSF objects that are not journaled to a database journal. By default, this value is used.

Keep list of active BSF—Specifies to keep a list of all journaled BSF objects that are being replicated and enables these objects to be displayed on the status monitors for the group. The list is stored in one of the following locations:

- Local journaling groups: in the HABSPEND file on the group primary node.

- Remote journaling groups: in the HARBSPEND file on the group backup node.

The list of active journaled BSF objects is updated during normal replication when the database journal scraper encounters a journal entry for an object that is within replication scope.

- *SAME—Keeps the current setting for this parameter.
- *YES—Specifies to keep a fully populated list of active BSF objects. To populate the list for groups that exist, you must restart the groups with one of the following methods:
 - Refresh
 - With marked journal positions (after you run **DMMRKPOS**)
 - With set journal positions (after you run **DMSETPOS**)
- *NO—Does not keep the list of active BSF objects. By default, this value is used.

Event log values (EVNTLOG)

Specifies settings for the cluster event log.

Default event message queue

The name of the message queue to hold event messages that are generated.

If a message queue is identified, event messages are placed in the message queue and the event log.

The possible values are:

- *SAME—Keeps the current setting for this parameter.
- <name>—Specifies the name of the event message queue.
- *NONE—Specifies that no message queues are used. Event messages are placed in the event log only. By default, this value is used.
- HAMSGQ—Specifies the event message queue that is supplied with iCluster.

Library—The library where the event message queue resides must be identified if you do not specify *NONE or *SAME. The parameter has two parts, and you must specify both the name and the library. The following values are possible:

- *PRODLIB—Specifies the iCluster installation library.
- *LIBL—Specifies the set of libraries in your library list. The libraries are searched in order for the first occurrence of the specified default event message queue.
- <name>—The name of the library where the message queue resides.

Remote date, time log display

Indicates which date and time settings should be used to display event messages that originate from a remote node. This value is relevant when local and remote nodes are located in different time zones.

The possible values are:

- *SAME—Keeps the current setting for this parameter.
- *LOCAL—Specifies the date and time in the time zone where the local node is located. By default, this value is used.
- *REMOTE—Specifies the date and time in the time zone where the remote node is located.

Logged message lifetime (days)

The maximum length of time that messages in the event log should be kept before they are removed. If the age of a specific message exceeds the time limit, this message is automatically removed when the user issues the **DMDSPLG** (Display Cluster Event Log) command. The default value is 30.

The possible values are:

- <time>—A time value expressed in days, from 1 - 365.
- *SAME—Keeps the current setting for this parameter.

-
- *NOMAX—Disables automatic removal of event messages. Messages stay in the log until you remove them.

Message generation levels

Specifies the severity levels of messages that are generated in the event log.

By default, only messages that have a severity level of 20, 30 or 40 are generated. When specific message levels are identified, messages in other levels are not generated. However, fatal error messages (Level 40) are always generated.

You can specify to generate more than one message level by using *ALL or by separating each message level in the parameter with a space. For example, MSGLVLS (10 20) specifies that only status and warning messages should be generated. The default value is (20 30).

The possible values are:

- <value>—Specifies a message level.
- *SAME—Keeps the current setting for this parameter.
- 00—Generates information messages (Level 00) (for example, Survival created...)
- 10—Generates non-critical status messages (Level 10)
- 20—Generates stop/start messages and warnings (Level 20) (for example, Mirroring started...)
- 30—Generates messages that report recoverable errors (Level 30)
- *ALL—Generates messages in all levels

Maximum event log records

Specifies the maximum number of records that can be added to the event log.

When the number of records in the event log reaches the specified maximum, the records in the log will be removed up to the last 100 records that were added to the log. Setting this parameter to *NOMAX means that a record will not be cleared from the event log until it has been in the event log longer than the number of days specified on the "Logged message lifetime" parameter. The default value is *NOMAX.

The possible values are:

- *SAME—Keeps the current setting for this parameter.
- *NOMAX—Specifies that there is no maximum number of records in the event log. Records will not be removed from the log until they have been in the log for longer than the number of days specified on the "Logged message lifetime" parameter.
- Maximum-records—Specify the maximum number of records that can be in the event log. Must be a number between 10 000 (ten thousand) and 1000 000 000 (one billion).

Latency check values (LATENCY)

Latency is the time difference between what is on the primary node and what is on the backup node.

There are two types of latency in iCluster:

- Source receive latency: The time difference between the last journal entry in the primary (source) node and the last journal entry received and put into the staging store on the backup node.
- Backup apply latency: The time difference between the last journal entry that is put into the staging store and the last journal entry that is applied on the backup node.

Total latency is the sum of source receive latency and backup apply latency.

Latency is checked and a warning message is issued when the threshold setting is exceeded. Configure the threshold with the LATENCY system values.

! **Note:** A warning message is issued only after the latency threshold is exceeded for one minute.

Latency check interval (min)

Specifies how often to check for latencies (source receive and target apply) and compare them with their corresponding thresholds.

The possible values are:

- <time>—A time value expressed in minutes, from 1- 1440 (one day).
- *SAME—Keeps the current setting for this parameter.

The default value is 5.

Source receive threshold (min)

This threshold value specifies the amount of source receive latency that is tolerated before a latency warning message is issued. Source receive latency indicates the difference between the timestamp of the journal entry last processed by the backup receiver for the journal, and the timestamp of the last journal entry deposited in the journal on the primary (source) node.

For idle or lightly used journals, the latency is determined with an entry in a journal that is idle for one minute since the last journal entry.

The possible values are:

- <time>—A time value expressed in minutes, from 1- 10080 (seven days).
- *SAME—Keeps the current setting for this parameter.

The default value is 60.

Backup apply threshold (min)

This threshold value indicates the amount of apply latency that can be tolerated before a latency warning message is issued. The backup apply latency is the difference in the timestamps of the last entry that is received by the journal receive process and the last entry that was applied by the journal apply process.

The possible values are:

- <time>—A time value expressed in minutes, from 1- 10080 (seven days).
- *SAME—Keeps the current setting for this parameter.

The default value is 240.

Continuous latency messages

Specifies whether a single or multiple OMI0308 messages are sent to the event log when the latency threshold is newly exceeded for a group-journal process.

The possible values are:

- *YES—Allows OMI0308 messages to continue to be sent once every hour for this group-journal process as long as the latency threshold is exceeded for the group-journal process.
- *NO—Specifies that when the latency threshold is newly exceeded for a group-journal process, a single OMI0308 message is sent to the event log.

The default value is *NO.

Use IBM Cluster Services for i (CLUSTER)

This setting specifies the failover mechanism that the cluster uses to detect node failures.

The possible values are:

- *SAME—Keeps the current setting for this parameter.
- *NO—Uses SwitchOver System as the failover mechanism. By default, this value is used.
- *YES—Uses Cluster Resource Services as the failover mechanism.

Performance values (PERFORM)

Specifies the cluster system values that affect the performance of iCluster in certain scenarios.

Comms channel

This setting allows you to specify whether a separate communications channel is used for each group or one channel for all groups.

The possible values are:

- *SAME—Keeps the current setting for this parameter.
- *SHARED—Uses a single communications channel for all groups. By default, this value is used.
- *PERGROUP—Uses a single communications channel for each group. With separate communications channels for each group, better line utilization can be obtained and problems in one group does not affect other groups. However, this requires more communications resources.

Staging store block management

This setting allows you to control how iCluster manages used staging store blocks. iCluster normally reallocates blocks to the memory pool after they are used, but now you can choose not to reallocate the staging store blocks for a performance improvement in situations where you are using multiple apply jobs with large data volumes. The possible values are:

- *SAME—Keeps the current setting for this parameter.
- *SHARED—Reallocates staging store blocks to the memory pool after they are used. This is the default behavior of iCluster. By default, this value is used.
- *PERGROUP—Does not reallocate staging store blocks to the memory pool after they are used. This can result in a performance improvement when you are using multiple apply jobs.

Maximum number of jobs for parallel processing—Set this parameter to the maximum number of jobs for parallel processing of iCluster resource-consuming requests, when applicable. The job that processes the request will submit a number of parallel processing jobs up to the specified maximum. The commands that can be processed in parallel are:

- DMMRKPOS
- DMSETPOS
- DMOBJCNT
- DMRPLCVRPT
- DMAUDITRPT
- DMSTRSC and STRHASC with SCTYPE(*CHECKSUM) or with SCTYPE(*FULL) RUNCHKSUM(*YES)
- DMSTRSCUSR and STRHASCUSR with SCTYPE(*CHECKSUM)

The possible values are:

- *SAME—Keeps the current setting for this parameter.
- maximum-number-of-jobs—Enter an integer value between 0 and 50. The value 0 means that all processing will be done by the job that is executing the request. This is the default value of the parameter.

Full sync check values (FULLSCOPTS)

Specifies the cluster system values for options that can be used with a *FULL sync check when it is run for a group.

These values can be used when running the commands DMSTRSC and STRHASC with the *FULL sync check type

and *CLUSTER as the value for the RUNCHKSUM or RUNCHKOBSL parameters on these commands. The values will also be used for continuous sync check, if the *FULL sync check type is specified for continuous sync check.

Check file contents

Specifies whether or not a *FULL sync check will also perform a file contents check.

The possible values are:

- *SAME—Keeps the current setting for this parameter.
- *YES—A file contents sync check is performed with *FULL sync check.
- *NO—A file contents sync check is not performed with *FULL sync check. By default, this value is used.

Check obsolete objects

Specifies whether or not a *FULL sync check will also perform a check of obsolete objects on the backup node.

The possible values are:

- *SAME—Keeps the current setting for this parameter.
- *YES—An obsolete objects check is performed with *FULL sync check.
- *NO—An obsolete objects check is not performed with *FULL sync check. By default, this value is used.

Continuous sync check values (SYNCCHECK)

Specifies the cluster system values that affect the performance of continuous sync checks by iCluster

Enable continuous sync check

Specifies that sync checks are continuously run.

The possible values are:

- *SAME—Keeps the current setting for this parameter.
- *YES—Enables continuous sync checks.
- *NO—Does not enable a continuous sync check. By default, this value is used.

Delay between objects (msec)

Specifies the delay time in milliseconds between objects.

This parameter is intended to minimize the interference from sync check processing on regular object replication. Sync check can be a CPU-intensive process and may cause unnecessary delays in the regular replication processing if there is no delay between the attribute checking of individual objects.

The possible values are:

- *SAME—Keeps the current setting for this parameter.
- <value>—Specifies a value between 0–10000. The default value is 200.

Sync check type

Specifies the type of continuous synchronization check to be performed. This parameter takes effect only if continuous sync check is enabled.

The possible values are:

- *SAME—Keeps the current setting for this parameter.
- *FILEATTR—Checks whether the *FILE objects on the primary node exist on the backup node. It also checks that the attributes of the primary node *FILE objects match the attributes of the backup node *FILE objects.
- *OBJATTR—Checks whether objects exist on the backup node. Also checks their authorities and journaling attributes (for journaled objects). BSF and native objects are checked.

-
- *FULL—Performs an *OBJATTR sync check on all objects that are replicated by the group and a *FILEATTR sync check on database file objects. Also performs a *CHKOBSLETE sync check to identify objects found on the backup node but not found on the primary node. By default, this value is used.

Auto-repair authority

Specifies whether to refresh the private authorities, ownership and primary group of objects, whose authorities (including ownership and primary group) as part of the sync check processing.

The possible values are:

- *NO—Does not refresh out-of-sync mismatched authorities. By default, this value is used.
- *YES—Refreshes out-of-sync mismatched authorities as part of the sync check processing.

Auto-repair journaling

Specifies whether to repair the journaled objects whose journaling is found to be mismatched by a *FULL, *FILEATTR, or *OBJATTR continuous sync check.

The possible values are:

- *NO—Does not repair out-of-sync mismatched journaling for journaled objects. By default, this value is used.
- *YES—Repairs out-of-sync mismatched journaling for journaled objects as part of the continuous sync check processing.

Auto-repair created by user

Specifies that database *FILE objects whose "Created by user" attribute is found to be mismatched by a *FULL or *OBJATTR sync check, are refreshed as part of the sync check processing. The possible values are:

- *NO—Does not refresh out-of-sync created by user for database *FILE objects.
- *YES—Refreshes out-of-sync created by user for database *FILE objects as part of continuous sync check.

Auto-repair creation date and time

Specifies that library (native) objects whose "Creation date and time" attribute is found to be mismatched by a *FULL or *OBJATTR sync check, are refreshed as part of the sync check processing. The possible values are:

- *NO—Does not repair mismatched creation date and time for library (native) objects.
- *YES—Repairs mismatched creation date and time for library (native) objects as part of continuous sync check.

Custom journal entries values (CUSTJRNE)

Specifies the custom journal entries to process.

Custom journal entries—Specifies the journal entries to process. The supported journal entry is journal code 'U' of entry type 'EH'.

- *SAME—Keeps the current setting for this parameter.
- UEH—Supports journal code U of entry type EH. Scrapes and applies the custom journal entry as an audit journal entry for Create Object (T CO). The **ENTDTA** parameter on the **SNDJRNE** command requires the following template:

```
'AbcdGhijklmnopQrstuvwxyz1234567'
```

where: A - d is *SYNG - p is the object name (padded with spaces)Q - z is the object library (padded with spaces)1 - 7 is object type (*USRSPC or *USRIDX only)

- *NONE—Does not process user-specified journal entries. By default, this value is used.

Replication monitoring values (DMMONHA)

Specifies options to start a job on the node that owns the iCluster metadata to automatically monitor group replication status, event log messages, and suspended objects.

Enable replication monitoring—Specifies to automatically enable replication monitoring.

- *SAME—Keeps the current setting for this parameter.*SAME—Keeps the current setting for this parameter.
- *NO—Do not enable automatic replication monitoring. By default, this value is used.
- *YES—Enable automatic replication monitoring. Start the **DMMONHA** job whenever the node that owns the iCluster metadata becomes active to monitor event log messages, suspended objects and group replication status and to clean up expired event log messages on all active nodes of the cluster.

Monitor event log—Specifies the level, or a set of levels, of severity of event log messages, or specific event log message IDs, that trigger an alert.

When event log messages are checked, an alert message is sent when a message that matches the message severity or message identifier list is placed in the event logs in the interval between the previous monitor check and the current monitor check.

- *SAME—Keeps the current setting for this parameter.
- *SEV30—Reports event log messages with a severity of 30 or higher. By default, this value is used.
- *SEV40—Reports event log messages with a severity of 40 or higher.
- *NONE—Reports no messages.
- *SELECT—Reports event log messages with a severity of 40 or higher, and also messages with the message identifiers (MSGID) in the list of message IDs. Select message IDs with the Event message IDs to monitor (**SELMSGID**) and Event message IDs to omit (**OMITMSGID**) parameters.

Monitor suspended objects—Specifies whether the monitoring job should check for suspended objects.

- *SAME—Keeps the current setting for this parameter.
- *YES—Check suspended objects and send an alert message when an object remains suspended for the time that is specified with the **Objects suspended for (min)** parameter. By default, this value is used.
- *NO—Do not check suspended objects.

Monitor group status—Specifies to check group replication status.

- *SAME—Keeps the current setting for this parameter.
- *YES—Check the status of group replication and send an alert message when the group status changes from *ACTIVE to *INACTIVE or *IN_DOUBT and stays inactive for the time that is specified with the **Group inactive or in doubt for (min)** parameter.
- *NO—Do not check group replication status. By default, this value is used.

Clean up event logs—Specifies to clear expired messages from the event logs for all active nodes in the cluster. The lifetime of event log messages is set with the **Logged Message Lifetime** parameter.

Note: The "Maximum event log records" value takes precedence over the "Clean up event logs" value. If the size of the event log on a node exceeds the maximum specified in the "Maximum event log records" value, the event log will be cleared regardless of the "Clean up event logs" value.

- *SAME—Keeps the current setting for this parameter.
- *YES—Remove expired messages from the event logs when the DMMONHA job performs a check.
- *NO—Do not remove expired messages from the event logs. By default, this value is used.

Check interval (min)—Specifies a number of minutes to define the interval between the checks that are done by the DMMONHA job.

- *SAME—Keeps the current setting for this parameter.
- <value>—Specifies a number of minutes.

Objects suspended for (min)—Specifies a number of minutes to define the minimum duration for an object to be suspended before the suspended object status is reported in an alert message. Applies only when **Monitor suspended options** is *YES.

- *SAME—Keeps the current setting for this parameter.

- <value>—Specifies a number of minutes.

Group inactive or in doubt for—Specifies a number of minutes to define the minimum duration for a group to be inactive before the inactive group status is reported in an alert message. Parameter applies only when **Monitor group status** is *YES.

- *SAME—Keeps the current setting for this parameter.
- <value>—Specifies a number of minutes.

Event message IDs to monitor—Specifies a list of message identifiers in the event log for monitoring by the DMMONHA job. Messages of severity 40 are always checked when event log checking is enabled.

- *SAME—Keeps the current setting for this parameter.
- *ALL—Check all messages of severity 30 or higher. By default, this value is used.
- *SUSPND—Check all messages that indicate an object is suspended.
- *LATNCY—Check all messages that indicate that a replication latency threshold is exceeded.
- <list>—Check messages of any severity in a list of up to 20 message identifiers.

Event message IDs to omit—Specifies a list of message identifiers in the event log to omit from monitoring by the DMMONHA job. Parameter applies only when **Monitor event log** is *SELECT.

- *SAME—Keeps the current setting for this parameter.
- *NONE—Omits no messages.
- <list>—Omits messages in a list of up to 20 message identifiers.

Destination type—Specifies which destinations to use for the alert messages that are generated by the DMMONHA job.

- *SAME—Keeps the current setting for this parameter. By default, this value is used.
- *ALL—Sends messages to all of the destinations that are defined with the DMADDALR command and are listed in the DMWRKALR panel, including email addresses and message queues.
- *EMAIL—Sends messages only to the email addresses that are listed in the DMWRKALR panel.
- *MSGQ—Sends messages only to the messages queues that are listed in the DMWRKALR panel.

Example

```
DMSETSVAL OPER(003000 000020 *NO *NO *FULL LIB1/DEFJD 5) ACT(*NO)
OBJ(*NO *NOMAX *NO *DISABLED *FULL *YES *YES) SPLF(*YES)
PF(*LEVEL2 *BOTH *YES *PRODLIB/HADJRN *NO 5) BSF(LIB1/BSFJRN)
EVNTLOG(LIB1/MSGQ1 *LOCAL 120 (00 20)) LATENCY(5 60 240) PERFORM(*SHARED *SHARED)
```

OPER

The default polling interval is 30 minutes.

The waiting period before a communications timeout is declared is 20 seconds.

Object compression is turned off.

Objects are not saved when they are in use.

A spooled file is generated during all save and restore operations to record what was saved and restored by iCluster.

The default job description is DEFJD in library LIB1.

A five-second delay between the start of replication processes will occur when cluster operations are initiated for groups.

ACT

Automatic reactivation will not be enabled for suspended objects.

OBJ

Objects are not be locked on backup nodes.

There is no maximum refresh size for physical files.

Configuration objects that are replicated to backup nodes are not held in files for creation later. iCluster will automatically create these objects immediately after they are received from a primary node.

All replicated user profiles on the backup node are set to a status of *DISABLED.

All dependent user profiles are replicated.

User profiles are replicated with authorization lists.

User profiles (except those owned by QSYS) starting with the letter Q are replicated.

SPLF

Spooled files are mirrored for the output queues that are selected for replication.

PF

*LEVEL2 commitment control is applied to replicated files.

Physical files that are replicated to backup nodes are journaled.

Physical files are journaled to the journal supplied with iCluster, HADJRN, which is located in the library where iCluster was installed.

Dependent logical files that are not selected to the same group for replication will not be deleted when refreshing a physical file. Physical files with these logical files cannot be refreshed.

After five record-level errors are produced, iCluster will suspend the file.

BSF

BSF objects are journaled to the journal **BSFJRN**, which is located in library **LIB1**.

EVNTLOG

In addition to the event log, iCluster messages will also be placed in the message queue MSGQ1 in library LIB1.

Dates and times are displayed in the time zone where the local node is located.

All messages that have been in the cluster event log for more than 120 days are automatically removed when the DMDSLOG—Display Cluster Event Log command is issued.

iCluster will only generate information (Level 00), warning (Level 20 and higher severity), and fatal error (Level 40) messages.

LATENCY

iCluster checks the latencies (source receive and target apply) and compares them to their respective thresholds every 5 minutes. The primary receive threshold is 60 minutes, and the backup apply threshold is 240 minutes.

PERFORM

One communications channel is used for all groups.

Staging store blocks are reallocated to the memory pool after they are used.

- !** **Note:** At least one space must separate each item in the **OPER**, **ACT**, **OBJ**, **SPLF**, **PF**, **BSF**, **EVNTLOG**, **LATENCY**, and **PERFORM** parameters. Follow the example to specify these parameters correctly.

Restrictions

You can issue this command on any active node in the cluster. You can also issue the command on an inactive node, as long as the cluster does not exist and that node is the first node that is defined in the cluster.

Minimum Authority Level

*ADMIN

Menu Access

Main Menu - Option 6

iCluster Commands menu - F22 (Shift + F10) - Option **30**

Related information

[Modifying default cluster system values](#)

[DMACTOBJ \(Activate Object command\)](#)

DMCHGTIME (Change System Date and Time command)

Changes the system clock date and time.

```
DMCHGTIME CHGTYPE( ) SYSVAL( ) VALUE( ) SECONDS( ) MINUTES( ) HOURS( )
```

Since journal entries are processed based on their timestamp, changing the system clock can affect the order in which journal entries are processed. The DMCHGTIME (Change System Date and Time) command helps to ensure that journal entries are applied in the correct sequence after the system time changes backwards.

You must use DMCHGTIME (Change System Date and Time) instead of your operating system's time adjustment commands if the time change is less than half an hour in the past. For example, you are changing the time from 2:00 AM to 1:45 AM.

If this is not the case, use your operating system's commands to change the system time. iCluster automatically detects and handles time changes in these situations and running this command is optional.

You can change only one value at a time. To change both the system date and time, you must issue this command twice.

If iCluster is currently scraping a large number of journals, then this command can take a long time to run. To reduce the processing time for this command, run DMCHGTIME (Change System Date and Time) when system activity is low.

To use this command, you must be signed on as QPGMR, QSYSOPR, QSRV, or have *ALLOBJ authority.

Input Parameters

CHGTYPE

Specifies the method of changing the date and/or time system values.

The possible values are:

- *SYSVAL—Indicates that the system date or time is set explicitly using **SYSVAL** and **VALUE** parameters.
- *FORWARD—Indicates that the system time is changed by moving the system clock forward by a specified number of hours, minutes, and/or seconds entered as separate parameters.
- *BACKWARD—Indicates that the system time is changed by moving the system clock backward by a specified number of hours, minutes, and/or seconds entered as separate parameters.

SYSVAL

Specifies the name of the system value to change. You can change either the system date or the system time.

This parameter is valid only when you specify *SYSVAL for the **CHGTYPE** parameter.

The new value must meet the format for the system value you specify to change.

The possible values are:

- QDATE—Specifies the system date. Changes take effect immediately.
- QTIME—Specifies the system time. Changes take effect immediately.

VALUE

Specifies the new value for the system parameter (either the system date or time) to change. The value that you enter must meet the format for the system parameter that you are changing.

This parameter is valid only when you specify *SYSVAL for the **CHGTYPE** parameter.

SECONDS

The number of seconds that the system time is incremented or decremented if either *FORWARD or *BACKWARD respectively is selected for **CHGTYPE** parameter.

Valid entries range from 0–59 seconds.

MINUTES

The number of minutes that the system time is incremented or decremented if either *FORWARD or *BACKWARD respectively is selected for **CHGTYPE** parameter.

Valid entries range from 0–59 minutes.

HOURS

The number of hours that the system time is incremented or decremented if either *FORWARD or *BACKWARD respectively is selected for **CHGTYPE** parameter.

Valid entries range from 0–23 hours.

Output

Depending on which system value you modified, either the system date or time is modified.

Example 1

```
DMCHGTIME CHGTYPE(*SYSVAL) SYSVAL(QDATE) VALUE('111913')
```

Assuming the system date is in the mmddyy format, this command changes the system date to November 19, 2013.

Example 2

```
DMCHGTIME CHGTYPE(*BACKWARD) HOURS(1)
```

Changes the system time backward by one hour.

Restrictions

None

Minimum Authority Level

You must be signed on as QPGMR, QSYSOPR, or QSRV, or else have *ALLOBJ authority.

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **31**

DMCLRLOG (Clear Cluster Event Log command)

Removes event messages from the event log.

```
DMCLRLOG EVNTTYPE( ) REPLEVNT( ) COMMEVNT( ) CLUSEVNT( )
```

Event messages are generated by iCluster during cluster operations. The messages can be filtered by replication, communication, and cluster event parameters.

Input Parameters

EVNTTYPE

The type of event messages to remove.

The possible values are:

- *REPL—Removes all event messages that provide information about replication.
- *ALL—Removes all types of event messages. By default, this value is used.
- *COMM—Removes all event messages that provide information about iCluster communications between nodes.
- *CLUSTER—Removes all event messages that provide information concerning cluster nodes and groups, as well as the outcome of some iCluster setup and configuration.

REPLEVNT

A parameter that determines which replication event messages are removed. REPLEVNT only has to be specified when **EVNTTYPE** is set to *ALL or *REPL.

Note: If **EVNTTYPE** is not *ALL or *REPL, the command ignores this parameter.

The parameter that filters replication event messages is as follows:

Group Name

The name of a group that is used to determine which messages are removed from the event log. Only messages that address the named group are removed.

The possible values are:

<name>—Removes all messages that address a specific group.

*ALL—Removes all replication event messages regardless of group. By default, this value is used.

COMMEVNT

A parameter that determines the communication event messages that are removed from the log. COMMEVNT only has to be specified when **EVNTTYPE** is set to *ALL or *COMM. You cannot filter communication event messages by group name.

 **Note:** If **EVNTTYPE** is not *ALL or *COMM, the command ignores this parameter.

The node name parameter filters communication event messages.

Node Name

The name of the node that generated the communication event messages.

The possible values are:

- <name>—Specifies the name of the node that generated the cluster event messages to remove from the event log.
- *ALL—Removes messages generated by all nodes. By default, this value is used.
- *LOCAL—Removes messages generated by the node you are currently using.

CLUSEVNT

A parameter that determines which cluster event messages are removed. CLUSEVNT only has to be specified when **EVNTTYPE** is set to *ALL or *CLUSTER. You cannot filter cluster event messages by group name.

 **Note:** If **EVNTTYPE** is not *ALL or *CLUSTER, the command ignores this parameter.

The parameter that filters cluster event messages is as follows:

Node Name

Specifies the name of the node that generated the cluster event messages to remove from the event log. Only *LOCAL can be specified. This setting removes cluster event messages that were generated by the node you are currently using.

Example 1

```
DMCLRLOG EVNTTYPE(*ALL) REPLEVNT(*ALL) COMMEVNT(NODE2) CLUSEVNT(*LOCAL)
```

Removes all types of event messages based on the specified filters.

The replication event messages that are removed refer to any group in the cluster.

The communication event messages that are removed are those that were generated by node NODE2.

The cluster event messages that are removed were generated by the local node.

Example 2

```
DMCLRLOG EVNTTYPE(*REPL) REPLEVNT(*ALL)
```

Removes event messages that provide information about replication (communications and cluster event messages are kept).

The replication event messages that are removed refer to any group in the cluster.

Restrictions

None.

Minimum Authority Level

*USER

Menu Access

Main Menu - Option 5

iCluster Commands menu - F22 (Shift + F10) - Option 33

HAPNGTCP (Ping Using TCP command)

Attempts to contact a remote node in the cluster to determine if the node is available and operational.

```
HAPNGTCP RMTNME( ) RMTPRT( ) PKTLEN( ) NBRPKT( ) RMTLIB( ) JOBD( )
```

After entering this command, you may have to wait a few seconds before receiving a response. Use this command as a diagnostic aid if you are having trouble replicating objects and data to the remote node. This command allows you to verify iCluster communications.

Input Parameters

RMTNME

The host name of a remote node in the cluster.

RMTPRT

The port number on the remote node that has been reserved for iCluster communications.

You should specify the port number that is used in the node's definition in iCluster.

The default value is 4545.

PKTLEN

The length, in bytes, of the packets you are sending to the remote node in the cluster.

The length can range from 32–32760 bytes.

The default value is 256.

NBRPKT

The number of packets you are sending to the remote node in the cluster.

This number can range from 1-999 packets.

The default value is 5.

RMTLIB

The name of the library on the remote node where iCluster was installed.

The default value is ICLUSTER.

JOBID

The job description associated with iCluster jobs running on the remote node. The default value is CSJOBD.

Example

```
HAPNGTCP RMTNME(NYSYS1) RMTPRT(4545) PKTLEN(256) NBRPKT(5) JOBD(ICJOBDB)
```

Attempts to contact the remote node that is identified by the host name **NYSYS1**, and whose remote port number is 4545.

Five packets are sent, with each packet being 256 bytes in length.

The job description **ICJOBDB** in the **ICLUSTER** library is associated with iCluster jobs that are running on the remote node.

Restrictions

To invoke this command, iCluster must be installed on both the local and remote nodes, and the iCluster TCP listener job must be running on the remote node in the iCluster subsystem **XDMCLUSTER**.

Minimum Authority Level

Any user with access to the iCluster product library.

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **34**

STRHATCP (Start TCP/IP Listener command)

Starts the **DMCHATL** TCP/IP listener job on the local node in the **XDMCLUSTER** subsystem.

```
STRHATCP JOBD( ) LIB( ) HOST( ) SERVICE( )
```

This listener job waits for iCluster messages from another node in the cluster. The listener job must be started before an iCluster node can communicate with another node using TCP/IP.

Input Parameters

JOBID

The job description to associate with the TCP/IP listener job (DMCHATL) that is started by this command.

The default value is CSJOBD.

LIB

The library that contains the job description you want to associate with the TCP/IP listener job (DMCHATL) that is started by this command.

The default value is ICLUSTER.

HOST

The name of the local host. This parameter is usually omitted to allow the listener job to accept incoming connection requests directed to any local address. You should specify this parameter only for a multi-homed site to restrict the listener to accept incoming connection requests to a single local address.

SERVICE

The name of the service that identifies the port on which the listener job listens.

The default service is dmcluster.

Example

```
STRHATCP JOBD(TCPJOBDB)
```

Starts the TCP/IP listener job with the job description **TCPJOBDB** in the **ICLUSTER** library.

Restrictions

You must issue this command on each node. The **DMCHATL** TCP/IP listener job must be running on all active nodes that are defined in the cluster. The **XDMCLUSTER** subsystem must be started before the **DMCHATL** job can be started.

Minimum Authority Level

Any user with access to the iCluster product library.

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **35**

WRKHAJOB (Work with Active Cluster Jobs command)

Displays the active cluster jobs for the subsystem that iCluster is running under (**XDMCLUSTER**), plus the subsystems **QCMN** and **QSYSWRK**.

```
WRKHAJOB
```

To view all active jobs regardless of the subsystem, execute the IBM i command **WRKACTJOB**.

Input Parameters

None.

Example

```
WRKHAJOB
```

Displays the **Work with Active Jobs** screen so that you can view active cluster jobs for the subsystem that iCluster is running under, plus the subsystems QCMN and QSYSWRK.

Restrictions

You must issue this command on the node where the operation will be performed.

Minimum Authority Level

Any user with access to the iCluster product library.

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **36**

DMDLTCLSTR (Delete Cluster command)

Ends cluster operations and delete cluster definitions.

```
DMDLTCLSTR CLSTR( )
```

When this command is invoked on an active node, cluster operations are ended and cluster definitions that are maintained by iCluster are deleted on all active nodes in the cluster. When this command is invoked on an inactive node, cluster operations are stopped and cluster definitions are deleted only on the node where the command is invoked. To ensure that all cluster operations are stopped and all cluster definitions are deleted, issue this command on each node in the cluster.

Input Parameters

CLSTR

The name of the cluster to delete.

The default value is *CURRENT.

When iCluster uses Cluster Resource Services, *CURRENT means the cluster in which the current system is defined. When Cluster Resource Services is not used, the name of the cluster is internally set to DM_CLUSTER, and cannot be changed.

Example

```
DMDLTCLSTR CLSTR(DM_CLUSTER)
```

Stops cluster operations and deletes cluster definitions in the cluster DM_CLUSTER. If invoked from an active node, cluster operations are stopped and cluster definitions are deleted on all active nodes in the cluster.

Restrictions

You can issue this command on any node in the cluster.

Minimum Authority Level

*ADMIN

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **37**

DMSETMSTR (Set Metadata Owner command)

Sets the current node as the owner of the iCluster metadata when the node that currently owns the metadata is no longer accessible to the cluster.

DMSETMSTR

If the node that is configured as the owner of the iCluster metadata is inaccessible, the ownership of the iCluster metadata is normally transferred to another active node in the cluster. If this automatic change does not occur, use this command to move the ownership of the iCluster metadata to another node in the cluster.

You can issue this command on any node that is not the current owner of the iCluster metadata.

- If the current node is active, this command assigns the ownership of the iCluster metadata to the node that should have become the owner of the metadata when the previous node that was the owner of the iCluster metadata ended.
- If the current node is inactive, this command sets the current node as owner of the iCluster metadata. The metadata ownership takes over when the current node is started.

If necessary, remove the previous owner of the iCluster metadata with the **DMRMVNOD**E (Remove Node) command.

Example

DMSETMSTR

There are no parameters.

Restrictions

The failover mechanism must be SwitchOver System. Verify that the **Use IBM Cluster Services for i** system value is *NO as defined with the **DMSETSVAL** (Set Cluster System Values) command.

Minimum Authority Level

*ADMIN

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **39**

Report commands

Use the report commands to produce and view reports for iCluster status, operations, configuration, object counts, sync check, and other aspects of the product and replication.

Related information

- [DMCFGRPT \(Configuration Report command\)](#)
- [DMSYSINF \(System Information command\)](#)
- [DMGRPSTS \(Group Status Report command\)](#)
- [DMSCRPT \(Sync Check Report command\)](#)
- [DMRPLCVRPT \(iCluster Replication Coverage command\)](#)
- [DMDSPLLOG \(Display Cluster Event Log command\)](#)
- [DMRPTTRG \(Report Files with Triggers command\)](#)
- [DMRPTCST \(Report Files with Constraints command\)](#)
- [DMANZJRN \(Analyze Journals command\)](#)
- [DMRPTXLF \(Report Cross-library LFs command\)](#)
- [DMOBJCNT \(Report Object Counts command\)](#)
- [DMAUDITRPT \(Audit Report command\)](#)
- [DMSUSOBRPT \(Suspended Object Report command\)](#)

DMCFGRPT (Configuration Report command)

Use on any node in the cluster to produce the iCluster Configuration Report.

```
DMCFGRPT NODES( ) GROUPS( ) OBJSPECs( ) SMPOBJSPEC( ) SYSVALs( ) OUTPUT( )
```

Input Parameters

NODES

Specifies the node or nodes in the cluster to include in the configuration report.

The possible values are:

- *NONE—Do not list nodes. By default, this value is used.
- *ALL—All nodes.
- <name>—The name of a single node.

GROUPS

Specifies the group or groups in the cluster to include in the configuration report.

- *NONE—Do not list groups. By default, this value is used.
- *ALL—All groups.
- <name>—The name of a single group.

OBJSPECs

Specifies whether to include object specifiers and their parameters in the configuration report.

- *NONE—Do not list object specifiers for any group in the cluster. By default, this value is used.
- *ALL—List all object specifiers of all groups in the cluster, including native object specifiers and BSF path specifiers.
- <name>—The name of a single group. List the object specifiers only for this group, including native object specifiers and BSF path specifiers.

SMPOBJSPEC

Specifies the group of the cluster whose object specifiers with simple attributes are the same as the **Work with Object Specifiers by Group** screen to list in the configuration report. Values include:

- *NONE—The configuration report does not list the object specifiers from any of the groups in the cluster.
- *ALL—The configuration report lists the object specifiers of all groups in the cluster with simple attributes. This includes native object specifiers, SQL specifiers, and BSF path specifiers.
- <name>—Specifies the name of a simple group whose object specifiers with simple parameters are listed in the configuration report. This includes native object specifiers, SQL specifiers, and BSF path specifiers.

SYSVALS

Specifies whether to include system values in the configuration report.

- *NO—Do not list system values. By default, this value is used.
- *YES—List system values.

OUTPUT

Specifies where to write the report output.

The possible values are:

- *—Displays the report and writes it to a spooled file, when the command is run in interactive mode. Writes the group status information only to a spooled file, when the command is run in batch mode.
- *PRINT—Writes the report only to a spooled file for print.

Example

```
DMCFGRPT NODES(*ALL) GROUPS(*ALL) OBJSPECS(*ALL) SMPOBJSPEC(*ALL) SYSVALS(*YES) OUTPUT(*)
```

Produces the iCluster Configuration Report for all nodes and all groups. The configuration report also lists the object specifiers in detail and simple format for all groups, and includes the system values. Displays the report and writes it to a spooled file.

Restrictions

None.

Minimum Authority Level

*USER

Menu Access

Main Menu - F22 (Shift + F10) - Option **9 - iCluster Reports** menu - Option **1**

DMSYSINF (System Information command)

Use on any node in the cluster to produce the iCluster System Report.

```
DMSYSINF OUTPUT( )
```

The nodes are displayed in the order that they were added to the cluster. Only the node name and IP address are displayed for nodes that are down.

The following iCluster system information is displayed and printed to the spooled file:

Field	System information
Local Node Name	Node on which the command is run
Local System Name	System on which the command is run
iCluster version	Installed iCluster version where this command is run
Additional features	Additional features that are installed.
Use Cluster Resource Services	Failover mechanism is SwitchOverSystem when Use Cluster Resource Services is *NO
Node	Node names
System	System name
IP address	Node IP addresses
OS Rls	IBM i operating system version
Type-Mod	IBM Power Systems server type and model number
P-Grp	IBM Power Systems Processor Group
Serial #	IBM Power Systems server hardware serial number
Sys Time	System time that the command was run
Staging Store Used (MB)	The total disk space in MB that the staging store is using for node and group level staging stores

Input Parameters

OUTPUT

Specifies where to write the report output.

The possible values are:

- *—Displays the report and writes it to a spooled file, when the command is run in interactive mode. Writes the group status information only to a spooled file, when the command is run in batch mode. By default, this value is used.
- *PRINT—Write the report only to a spooled file for print.

None.

Example

```
DMSYSINF
```

Displays system information.

Restrictions

None

Minimum Authority Level

*USER

Menu Access

Main Menu - Option 7

Main Menu - Option 9 - Option 2

DMGRPSTS (Group Status Report command)

Use on any node in the cluster to produce the iCluster Group Status Report.

```
DMGRPSTS GROUP( ) OUTPUT( ) DETAILS( ) JRNPOS( )
```

The following iCluster group status information is displayed and printed to the spooled file:

Input Parameters

GROUP

Indicates which group's status information is to be retrieved.

The possible values are:

- *ALL—Specifies that group status information is to be retrieved for all groups in the cluster. Note that groups without a replication history or that are ineligible for replication, for example, *STDBY groups, are not listed. By default, this value is used.
- group-name—Specifies that group status information is to be retrieved only for the specified group.

OUTPUT

Indicates whether the group status information is displayed on the screen or written to a spooled file for printing.

The possible values are:

- *—Displays the group status information on the console and writes it to a spooled file when the DMGRPSTS command is run in an interactive job. When the DMGRPSTS command is run in a batch job, the output is only written to the spooled file. By default, this value is used.
- *PRINT—Writes the report only to a spooled file for print.

DETAILS

Indicates whether the group status details are added to the report.

The possible values are:

- *NO—Does not add the group status details to the report. By default, this value is used.
- *YES—Adds the group status details to the report in a separate section from the group summary list.

JRNPOS

Indicates whether the group's journal positions are added to the report.

The possible values are:

-
- *NO—Does not add the group's journal positions to the report. By default, this value is used.
 - *YES—Adds the group's journal positions to the report in a separate section from the group summary list.

Example

```
DMGRPSTS OUTPUT(*PRINT) DETAILS(*NO)
```

Creates the group status report without group status details and writes it to the spooled file for printing.

Restrictions

None

Minimum Authority Level

*USER

Menu Access

Main Menu - Option 7

Main Menu - Option 9 - Option 3

DMSCRPT (Sync Check Report command)

Use on the backup node in the cluster to produce the iCluster Sync Check Report.

```
DMSCRPT GROUP( ) SORTBY( ) OUTPUT( ) SENDTO( ) DETAILS( ) BACKUP( )
```

The node can be active or inactive. The report lists sync check results only for the backup node where it is issued and for the groups that you specify.

Input Parameters

GROUP

Specifies the group or groups to include in the report. The groups must have been sync checked.

The possible values are:

- <name>—The name of a single group when you run this command on the backup node for the group.
- *LOCAL—All groups that have the current node as their backup node. By default, this value is used.

SORTBY

Specifies how to sort the output of the sync check report.

SORTBY is valid only if the value of the **GROUP** parameter is set to *LOCAL.

The possible values are:

- *GROUP—Sort the sync check results alphabetically by group name. By default, this value is used.
- *OBJECT—Sort the sync check results alphabetically by object name.

OUTPUT

Specifies where to write the report output.

The possible values are:

- *—Displays the report and writes it to a spooled file, when the command is run in interactive mode. Write the report only to a spooled file, when the command is run in batch mode. By default, this value is used.
- *PRINT—Writes the report only to a spooled file.

SENDTO

Specifies where to send the spooled file results of the sync check. This parameter is ignored if the value of the **GROUP** parameter is *LOCAL.

- ! **Note:** To send a report to print on a node, that node's LPD port must be available. For example, to send the spooled file results of a sync check from the backup node to the primary node, the LPD port on the primary node must be open. See "Adding the dmcluster entry to the TCP/IP service table" in the *Rocket iCluster Installation Guide*, or check with your network administrator.

The possible values are:

- *ALL—Send the spooled file results of the sync check from the backup node to all active nodes in the cluster. By default, this value is used.
- *PRIMARY—Send the spooled file results from the backup node to the primary node, as long as the primary node is active.
- *BACKUP—Do not send the spooled file results of the sync check to any other node. The spooled file results exist only on the backup node of the specified group.

- ! **Note:** The sync check report is sent to the output queue that is listed in the CSJOB job description in the ICLUSTER library on the group's backup node. If the output queue listed in the CSJOB job description on the backup node does not exist on the node to which the report is sent, the report is put into the output queue specified for spooled files received through a network (use the DSPNETA command to see the name of this output queue).

DETAILS

Specifies whether to list the details of the mismatched attributes of the out-of-sync objects.

The possible values are:

- *YES—Includes details of the mismatched attributes.
- *NO—Does not include details. By default, this value is used.

BACKUP

The name of the backup node of the group or groups listed in the report.

- ! **Note:** If the backup node named in this parameter is not the current node or *CURRENT, the report will only be available as a spooled file. It will not be displayed on the screen even if the OUTPUT parameter is set to *.

The possible values are:

- Backup node name
- *CURRENT

The sync check report will list the sync check results for the current node as the backup node.

Example

```
DMSCRPT GROUP(*LOCAL) SORTBY(*GROUP) SENDTO(*PRIMARY) OUTPUT(*) BACKUP(*CURRENT)
```

All the groups that have the current node as the backup node are used in the sync check report.

The output of the report is sorted alphabetically by group name.

The report is sent to the primary node.

The output of the report is displayed and written to a spooled file.

Restrictions

The command must be used on the backup node for the groups that you specify in this command.

Minimum Authority Level

*USER

Menu Access

Main Menu - F22 (Shift + F10) - Option **9 - iCluster Reports** menu - Option **4**

DMRPLCVRPT (iCluster Replication Coverage command)

Use this command on any node to produce the iCluster Replication Coverage Report. This report lists the user libraries, IFS directories, and DLO folder on the node where the command is issued.

```
DMRPLCVRPT CVGTYPE( ) ORDER( ) GROUP( ) OUTPUT( )
```

The report truncates directory path names that are longer than 50 characters. Truncated path names are assigned a numerical identifier, and full path names are listed in a mapping table at the end of the report. The latest report data is in the **DMRPLCVG** database file in the **ICLUSTER** library.

Directory pathnames cannot be listed if their owner is QSYS or the owner is owned by QSYS. Directory pathnames in these paths cannot be listed: '/QSYS.LIB/', '/QLANSrv/', '/QOPT/', '/QFileSvr.400/', '/UDFS/', '/NFS/', '/QNetWare/', '/QNTC/'.

Input Parameters

CVGTYPE

Specifies the type of replication coverage to include in the report.

The possible values are:

- *ALL—Include libraries, IFS directories, and DLO folders. The first part of the two-part report includes libraries, the second part includes IFS directories and DLO folders. By default, this value is used.
- *LIBRARY—Include only libraries.
- *SQL—Include only libraries which have SQL objects.

- *PATH—Include only IFS directories and DLO folders.

ORDER

Specifies how to sort the output of the report. Library objects are listed separately.

The possible values are:

- *ALPHA—Sort the output alphabetically by libraries and IFS directories and DLO folders. By default, this value is used.
- *NOTREPL—Sort the output with the libraries and IFS directories and DLO folders that are not being replicated first, and then the libraries and user IFS directories and DLO folders that are being partially replicated by iCluster, followed by the user libraries and user IFS directories and DLO folders that are being replicated by iCluster.
- *REPL—Sort the output with the libraries and IFS directories and DLO folders that are being replicated first, and then the libraries and user IFS directories and DLO folders that are being partially replicated by iCluster, followed by the libraries and user IFS directories and DLO folders that are not being replicated by iCluster.
- *ELIGIBLE—Sort the output with the libraries and IFS directories and DLO folders that are eligible for replication first. Libraries and IFS directories and DLO folders that are not eligible for replication are listed next.

GROUP

Specifies the name of the group for which to list all replicated objects (native and BSF) in the report.

The possible values are:

- *NONE—Does not display the Replication Coverage report for a group.
- *ALL—Displays the Replication Coverage report for all groups.
- Name—Displays the Replication Coverage report for a specified group.

OUTPUT

Specifies where to write the report output.

The possible values are:

- *—Displays the report and writes it to a spooled file, when the command is run in interactive mode. Writes the group status information only to a spooled file, when the command is run in batch mode. By default, this value is used.
- *PRINT—Writes the report only to a spooled file for print.

The maximum number of records reported to the spooled file is 100,000 by default. Increasing the value of the MAXRCDs attribute of the DMRPLCVRPT printer file will report more records but consume more disk space. For example:

```
CHGPRTF FILE(ICLUSTER/DMRPLCVRPT) MAXRCDs(990000)
```

*NOMAX can be used if there is a large amount of disk space available for the file.

Example

```
DMRPLCVRPT CVGTYPE(*ALL) ORDER(*ALPHA) GROUP(*NONE) OUTPUT( * )
```

The report lists the replication coverage of libraries, IFS directories, and DLO folder replication coverage.

The output of the report is sorted (alphabetically) by library and IFS directory.

The output of the report is displayed and written to a spooled file.

Restrictions

Your IBM i user profile must be enrolled in the system directory to use this command for DLO folders. To access all of the libraries and directories on the system, your IBM i user profile should have *ALLOBJ authority.

Minimum Authority Level

*USER

Menu Access

Main Menu - F22 (Shift + F10) - Option **9 - iCluster Reports** menu - Option **5**

DMDSPLOG (Display Cluster Event Log command)

Displays event messages that are generated during cluster operations.

```
DMDSPLOG EVNTTYPE( ) REPLEVNT( ) COMMEVNT( ) CLUSEVNT( ) STRDATE( ) STRTIME( )
ENDDATE( ) ENDTIME( ) MSGID( ) OMITMSGID( ) OUTPUT( ) DETAIL( ) MSGLVLS( )
```

You can filter messages by replication, communication, and cluster type events. You can also filter event messages by time and date, message ID, message type, and message level.

Input Parameters

EVNTTYPE

The type of event messages to display.

The possible values are:

- *REPL—Displays all event messages that provide information about replication. By default, this value is used.
- *ALL—Displays all types of event messages.
- *COMM—Displays all event messages that provide information about iCluster communications between nodes.
- *CLUSTER—Displays all event messages that provide information concerning cluster nodes and groups, as well as the outcome of some iCluster setup and configuration operations.

REPLEVNT

Specifies which replication event messages are displayed when **EVNTTYPE** is *ALL or *REPL.



Note: This parameter is ignored if the **EVNTTYPE** parameter is not *ALL or *REPL.

Group Name

The possible values are:

- <name>—Specifies a group name. The name of an existing group. Only replication event messages for the specified group are displayed.

- *ALL—Displays all replication event messages. By default, this value is used.

COMMEVNT

Specifies which communication event messages are displayed when **EVNTTYPE** is set to *ALL or *COMM.

 **Note:** This parameter is ignored if **EVNTTYPE** is not *ALL or *COMM.

Node Name

The possible values are:

- <name>—The name of the node that generated the communication event messages. You cannot filter communication event messages by group name.
- *ALL—Displays communication event messages generated by all nodes. By default, this value is used.
- *LOCAL—Displays communication event messages generated by the node you are currently using.

CLUSEVNT

Specifies which cluster event messages are displayed when **EVNTTYPE** is set to *ALL or *CLUSTER.

 **Note:** This parameter is ignored if **EVNTTYPE** is not *ALL or *CLUSTER.

Node Name

The name of the node that generated the cluster event messages. Only *LOCAL can be specified. This setting displays cluster event messages generated by the node you are currently using.

STRDATE

The earliest date from which to display event messages from the event log.

The possible values are:

- blank—Display event log messages starting from the beginning of the event log. By default, this value is used.
- <date>—The earliest date from which to display event messages from the event log. The date must be entered with the same date format determined by the 'Date format' and 'Date separator' job definition attributes of the current job.

Example: Specify '2020,100' when 'Date format' is *JUL and 'Date separator' is ','. Specify 10/12/20' when 'Date format' is MDY and 'Date separator' is '/'.

STRTIME

The earliest time from which to display event messages from the event log. This parameter applies only when a date is specified for the **STRDATE** parameter.

The possible values are:

- blank—Display all event log messages. By default, this value is used.
- <time>—The earliest time from which to display event messages from the event log. The time must be entered in HH MM SS format with the same time separator determined by the 'Time separator' job definition attribute of the current job.

Example: Specify '09:20:56' when 'Time separator' is ':'. Specify '09.20.56' when 'Time separator' is '.'.

ENDDATE

The latest date from which to display event messages from the event log.

The possible values are:

- blank—Display event log messages to the end of the event log. By default, this value is used.
- <date>—The latest date from which to display event messages from the event log. The date must be entered with the same date format determined by the 'Date format' and 'Date separator' job definition attributes of the current job.

Example: Specify '2020,100' when 'Date format' is *JUL and 'Date separator' is ','. Specify '10/12/20' when 'Date format' is MDY and 'Date separator' is '/'.

ENDTIME

The latest time from which to display event messages from the event log.

The possible values are:

- blank—Display event log messages to the end of the event log. By default, this value is used.
- <time>—The latest time from which to display event messages from the event log. The time must be entered in HH MM SS format with the same time separator determined by the 'Time separator' job definition attribute of the current job.

Example: Specify '09:20:56' when 'Time separator' is ':'. Specify '09.20.56' when 'Time separator' is '.'.

MSGID

Specifies to display messages that match the specified message IDs.

You can find message IDs in the event log.

The possible values are:

- <value>—Specifies up to 20 unique message IDs.
- *ALL—Displays messages for all message IDs. By default, this value is used.
- *LATNCY—Displays only messages whose message ID is OMI0308. These messages are generated when either the primary receive latency or the backup apply latency threshold is exceeded. The **EVNTTYPE** must be *ALL or *REPL.

Note: When latency exceeds its threshold, message OMI0308 is issued. When latency drops back below the threshold, message OMI0318 is issued.

- *SUSPND—Displays only messages that are related to object suspension. The **EVNTTYPE** parameter must be *ALL or *REPL.

OMITMSGID

Specifies a list of message identifiers to be omitted from the event log display. *NONE is the default.

The possible values are:

- *NONE— The DMDSPOLOG output will not omit any message identifiers from the event log display.
- <message-identifier-list>— You can specify up to 20 message identifiers to be omitted from the event log display. These message identifiers can be for messages of any severity.

OUTPUT

Specifies where to display the messages.

The possible values are:

- *—Displays messages. By default, this value is used.

- *PRINT—Directs messages to a spooled file.

DETAIL

Specifies the level of detail for each message that is displayed or directed to a spooled file.

The possible values are:

- *BASIC—Displays first level message text. By default, this value is used.
- *FULL—Displays more detailed messages (first and second levels of message text).

MSGLVLS

Specifies the severity levels of the messages to display.

- ! **Note:** To display the messages of a particular severity level, you must ensure that the messages are generated by setting an appropriate **MSGLVLS** system value in the DMSETSVAL (Set Cluster System Values) command.

The possible values are:

- *ALL—Displays messages in all levels. Recommended for initial use to see all of the messages. By default, this value is used.
- 00—Displays information messages (Level 00). For example, Savefile created.
- 10—Displays non-critical status messages (Level 10).
- 20—Displays stop/start messages (Level 20). For example, Start Mirroring.
- 30—Displays messages that report recoverable errors (Level 30).

When specific message levels are identified, messages in other levels are not displayed. Fatal error messages (Level 40) are always displayed.

To display more than one message level, specify *ALL or separate each message level in the parameter with a space. For example, MSGLVLS (10 20) specifies to display only status and warning messages.

- ! **Note:** After the nodes and groups are defined, you can identify the specific level of messages to display.

Example 1

```
DMDSPLOG EVNTTYPE(*ALL) REPLEVNT(*ALL) COMMEVNT(NODE2) CLUSEVNT(*LOCAL) OUTPUT(*)
DETAIL(*FULL) MSGLVLS(*ALL)
```

Displays all types of event messages that are based on the specified filters.

The replication event messages that are displayed refer to any group in the cluster.

The communication event messages were generated by node **NODE2**.

The cluster event messages were generated by the local node.

Messages contain first and second level text and are displayed.

Messages in all levels are displayed.

Example 2

```
DMDSPLLOG EVNTTYPE(*REPL) REPLEVNT (*ALL) OUTPUT(*PRINT) DETAIL(*BASIC) MSGLVL(00)
```

Displays event messages that provide information about replication. Communications and cluster event messages are hidden.

These replication event messages refer to any group in the cluster.

Messages contain first level text, and are directed to the spooled output for the job.

Only information (Level 00) and fatal error (Level 40) messages are displayed.

Example 3

```
DMDSPLLOG EVNTTYPE(*COMM) COMMEVNT(NODE3) OUTPUT(*) DETAIL(*BASIC) MSGLVL(10 20)
```

Displays event messages that provide information about communications between primary and backup nodes. Replication and cluster event messages are hidden.

The communication event messages were generated by node **NODE3**.

Messages contain first level text and are displayed.

Only status (Level 10), warning (Level 20), and fatal error (Level 40) messages are displayed.

Example 4

```
DMDSPLLOG EVNTTYPE(*ALL) STRDATE('08/23/2014') STRTIME('09:00') ENDDATE('08/30/2014')  
ENDTIME('17:00') MSGID (*ALL) OUTPUT(*PRINT) DETAIL(*BASIC) MSGLVL(00)
```

Displays all event messages from August 23, 2014 at 9 a.m. to August 30, 2014 at 5 p.m. on a system where the date format is *MMDDYYYY*.

Messages contain first level text and are directed to a spooled file.

Only information (Level 00) and fatal error (Level 40) messages are displayed.

Example 5

```
DMDSPLLOG EVNTTYPE(*REPL) ENDDATE('08/30/2013') ENDTIME('17:00')
```

Displays all replication event messages until August 30, 2013 at 5 p.m.

Restrictions

You can issue this command from any node that is either active or inactive in the cluster.

Minimum Authority Level

*USER

Menu Access

Main Menu - Option 4

iCluster Commands menu - F22 (Shift + F10) - Option **32**

Main Menu - F22 (Shift + F10) - Option **9** - **iCluster Reports** menu - Option **6**

DMRPTTRG (Report Files with Triggers command)

Use on any node to produce the Files with Triggers Report.

```
DMRPTTRG GROUP( ) OUTPUT( ) OUTFILE( )
```

The report lists the files with triggers that are on the current node and are in replication scope for the specified group or groups. The report also lists the number of database files that are processed for each group.

Input Parameters

GROUP

Specifies the group or groups to include in the report. The object specifiers of the groups are used to select the files with triggers.

The possible values are:

- *ALL—All replication groups that have the current node as their primary node. By default, this value is used.
- <name>—The name of a single group.

OUTPUT

Specifies where to write the report output.

The possible values are:

- *—Displays the report and writes it to a spooled file, when the command is run in interactive mode. Writes the group status information only to a spooled file, when the command is run in batch mode. By default, this value is used.
- *PRINT—Writes the report only to a spooled file for print.

OUTFILE

Specifies whether to write the list of files and their triggers to a database file.

The possible values are:

- *NONE—Do not write the list of files and their triggers to a database file. By default, this value is used.
- <name>—The name and library of the database file. For example, LIB1/TRGLIST1. If the database file already exists, the file is overwritten with the new file.

Example

```
DMRPTTRG GROUP(MYGROUP) OUTPUT(*) OUTFILE(LIB1/TRGLIST1)
```

The report lists the files on the current node that have triggers and are in the replication group **MYGROUP**.

The output of the report is displayed and written to a spooled file.

The list of files and their triggers is written to the **TRGLIST1** database file in the **LIB1** library.

Restrictions

None.

Minimum Authority Level

*USER

Menu Access

Main Menu - F22 (Shift + F10) - Option **9 - iCluster Reports** menu - Option **7**

DMRPTCST (Report Files with Constraints command)

Use on any node to produce the Files with Constraints Report.

```
DMRPTCST GROUP( ) OUTPUT( ) OUTFILE( )
```

The report lists the files with constraints that are on the current node and are in replication scope for the specified group or groups. The report also lists the number of database files that are processed for each group.

Input Parameters

GROUP

Specifies the group or groups to include in the report. The object specifiers of the groups are used to select the files with constraints.

The possible values are:

- *ALL—All replication groups that have the current node as their primary node. By default, this value is used.
- <name>—The name of a single group.

OUTPUT

Specifies where to write the report output.

The possible values are:

- *—Displays the report and writes it to a spooled file, when the command is run in interactive mode. Writes the group status information only to a spooled file, when the command is run in batch mode. By default, this value is used.
- *PRINT—Writes the report only to a spooled file for print.

OUTFILE

Specifies whether to write the list of files and their constraints to a database file.

The possible values are:

- *NONE—Do not write the list of files and their constraints to a database file. By default, this value is used.

- <name>—The name and library of the database file. For example, LIB1/CSTLIST1. If the database file already exists, the file is overwritten with the new file.

Example

```
DMRPCST GROUP(MYGROUP) OUTPUT(*) OUTFIL(LIB1/CSTLIST1)
```

The report lists the files on the current node that have constraints and are in the replication group, MYGROUP.

The output of the report is displayed and written to a spooled file.

The list of files and their constraints is written to the CSTLIST1 database file in the LIB1 library.

Restrictions

None.

Minimum Authority Level

*USER

Menu Access

Main Menu - F22 (Shift + F10) - Option **9 - iCluster Reports** menu - Option **8**

DMANZJRN (Analyze Journals command)

Use on any node to analyze the specified journal or journals and produce journal statistics reports.

```
DMANZJRN JRN( ) STRDATE( ) STRTIME( ) ENDDATE( ) ENDTIME( ) CLRANZ( )
```

This command submits a job that performs the journal analysis and generates the reports.

The reports are generated as database files and spooled files with self-describing file names that are derived from QPRINT plus the user-specified data. The records in the database files are derived from the DMANZJRNPF file in the iCluster library. The basic journal analysis data is stored in the DMANZJRNPF file during DMANZJRN command processing. The journal receivers that are processed for analysis are stored in the DMANZJRCV file.

Input Parameters

JRN

Specifies the journal or journals to analyze. This is a required parameter.

The possible values are:

- *ALL—Analyze all journals. Available only when the **STRJRNRCV** parameter is set to *CURRENT or *CURCHAIN and the **ENDJRNRCV** parameter is set to *CURRENT.
- *ALLDTA—Analyzes all database journals except the **QAUDJRN** journal in the **QSYS** library. Available only when the **STRJRNRCV** parameter is set to *CURRENT or *CURCHAIN and the **ENDJRNRCV** parameter is set to *CURRENT.
- <name>—The journal name and library. Prefix the journal with the name of the library where the journal is located. For example, LIB1/JRN1.

STRJRNRCV

Specifies a journal receiver to identify where to start the analysis.

The possible values are:

- *CURCHAIN—Specifies the first journal receiver in the chain for the journal.
- *CURRENT—Specifies the current journal receiver for the journal.
- <name>—The journal receiver name and library. Specify the library where the journal receiver is located.
For example, LIB1/JRNRCV1.

ENDJRNRCV

Specifies a journal receiver to identify where to end the analysis.

The possible values are:

- *CURCHAIN—Specifies the first journal receiver in the chain for the journal.
- *CURRENT—Specifies the current journal receiver for the journal.
- <name>—The journal receiver name and library. Specify the library where the journal receiver is located.
For example, LIB1/JRNRCV1.

If you specify ENDJRNRCV(*CURRENT), or the specified journal receiver library and name match the currently attached journal receiver of the journal, the journal is updated with a newly attached receiver. The journal is not analyzed for the newly attached receiver, but is analyzed to the previously attached receiver.

STRDATE

The earliest date from which to start the journal analysis.

The possible values are:

- blank—Start journal analysis from the earliest journal receiver in the receiver range. By default, this value is used.
- <date>—In the journal date format, the earliest date from which to start the journal analysis.

STRTIME

The earliest time from which to start the journal analysis.

The possible values are:

- blank—Start journal analysis from the earliest journal receiver in the receiver range. By default, this value is used.
- <time>—Applies only when the **STRDATE** parameter is not blank. In journal time format, the earliest time from which to start the journal analysis.

ENDDATE

The latest date of the journal entry to end the analysis.

The possible values are:

- blank—End journal analysis with the currently attached journal receiver. By default, this value is used.
- <date>—In the journal date format, the latest date of a journal entry to include in the journal analysis.

ENDTIME

The latest time from which to display event messages from the event log.

The possible values are:

- blank—End journal analysis with the currently attached journal receiver. By default, this value is used.

- <time>—Applies only when the **ENDDATE** parameter is not blank. In journal time format, the latest time of a journal entry to include in the journal analysis.

CLRANZ

Specifies whether to clear the **DMANZJRNPF** and **DMANZJRCV** analysis files before starting the journal analysis.

The possible values are:

- *NO—Do not clear the **DMANZJRNPF** and **DMANZJRCV** files before journal analysis. Improves command processing performance when the **DMANZJRN** command is already run on the same journals with the same receiver range and the same date and time range. By default, this value is used.
- *YES—Clear the **DMANZJRNPF** and **DMANZJRCV** analysis files before starting the journal analysis.

Example

```
DMANZJRN JRN(ICLUSTER/HADJRN) CLRANZ(*YES)
```

The report lists.

The output of the report is written to a spooled file and a database file.

The contents of the **DMANZJRNPF** and **DMANZJRCV** analysis files are cleared before the journal analysis is started.

Restrictions

None.

Minimum Authority Level

*USER

Menu Access

Main Menu - F22 (Shift + F10) - Option **9 - iCluster Reports** menu - Option **10**

DMRPTXLF (Report Cross-library LFs command)

Use on any node to produce the iCluster Cross-library Logical Files Report.

```
DMRPTXLF LIB( ) OUTPUT( )
```

Depending on what libraries you specify, the report search can be extensive. The recommended way to produce the report is to submit a batch job that runs the **DMRPTXLF** command.

Cross-library logical files have based-on physical files in libraries that are different from the libraries of the logical files. Cross-library physical files have logical files in libraries that are different from the libraries of the physical files. The results of the report are sent to the **DMRPTXLF** spooled file.

Input Parameters

LIB

Specifies the library or libraries to search. This is a required parameter.

The possible values are:

- *ALL—Search all libraries on the system.
- *ALLUSR—Search all user libraries on the system except those that are owned by QSYS, or owned by a user profile that is owned by QSYS. Libraries that are not searched are listed at the end of the report.
- *LIBL—Search all libraries in the current library list.
- *USRLIBL—Search all user libraries in the current library list.
- <name>—The name of the library to search.

OUTPUT

Specifies where to write the report output.

The possible values are:

- *—Displays the report output on the console and writes it to a spooled file when the command is run in interactive mode. Writes the report output only to a spooled file when the command is run in batch mode. This is the default value.
- *PRINT—Writes the report output only to a spooled file.

Example

```
DMRPTXLF LIB(MYLIB)
```

The report lists the cross-library logical files that are based on physical files in libraries that are different from the **MYLIB** library of the logical files, and then lists the cross-library physical files that have logical files in the **MYLIB** library that are different from the libraries of the physical files. The results of the report are sent to the **DMRPTXLF** spooled file.

The output of the report is written to the console and the **DMRPTXLF** spooled file.

Restrictions

None.

Minimum Authority Level

*USER

Menu Access

Main Menu - F22 (Shift + F10) - Option **9** - **iCluster Reports** menu - Option **11**

DMOBJCNT (Report Object Counts command)

Use on any node to produce the iCluster Object Counts Report.

```
DMOJCNT LIB( ) MAXWAIT( ) OUTPUT( )
```

The report lists the object counts and total sizes for the objects in the specified library or libraries. Depending on what libraries you specify, the report search can be extensive. The recommended way to produce the report is to submit a batch job that runs the **DMOJCNT** command.

The report is produced in the **DMOJCNT** spooled file, and optionally displayed on the console.

Input Parameters

LIB

Specifies the library or libraries to search. This is a required parameter.

The possible values are:

- *ALL—Search all libraries on the system.
- *ALLUSR—Search all user libraries on the system except those that are owned by QSYS, or owned by a user profile that is owned by QSYS. Libraries that are not searched are listed at the end of the report.
- *LIBL—Search all libraries in the current library list.
- *USRLIBL—Search all user libraries in the current library list.
- <name>—The name of the library to search.

MAXWAIT

Specifies the maximum wait in hours to return search results when the **DBOBJCNT** command processor performs the command in parallel processing mode. To select the parallel processing mode, set the iCluster system value MAXNUMJOBS ('Max. # of parallel processes') to a value greater than 0.

The possible values are:

- *NOMAX—Specifies that there is no maximum length of time for the **DBOBJCNT** command processor to perform the command in parallel processing mode. By default, this value is used.
- <maximum-wait-time>—Sets the maximum length of time, in hours, the **DBOBJCNT** command processor performs the command in parallel processing mode. If processing exceeds this time limit, command processing stops, and no report is generated.

OUTPUT

Specifies where to write the report output.

The possible values are:

- *—Displays the report output on the console and writes it to a spooled file when the command is run in interactive mode. Writes the report output only to a spooled file when the command is run in batch mode.
- *PRINT—Writes the report output only to a spooled file.

Example

```
DMOJCNT LIB(MYLIB)
```

The output of the report is written to the console and the **DMOJCNT** spooled file.

Restrictions

None.

Minimum Authority Level

*USER

Menu Access

Main Menu - F22 (Shift + F10) - Option **9 - iCluster Reports** menu - Option **12**

DMAUDITRPT (Audit Report command)

Generates a report that lists the discrepancies between the user objects on the current machine as the backup node, and the primary node specified on the command.

```
DMAUDITRPT PRIMNODE( ) CHECKIFS( ) MAXWAIT( ) RESULTFILE( ) RESULTLIB( )
```

The report also provides an audit reason code related to the replication status of each object in the list. If an object is in the replication scope of a group replicating from the specified primary node to the current node, or if it is deliberately excluded from replication by a group that is replicating between the two nodes, the group to which the object is selected will be listed in the audit report.



Note: Your IBM i user profile must be enrolled in the system directory in order to use the DMAUDITRPT command for DLO folders, and it is recommended that your user profile has *ALLOBJ authority in order to have access to all libraries and directories on the system.

This command must be invoked on a node in the cluster, as long as this node is the backup node for all groups that replicate from the specified primary node to the node where the command is invoked.

Input Parameters

PRIMNODE

Specifies the primary node that will be checked for this audit report. The backup node for the audit report is always the current note.

CHECKIFS

Specifies whether or not objects in the Integrated File System (IFS) and the Document Library Object (DLO) system will be audited between the two nodes.

The possible values are:

- *YES—The audit report will audit user objects in the IFS and the DLO. This is the default.
- *NO—The audit report will not audit objects in the IFS or DLO systems on the specified nodes.

MAXWAIT

Specifies the maximum length of time, in hours, that the DMAUDITRPT command processor will wait to receive the list(s) of objects from the primary node, before beginning the comparison of the object lists retrieved from the two nodes.

The possible values are:

- *NOMAX—Specifies that there is no maximum length of time for the DMAUDITRPT command processor to wait for the object lists from the primary node. The command processor will wait forever, or until it is ended by another job. By default, this value is used.
- <maximum-wait-time>—Specified the maximum length of time, in hours, the DMAUDITRPT command processor will wait for the object list(s) to be retrieved from the primary node. If this time is exceeded, no report will be generated.

RESULTFILE

Specifies the name of the file that will contain the results of the audit. The default is DMAUDITRPT.

RESULTLIB

Specifies the name of the library that will contain the results of the audit. The library must exist when the command is invoked.

OUTPUT

Indicates whether the audit report is displayed on the screen or written to a spooled file for printing.

The possible values are:

- *—Displays the audit report on the console and writes it to a spooled file when the DMAUDITRPT command is run in an interactive job. When the DMAUDITRPT command is run in a batch job, the output is only written to the spooled file. By default, this value is used.
- *PRINT—Writes the audit report to a spooled file for printing but does not display it on the screen.

Example

```
DMAUDITRPT PRIMNODE(TORONTO01) CHECKIFS(*YES) MAXWAIT(1) RESULTFILE(DMAUDITRPT)
```

```
RESULTLIB(REPORTLIB)
```



Note: Generating the audit report, especially if the Integrated File System is audited, can take a long time. It is recommended that this command be run in a submitted job.

DMSUSOBRPT (Suspended Object Report command)

Use on any node to generate a report that lists the objects that are currently in suspended state in your iCluster replication configuration.

```
DMSUSOBRPT LISTBSF( ) RESULTFILE( ) RESULTLIB( ) OUTPUT( )
```

This command can be issued on any node in the cluster.

Input Parameters

LISTBSF

Specifies whether or not objects in the Integrated File System (IFS) and the Document Library Object (DLO) system will be listed in the suspended objects report.

The possible values are:

- *YES—The suspended objects report will include objects in the IFS and the DLO. By default, this value is used.
- *NO—The suspended objects report will not include objects in the IFS or DLO systems

RESULTFILE

Specifies the name of the file that will contain the suspended objects list. The default is DMSUSOBLST.

RESULTLIB

Specifies the name of the library that will contain the suspended objects list. The library must exist when the command is invoked.

OUTPUT

Indicates whether the suspended objects report is displayed on the screen or written to a spooled file for printing.

The possible values are:

- *—Displays the suspended objects report on the console and writes it to a spooled file when the DMSUSOBRPT command is run in an interactive job. When the DMSUSOBRPT command is run in a batch job, the output is only written to the spooled file. By default, this value is used.
- *PRINT—Writes the suspended objects report to a spooled file for printing but does not display it on the screen.

Example

```
DMSUSOBRPT LISTBSF(*YES) RESULTFILE(SUSOBRPT) RESULTLIB(MYLIB) OUTPUT(*)
```

Displays the suspended objects report, including objects in the IFS and the DLO, on the console and writes it to the SUSOBRPT database file in the MYLIB library.

Restrictions

None.

Minimum Authority Level

*USER

Menu Access

Main Menu - F22 (Shift + F10) - Option **9** - **iCluster Reports** menu - Option **9**

Cluster operation commands

Use the cluster operation commands to perform cluster operations, including switchovers.

Related information

- [DMSTRCST \(Start iCluster Processes command\)](#)
- [DMENDCST \(End iCluster Processes command\)](#)
- [DMSTRNODE \(Start Node command\)](#)
- [DMENDNODE \(End Node command\)](#)
- [DMREJOIN \(Rejoin Cluster command\)](#)

[DMSTRGRP \(Start Group command\)](#)
[DMENDGRP \(End Group command\)](#)
[DMSTRSWO \(Switchover Group command\)](#)
[DMSETPRIM \(Prepare Primary command\)](#)
[DMCHGROLE \(Change Primary command\)](#)
[DMSWTCHRDY \(Switchover Ready command\)](#)
[RTVSWCHRDY \(Retrieve Switchover Ready Status command\)](#)
[DMSTRRSS \(Start Roleswitch Simulator command\)](#)
[DMENDRSS \(End Roleswitch Simulator command\)](#)
[DMINZSWO \(Initialize System Switchover command\)](#)
[DMFNZSWO \(Finalize System Switchover command\)](#)
[DMGENEXC \(Generate Command Exceptions command\)](#)

DMSTRCST (Start iCluster Processes command)

Starts the cluster nodes and the groups that are configured to autostart. Use this command from a job scheduler or an IPL startup program.

```
DMSTRCST STRGRP( ) STRAYP( ) MSGQUEUE( ) MAXWAITTCP( )
ATTEMPTS( ) MAXWAITNOD( ) GRPSTRDLY( ) MAXWAITGRP( )
```

The **DMSTRCST** command performs the following steps:

1. If specified, waits for the TCP/IP to be active.
2. If not yet started, starts the **XDMCLUSTER** subsystem.
3. Starts the node that owns the iCluster metadata.

Note that if the current node does not own the iCluster metadata, a communications link is opened to the node that owns the iCluster metadata so that the node can be started. To open the communications link, the other system must meet the following prerequisites:

- TCP/IP is running.
 - The **XDMCLUSTER** subsystem is active.
 - The **DMCHATL** job is running in the **XDMCLUSTER** subsystem.
4. If not yet started, starts the current node.
 5. Waits for the specified time for the node to be active. If the node is not in *ACTIVE status in the specified time, the command ends.
 6. Starts the other nodes of the cluster and waits until they are active.
 7. Starts the groups that are configured to autostart.

Input Parameters

STRGRP

Specifies whether to start groups as part of the command processing. Refresh-only groups are not started.

The possible values are:

- *YES—Starts the groups that are configured to autostart if their primary and backup nodes are active. Groups that are configured to autostart have the **AUTOSTART** group parameter set to *YES. By default,

-
- this value is used.
 - *NO—Does not start groups.

STRAPY

Indicates whether the apply process for the replication groups that are being started is started on the backup node when replication begins.

The possible values are:

- *NOCHG—Preserves the last operational status of the apply jobs on the backup node. By default, this value is used.
- *YES—Starts the apply jobs on the backup node for the groups that are being started.
- *NO—Does not start the apply jobs on the backup node for the groups that are being started.

MSGQUEUE

Specifies the message queue that receives status and alert messages from the **DMSTRCST** command process.

The possible values are:

- *CURRENT—Sends messages to the user who issued the command.
- QSYSOPR—Sends messages to the system operator message queue. By default, this value is used.
- *NONE—Does not send status and alert messages.
- <name>—The message queue name. Prefix the message queue with the name of the library where the queue is located. For example, LIB1/MSGQ1

MAXWAITTCP

Specifies the maximum time that the **DMSTRCST** command process waits for the TCP/IP communications on the current node to become active.

The possible values are:

- 10—Wait ten minutes. By default, this value is used.
- <number>—The number of minutes.

ATTEMPTS

Specifies the number of attempts to start a node.

The possible values are:

- 1—Make one attempt. By default, this value is used.
- <number>—The number of attempts.

MAXWAITNOD

Specifies the maximum time that the **DMSTRCST** command process waits for a node to become active after the node is started.

The possible values are:

- 10—Wait 10 minutes. By default, this value is used.
- <number>—The number of minutes.

GRPSTRDLY

Specifies the time that the **DMSTRCST** command process waits to start the groups after starting the nodes. Only groups that are configured to autostart are started. This wait time ensures that the required iCluster processes

are fully active.

The possible values are:

- 30—Wait 30 seconds. By default, this value is used.
- <number>—The number of seconds.

MAXWAITGRP

Specifies the maximum time that the **DMSTRCST** command process waits for a group to become active after the group is started.

The possible values are:

- 10—Wait 10 minutes. By default, this value is used.
- <number>—The number of minutes.

Example

```
DMSTRCST STRGRP(*YES) STRAPY(*YES) MSGQUEUE(*CURRENT)
```

Starts the current node and automatically starts the groups that are configured for automatic start.

Starts the apply process on the backup node for the replication groups that are being started.

Sends the alert and status messages to the current user.

Waits for 10 minutes for the TCP/IP communications on the current node to become active.

Makes one attempt to start the node.

Waits for 10 minutes for the node to start.

Waits for 30 seconds before starting the group.

Waits for 10 minutes for the groups to become active after the groups are started.

Restrictions

None.

Minimum Authority Level

*OPERATOR

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **108**

DMENDCST (End iCluster Processes command)

Ends the active groups that are using the current node, and then ends the node and the **XDMLCLUSTER** subsystem. Use this command from a job scheduler or a CL shutdown program.

```
DMENDCST OPTION( ) DELAY( ) MSGQUEUE( ) MAXWAITGRP( )
ENDNODES( ) MAXWAITNOD( ) ENDSBS( ) WAITSBSEND( )
```

The **DMENDCST** command performs the following steps:

1. If the groups that are using the current node are in *ACTIVE status, ends the groups.
2. Waits until the groups end their operations.
3. If the current node is in *ACTIVE status, ends the current node.
4. Waits for the specified time for the node to be inactive. The command ends if the node does not change to *INACTIVE status in the specified time.
5. If the **XDMCLUSTER** subsystem is running, ends the **XDMCLUSTER** subsystem.

Input Parameters

OPTION

Specifies whether to stop replication for the group immediately or in a controlled manner.

A controlled stop completes tasks to ensure that replication ends gracefully. However, the completion of these tasks may take some time. Controlled stops are recommended when possible.

The possible values are:

- *CNTRLD—Ends replication for the group in a controlled manner. By default, this value is used.
- *IMMED—Ends replication for the group immediately. Operations might not stop as expected.
- *DRAIN—Fully drains the staging store. The group apply processes will merge the staging store entries from the audit and database journal channels. Once an apply process reaches the last entry of the journal channel with fewer entries, the merge will stop. The apply process will then continue to drain and apply the entries from the other journal channel until it is empty, and then stop. The *DRAIN option does not apply to groups started with FRCSCRAPE(*YES). Groups started with FRCSCRAPE(*YES) will be ended *CNTRLD if the *DRAIN option is specified.

DELAY

The maximum amount of time, in seconds, for replication to stop in a controlled manner without intervention.

If replication cannot be completed within the timeout period, replication is immediately stopped after the timeout period expires. This parameter applies only when the **OPTION** parameter is set to *CNTRLD.

- 60—Wait for 60 seconds. By default, this value is used.
- <time>—Specifies, in seconds, the maximum amount of time for replication to stop in a controlled manner.

MSGQUEUE

Specifies the message queue that receives status and alert messages from the **DMENDCST** command process.

The possible values are:

- *CURRENT—Sends messages to the user who issued the command.
- QSYSOPR—Sends messages to the system operator message queue. By default, this value is used.
- *NONE—Does not send status and alert messages.
- <name>—The message queue name. Prefix the message queue with the name of the library where the queue is located. For example, LIB1/MSGQ1

MAXWAITGRP

Specifies the maximum time that the **DMENDCST** command process waits for a group to end operations.

The possible values are:

- 10—Wait for 10 minutes. By default, this value is used.
- <number>—The number of minutes.

ENDNODES

Specifies whether to end nodes as part of the command processing.

The possible values are:

- *YES—Ends all nodes of the cluster as part of the command processing. By default, this value is used.
- *NO—Does not end nodes.
- *CURRENT—Ends the current node as part of the command processing. Other nodes are not ended.

MAXWAITNOD

Specifies the maximum time, in minutes, that the **DMENDCST** command process waits for a node to become *INACTIVE.

The possible values are:

- 10—Wait for 10 minutes. By default, this value is used.
- <number>—The number of minutes.

ENDSBS

Specifies whether to end the **XDMCLUSTER** subsystem as part of the command processing.

The possible values are:

- *YES—Ends the **XDMCLUSTER** subsystem.
- *NO—Does not end the **XDMCLUSTER** subsystem. By default, this value is used.

WAITSBSEND

Specifies whether to wait for the **XDMCLUSTER** subsystem to end before returning control to its caller. This parameter applies only when the value of the **ENDSBS** parameter is *YES.

The possible values are:

- *YES—Wait for the **XDMCLUSTER** subsystem to end.
- *NO—Does not wait for the **XDMCLUSTER** subsystem to end. By default, this value is used.

Example

```
DMENDCST ENDGRP(*CNTRLD)
```

Ends replication for the groups in a controlled manner.

Uses the default values for the other fields to end the iCluster processes.

Restrictions

None.

Minimum Authority Level

*OPERATOR

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **109**

DMSTRNODE (Start Node command)

Starts cluster operations at a node.

```
DMSTRNODE NODE( )
```

This command sets the status of the node to *ACTIVE.

Input Parameters

NODE

The name of the node where cluster operations are started.

Example

```
DMSTRNODE NODE(NODE1)
```

Starts cluster operations at node **NODE1**.

Sets the status of the node to *ACTIVE.

If Cluster Resource Services is the failover mechanism and **NODE1** is the backup node of an active group, cluster operations for the group are also started at this node.

Restrictions

You must issue this command on an active node in the cluster unless there are no active nodes in the cluster.



Note: If this command is invoked on different nodes when the failover mechanism is Cluster Resource Services, separate clusters are effectively activated. To ensure that only a single cluster is maintained, issue this command for each node in the cluster from a single node. The first time you invoke the command, reference the name of the single node. Then, from the single node, issue the command for each other node in the cluster.

Minimum Authority Level

*OPERATOR

Menu Access

Main Menu - Option **11**

iCluster Commands menu - F22 (Shift + F10) - Option **40**

Work With Nodes screen - Option **1**

DMENDNODE (End Node command)

Ends cluster operations at a node.

```
DMENDNODE NODE( )
```

Use this command if you intend to restart cluster operations at the node. Otherwise, later, you can use the DMRMVNODE (Remove Node) command to remove the node from the cluster when it is active.

This command sets the status of the node to *INACTIVE.

If Cluster Resource Services is the failover mechanism and the node is the backup node of a replication group that has an active status, cluster operations for the group are stopped at the node before cluster operations at the node are ended. The group still has a status of *ACTIVE even though replication operations have stopped.

- ! **Note:** Cluster operations at the node cannot be stopped if the node is a primary node in an active group.

Input Parameters

NODE

The name of the node where cluster operations are stopped.

Example

```
DMENDNODE NODE(NODE1)
```

Ends cluster operations at node **NODE1**.

Sets the status of the node to *INACTIVE.

If Cluster Resource Services is the failover mechanism and **NODE1** is the backup node of an active group, cluster operations for the group are ended at this node before cluster operations for the node are stopped.

Restrictions

You must issue this command on an active node in the cluster.

Minimum Authority Level

*OPERATOR

Menu Access

Main Menu - Option 12

iCluster Commands menu - F22 (Shift + F10) - Option **41**

Work With Nodes screen - Option 4

DMREJOIN (Rejoin Cluster command)

Restarts cluster node operations on the current machine in such a way that it rejoins the cluster of which it was previously an active member.

```
DMREJOIN NUMTRIES( ) DELAY( ) STARTNEW( )
```

It also sets the status of the node on the current machine to active if the node successfully rejoins the cluster.

If Cluster Resource Services is the failover mechanism and the current machine node is the backup node of a group that has active status, cluster group operations are also started at the current machine node.

This command requires that the current machine is inactive in the cluster and is recognized as a node by at least one active node in the cluster.

This command may take a few minutes to complete.

Input Parameters

NUMTRIES

Specifies the number of attempts that are made to rejoin an existing cluster.

The default value is 1.

DELAY

Specifies the time interval in seconds between attempts to restart cluster operations at the current node.

The default value is 10.

STARTNEW

Specifies whether a new cluster is started on the current machine if attempts to rejoin an existing cluster fail.

The possible values are:

- *YES—Start a new cluster on the current machine if attempts to rejoin an existing cluster fail. The new cluster will not be part of an existing cluster.
- *NO—Do not start a new cluster on the current machine if attempts to rejoin an existing cluster fail. The current node will remain inactive if attempts to rejoin an existing cluster fail. By default, this value is used.

Output

The status of the node should change to *ACTIVE.

Example

```
DMREJOIN NUMTRIES(5) DELAY(10) STARTNEW(*NO)
```

iCluster will attempt to restart operations, to a maximum of five times, on the current machine. The interval between retries is 10 seconds. If the attempts to rejoin an existing cluster fail, a new cluster on the current machine is not started.

Restrictions

You can issue this command only on an inactive node in the cluster.

Minimum Authority Level

*OPERATOR

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **42**

DMSTRGRP (Start Group command)

Starts cluster operations for the specified group or groups.

```
DMSTRGRP GROUP( ) PRIMNODE( ) BACKUP( ) STRAPY( ) STRRFSHGRP( ) REFRESH( )
USEMARKED( ) FRCESCRAPE( )
```

The high availability support provided by iCluster is started for replication groups and WebSphere MQ groups.

This command sets the cluster status of the group or groups to *ACTIVE when the failover mechanism is Cluster Resource Services. When a group is a replication group, the replication status of the group will become *ACTIVE when replication processes are started.

! Notes:

- The **STRAPY**, **REFRESH**, **USEMARKED**, and **FRCESCRAPE** parameters are ignored if the group is a *SWDEV group.
- The **REFRESH**, **USEMARKED**, and **FRCESCRAPE** parameters are ignored if the group is a *RFSH group.

Input Parameters

GROUP

The name of the group that has cluster operations started by this command.

The possible values are:

- <name>—Starts cluster operations for the specified group.
- *ALL—Starts all groups on all nodes, regardless of their current recovery domain.
- *PRIMNODE—Starts all groups with the primary node specified. If you specify this option, you must specify the name of the primary node in the **PRIMNODE** parameter.

PRIMNODE

Indicates the name of the primary node for the groups to be started. If a specific group is named on the GROUP parameter and a specific primary node is named on the PRIMNODE parameter, the group will only be started if the named primary node is the group's current primary node.

Specify the name of the primary node or one of the following values:

- *ALL—Allows groups with any node as primary node to be started. By default, this value is used.
- *CURRENT—Starts the group(s) specified on the GROUP parameter if their primary node is the current node where the command is executed.

BACKUP

The backup node of the groups that start cluster operations. A specific backup node can only be specified when the **GROUP** parameter is *ALL or *PRIMNODE.

The possible values are:

- <name>—The name of the backup node of the groups that start cluster operations.
- *ALL—Allows groups to start with any node as the backup node.
- *HABACKUP—Specifies that replication will only be started to the HA backup node of the group. Valid only when starting a single *HADR group.
- *DRBACKUP—Specifies that replication will only be started to the DR backup node of the group. Valid only when starting a single *HADR group.

STRAPY

Indicates whether to start the apply process for the group on the backup node when replication begins.

The possible values are:

- *NOCHG—Preserves the last operational status of the apply jobs on the backup node. By default, this value is used.
- *YES—Starts the apply jobs on the backup node for the group.
- *NO—Does not start the apply jobs on the backup node for the group.

STRRFSHGRP

Indicates whether groups of type *RFSH are to be started when *ALL or *PRIMNODE is selected for the **GROUP** parameter.

The possible values are:

- *CLUSTER—Uses the cluster system value STRRFSHGRP assigned in the DMSETSVAL command to determine whether or not to start groups of type *RFSH when *ALL or *PRIMNODE are selected for the **GROUP** parameter. By default, this value is used.

(Note that the default value of the STRRFSHGRP system value is *YES.)

- *YES—Starts groups of *RFSH type as well as all other groups that match the *ALL or *PRIMNODE setting of the **GROUP** parameter.
- *NO—Does not start groups of *RFSH type when starting other groups that match the *ALL or *PRIMNODE setting of the **GROUP** parameter.

REFRESH

Indicates whether an initial refresh of selected objects in the replication group is performed before mirroring is started.

If the **USEMARKED** parameter is set to *YES, this parameter is ignored.

The possible values are:

- *YES—Performs an initial refresh of selected objects in the group.
- *NO—Does not perform an initial refresh of selected objects in the group. By default, this value is used.
- *MQRUNSTR—Refreshes objects on the backup node for WebSphere MQ groups and is recommended for environments where WMQ transactions may be started while the group is inactive. This option is similar to *YES, except that the apply jobs will process MQ journal entries from the time the group was last ended instead of from the time you run the command. You can only use this option with WebSphere MQ groups that were previously active.

If a refresh ends unexpectedly (for example, power failure), you must restart the refresh or ensure that the objects on the primary and backup nodes are synchronized before starting mirroring. For example, perform a tape save-restore.

USEMARKED

Indicates whether mirroring is started at the marked journal positions for the group.

The possible values are:

- *YES—Starts mirroring at the journal positions that have been marked for the group through the DMMRKPOS (Mark Journal Positions) command. The primary and backup nodes must have been synchronized at the time the DMMRKPOS (Mark Journal Positions) command was issued. The **REFRESH** parameter is ignored when this parameter is set to *YES. You cannot set this parameter to *YES when starting an MQ Series group for the first time.
- *NO—Does not start replication at positions marked using the DMMRKPOS (Mark Journal Positions) command. By default, this value is used.

iCluster replication will start at either:

- Journal positions set using the DMSETPOS (Set Journal Start Position) command.
- Journal positions registered using the DMREGPOS (Register Positions) command.
- The journal entry following the last journal position received on the backup system when replication was previously stopped.



Notes:

- Specifying *YES on this parameter overwrites any starting journal positions that were previously set by the DMSETPOS (Set Journal Start Position) or DMREGPOS (Register Positions) commands.
- If the DMMRKPOS (Mark Journal Positions) command has not been invoked for the named group, and the value specified for this parameter is *YES, mirroring is started at the journal positions where it was previously ended.

FRCESCRAPE

Indicates whether the scrape process for a remote journal replication group is started when the group's primary node has failed.

The possible values are:

- *YES—Starts the scrape jobs on the backup node for the remote journal group.
- *NO—Does not start the scrape jobs on the backup node for the remote journal group. By default, this value is used.

Example 1

```
DMSTRGRP GROUP(GRP1) STRAPY(*YES) USEMARKED(*YES)
```

Mirroring will start at the journal positions that have been marked for **GRP1** through the DMMRKPOS (Mark Journal Positions) command.

Update/apply jobs on the backup node are started.

Example 2

```
DMSTRGRP GROUP(GRP2) STRAPY(*NO) REFRESH(*NO) USEMARKED(*NO)
```

An initial refresh of selected objects in group **GRP2** is not performed before mirroring is started.

Mirroring is started at the last scraped journal positions for the group **GRP2**.

The apply processes on the backup node are not started when mirroring begins.

Example 3

```
DMSTRGRP GROUP(*PRIMNODE) PRIMNODE(SYSIHQ) STRAPY(*NO) USEMARKED(*NO)
```

All groups that have node **SYSIHQ** as their primary node are started.

An initial refresh is not performed before mirroring is started.

Replication starts at the last scraped journal positions for all groups of type *REPL and *MQSERIES that have the specified primary node. A full refresh is performed for groups of type *RFSH that have the specified primary node.

The apply processes on the backup node are not started when mirroring begins.

Example 4

```
DMSTRGRP GROUP(*ALL) STRRFSHGRP(*NO)
```

All groups are started on all nodes except *RFSH groups regardless of their recovery domain.

Restrictions

You must issue this command on an active node in the cluster.

Minimum Authority Level

*OPERATOR

Menu Access

Main Menu - Option 13

DMENDGRP (End Group command)

Ends cluster operations for the specified group or groups. This command sets the cluster status of the group to *INACTIVE.

```
DMENDGRP GROUP( ) PRIMNODE( ) BACKUP( ) OPTION( ) DELAY( ) ENDAPYOPT( )
```

If replication for a group fails, use this command to end the group and return it to *INACTIVE status.

Input Parameters

GROUP

The name of the group to stop cluster operations for.

The possible values are:

- <name>—The group name.
- *ALL—Ends cluster operations for all groups on all nodes, regardless of their current recovery domain.
- *PRIMNODE—Ends cluster operations for all groups with the specified primary node. If you specify this option, you must specify the name of the primary node in the **PRIMNODE** parameter.

PRIMNODE

The name of the primary node of the groups that end cluster operations. This parameter applies only when the **GROUP** parameter is *PRIMNODE.

BACKUP

The backup node of the groups that end cluster operations. A specific backup node can only be specified when the **GROUP** parameter is *ALL or *PRIMNODE.

The possible values are:

- <name>—The name of the backup node of the groups that end cluster operations.
- *ALL—Allows groups to end with any node as the backup node.
- *HABACKUP—Specifies that replication will only be ended for the HA backup node of the group. Valid only when starting a single *HADR group.
- *DRBACKUP—Specifies that replication will only be ended for the DR backup node of the group. Valid only when starting a single *HADR group.

OPTION

Specifies whether to stop replication for the group immediately or in a controlled manner.

A controlled stop completes tasks to ensure that replication ends gracefully. However, the completion of these tasks may take some time. Controlled stops are recommended when possible.

The possible values are:

- *CNTRLRD—Ends replication for the group in a controlled manner. By default, this value is used.
- *IMMED—Ends replication for the group immediately.

DELAY

The maximum amount of time, in seconds, for replication to stop in a controlled manner without intervention.

If replication cannot be completed within the timeout period, replication is immediately stopped after the timeout period expires. This parameter applies only when the **OPTION** parameter is set to *CNTRLRD.

There are certain conditions where the apply jobs for a group might not end quickly. For example, if commitment control is used, the apply cannot end until it reaches a point where there are no open commit cycles. For a busy system, this might take a very long time to happen. Another example would be when a large refresh is taking place. It is therefore important that situations such as those mentioned be considered, and a timeout is set appropriately to ensure the backup database is in a consistent state while the apply is inactive. In general, the default setting for this parameter should be long enough, but individual circumstances might require a longer timeout.

- <time>—The maximum number of seconds for replication to stop in a controlled manner. The default value is 3600. Long running refreshes or commitment control might require a longer timeout to ensure consistency of the data on the backup system.

ENDAPYOPT

Indicates how the apply processes for the group are ended. A controlled stop is the recommended selection.

An immediate stop ends the apply processes when the command is invoked with no guarantee that the apply processes are ended on a transaction boundary. This immediate stop could lead to record-level errors in the backup apply processes if the group is restarted.

The possible values are:

- *CNTRLRD—Ends the apply processes in a controlled manner. By default, this value is used.
- *ENDAPYCTL—End the apply processes on the next transaction boundary. Transactions in the group staging store might be left to process. Equivalent to ENDAPY OPTION(*CNTRLRD).
- *ENDOPT—Ends the apply processes in the same manner as the other group operations are ended.
- *DRAIN—Fully drains the staging store. The group apply processes will merge the staging store entries from the audit and database journal channels. Once an apply process reaches the last entry of the journal channel with fewer entries, the merge will stop. The apply process will then continue to drain and apply the entries from the other journal channel until it is empty, and then stop.

The *DRAIN or *ENDAPYCTL options do not apply to groups that start with *FRCSCRAPE(*YES)*. Use either the *CNTRLRD or *ENDOPT options for groups that start with *FRCSCRAPE(*YES)*.

ENDRMTJRN

Indicates whether to end the remote journal links for the group that uses remote journaling. If a remote journal link is being used by other active groups, it will not be ended.

This parameter is used only if the **GROUP** parameter is set to a specific group name.

The possible values are:

- *YES—DMENDGRP processing ends the remote journal links.
- *NO—DMENDGRP processing does not end the remote journal links.

This is the default behavior of iCluster.

Example 1

```
DMENDGRP GROUP(GRP2) OPTION(*CNTRLRD) DELAY(120)
```

Cluster operations for group **GRP2** are stopped in a controlled manner.

If replication cannot be stopped in a controlled manner within 120 seconds (two minutes) after the command was invoked, replication in group **GRP2** is stopped immediately.

Example 2

```
DMENDGRP GROUP(*ALL) OPTION(*IMMED)
```

Cluster operations are ended immediately for all groups on all nodes.

Example 3

```
DMENDGRP GROUP(*PAYROLL) OPTION(*IMMED) ENDAPYOPT(*ENDOPT)
```

Cluster operations and apply options are ended immediately for the group **PAYROLL** on all nodes. Backup apply processes are ended the same way that other group operations are ended.

Restrictions

You must issue this command on an active node in the cluster.

Minimum Authority Level

*OPERATOR

Menu Access

Work with Groups screen - Option **14**

iCluster Commands menu - F22 (Shift + F10) - Option **44**

DMSTRSWO (Switchover Group command)

Initiates a role switch within a group, which causes a switchover that swaps the roles of the primary and backup nodes in the group.

```
DMSTRSWO GROUP( ) STRREPL( ) CHGOBJAUD( ) SWOUPRF( )
```

When this command is issued, the current primary node becomes the backup node, and the current backup node assumes the role of the primary node. In addition, the user exit programs that were specified for the group are invoked on the group's nodes.

Invoke this command on an active node in the cluster. The command can be applied only to active groups.

Notes:

- If Cluster Resource Services is the failover mechanism, the status of low-level cluster support that is provided by Cluster Resource Services for the group must be ***ACTIVE**.
- A message stating whether the role switch completed successfully is placed into the event log on



all nodes of the cluster.

Input Parameters

GROUP

The name of the cluster resource group where the roles of the primary and backup nodes in the group will be reversed.

The possible values are:

- <name>—Specifies the group name. The group must be active.
- *ALL—Switches the roles of primary and backup nodes in all active groups.

STRREPL

Indicates whether to restart replication after a switchover has completed.

The possible values are:

- *YES—Restarts replication processes for the group from the new primary node to the new backup node when switchover processing is complete. By default, this value is used.
- *NO—Does not restart replication processes for the group from the new primary node to the new backup node when switchover processing is complete. If this value is specified, the DMSTRGRP (iCluster Start Group) command can be used to restart replication later. The Use marked journal positions (**USEMARKED**) parameter must be set to *YES when invoking the DMSTRGRP (iCluster Start Group) command for the first time after a switchover.

CHGOBJAUD

Specifies whether to change the object auditing value to *CHANGE for all native and BSF objects in the replication scope of the group(s) or resilient application on the new primary node after roleswitch has taken place. iCluster requires the changes of all replicated objects to be audited on the primary node. However, if you are certain that the objects on the node that will become the primary node (currently the backup node) already have object auditing value *CHANGE or *ALL, you can specify CHGOBJAUD(*NO) to speed up the roleswitch processing.



Note: This parameter is not used for a group whose CHKJRN (Check journaling at roleswitch) parameter is *NO. When the group's CHKJRN parameter is *NO, no object auditing or journaling is done during a roleswitch.

The possible values are:

- *YES—Changes the object auditing value of all objects in the replication scope of the group(s) to *CHANGE on the new primary node. This is the default.
- *NO—Does not change the object auditing value of the objects in the replication scope.

SWOUPRF

Specifies whether to switch user profiles statuses on old primary node and backup node during switch over process.

The possible values are:

- *YES—Switches user profile statuses on old primary node and old backup node.

- *NO—Does not switch user profile statuses on old primary node and old backup node.

Example

```
DMSTRSWO GROUP(GRP1) STRREPL(*YES)
```

Switches the roles of primary and backup nodes in the active group **GRP1**.

With the **STRREPL** parameter set to *YES, replication processing is restarted from the new primary node to the new backup node when switchover processing is complete.

Any user exit program that is specified for group **GRP1** through the DMADDGRP (Add Group) or DMCHGGRP (Change Group) commands is invoked on the group's nodes.

```
DMSTRSWO GROUP(*ALL) STRREPL(*NO)
```

Switches the roles of primary and backup nodes in all active groups.

With the **STRREPL** parameter set to *NO, replication processing will not be restarted from the new primary node to the new backup node when switchover processing is complete.

User exit programs that were specified for each group through the DMADDGRP (Add Group) or DMCHGGRP (Change Group) commands are invoked on the group's nodes.

Restrictions

Issue this command on an active node in the cluster. The command can be applied only to active groups.

Minimum Authority Level

*ADMIN

Menu Access

Main Menu - Option 15

iCluster Commands menu - F22 (Shift + F10) - Option **45**

Work With Groups screen - Option **20**

DMSETPRIM (Prepare Primary command)

Performs the necessary tasks to prepare a primary node to be the source node for replication after a failover of the previous primary node. This is available only when the failover mechanism is Cluster Resource Services.

```
DMSETPRIM GROUP( )
```

After preparation is complete, the node can be the source node for replication for the group.

This command is intended for groups that do not have automatic role switch mechanism enabled. An automatic role switch is enabled or disabled by setting the ROLESWITCH parameter in the DMADDGRP (Add Group) or DMCHGGRP (Change Group) commands.

A status message of the complete role switch is generated in the event log on all nodes of the cluster.

This command is available only when the **Use Cluster Resource Services** system value is *YES.

Input Parameters

GROUP

The name of the replication group that will have its primary node prepared to be the source node for replication.

Example

```
DMSETPRIM GROUP(GRP1)
```

After a failover, the primary node for the **GRP1** replication group is prepared to be the active node for replication.

If a user exit program was specified for group **GRP1** through the DMADDGRP (Add Group) or DMCHGGRP (Change Group) commands, the user exit program is invoked on the new primary node.

Restrictions

You must issue this command on an active node in the cluster.

Minimum Authority Level

*OPERATOR

Menu Access

This command is not available from the iCluster menus.

DMCHGROLE (Change Primary command)

Changes the primary node of a group to the current backup node.

```
DMCHGROLE GROUP( ) PRIMNODE( ) CHGOBJAUD( ) RESUME( ) SWOUPRF( )
```

The groups remain *INACTIVE until they are started with the DMSTRGRP (iCluster Start Group) or DMSTRAPP (iCluster Start Application) commands.

The current backup is prepared to become the primary node similar to when a failover happens when the group is active. A message (CSI1314) that states whether the role switch completed successfully is placed in the event log on all nodes of the cluster.

Input Parameters

GROUP

The name of the group. If you specify a group, the group has to be in *INACTIVE status.

The possible values are:

- <name>—Specifies the group.

- *ALL—Performs a role switch for all groups in the cluster. Changes the primary node of all groups in the cluster to the current first backup node.
- *PRIMNODE—Performs a role switch for all groups whose primary node is specified on the PRIMNODE parameter. If you specify this option, you must specify the name of the primary node in the **PRIMNODE** parameter.

PRIMNODE

Indicates the name of the primary node of the groups for which you are performing a role switch. This parameter is applicable only when the **GROUP** parameter is set to *PRIMNODE.

CHGOBJAUD

Specifies whether to change the object auditing value to *CHANGE for all native and BSF objects in the replication scope of the group(s) or resilient application on the new primary node after role switch has taken place. iCluster requires the changes of all replicated objects to be audited on the primary node. However, if you are certain that the objects on the node that will become the primary node (currently the backup node) already have object auditing value *CHANGE or *ALL, you can specify CHGOBJAUD(*NO) to speed up the role switch processing.

- ! **Note:** This parameter is not used for a group whose CHKJRN (Check journaling at role switch) parameter is *NO. When the group's CHKJRN parameter is *NO, no object auditing or journaling is done during a role switch.

The possible values are:

- *YES—Changes the object auditing value of all objects in the replication scope of the group(s) to *CHANGE on the new primary node. This is the default.
- *NO—Does not change the object auditing value of the objects in the replication scope.

RESUME

Specifies whether to resume the role switch processing when the previous role switch processing failed at the *CONSTR or *CHKIDCOL group status.

- The RESUME parameter cannot resolve role switch errors automatically. Before using this parameter, to resolve all role switch errors such as **HAS0314**, refer to the message description in detail at the event log or job log. If the role switch errors are not resolved and this parameter is used, an error such as a data inconsistency problem might remain after the role switch finishes.
- This parameter takes effect only when the current group status is *CONSTR or *CHKIDCOL.
- If the DMCHGROLE/GROUP parameter is not for a single group, RESUME will perform on multiple groups if the groups meet the requirement.
- Use the same 'Change object auditing (CHGOBJAUD)' parameter from the previous role switch command.
- If the previous role switch ran by command DMSTRSWO, the 'Re-start replication (STRREPL)' parameter of the DMSTRSWO command will be ignored because DMCHGROLE does not have this parameter.

Possible values for RESUME include:

- *YES—Specifies that role switch processing resumes.
- *NO—Specifies that role switch processing does not resume. Instead, a new role switch processing starts from the beginning phase.

SWOUPRF

Specifies whether to switch user profile statuses on old primary node and backup node during switch over process.

The possible values are:

- *YES—Switches user profile statuses on old primary node and old backup node.
- *NO—Does not switch user profile statuses on old primary node and old backup node.

Example

```
DMCHGROLE GROUP(GROUP1)
```

Changes the primary node to the backup node and the backup node to the primary node for **GROUP1**. The group must have a status of *INACTIVE.

```
DMCHGROLE GROUP(*ALL)
```

Changes the primary node to the backup node and the backup node to the primary node for all groups in the cluster. The groups must have a status of *INACTIVE.

Restrictions

Invoke the DMCHGROLE (iCluster Change Primary Node)

Minimum Authority Level

*ADMIN

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **46**

Work With Groups screen - Option **20**

DMSWTCHRDY (Switchover Ready command)

Use on an active node to determine the readiness of a group for a switchover or a role swap.

```
DMSWTCHRDY ICGROUP( ) ALWOOS( ) ALWSUS( )
```

The readiness check verifies the following conditions:

- There are no suspended or out-of-sync objects
- The replication latency does not exceed the user-defined thresholds
- There are no open commitment control cycles

If all of these conditions are satisfied, the group is ready for a switchover. As an option, suspended or out-of-sync objects can be ignored if they are not critical to a switchover or a role swap. You can use the RTVSWCHRDY command in a CL program to retrieve the switchover readiness for a group.

Switchover is not supported for the following types of groups:

- Groups that are refresh-only.
- Groups that are local loopback.
- Groups that have a backup node library other than *PRIMARY.

- Groups that have an object specifier whose library is not *GROUP or *PRIMARY.

Input Parameters

ICGROUP

Specifies the name of the replication group. The group can be *ACTIVE or *INACTIVE.

- <name>—Specifies the group name. This is a required parameter.

ALWOOS

Specifies whether out-of-sync objects should be allowed for the specified group. This is a required parameter.

The possible values are:

- *NO—Do not allow out-of-sync objects. If out-of-sync objects are detected, the group is not ready for a switchover. By default, this value is used.
- *YES—Allow out-of-sync objects.

ALWSUS

Specifies whether suspended objects should be allowed for the specified group. This is a required parameter.

The possible values are:

- *NO—Do not allow suspended objects. If suspended objects are detected, the group is not ready for a switchover. By default, this value is used.
- *YES—Allow suspended objects.

Example

```
DMSWTCHRDY ICGROUP(PAYROLL) ALWOOS(*NO) ALWSUS(*YES)
```

The DMSWTCHRDY verifies the following conditions for the group PAYROLL:

- There are no out-of-sync objects
- Suspended objects are allowed
- The replication latency does not exceed the user-defined thresholds
- There are no open commitment control cycles

Restrictions

None.

Minimum Authority Level

None.

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **47**

RTVSWCHRDY (Retrieve Switchover Ready Status command)

In a CL program, use on an active node to determine the readiness of a group for a switchover or a role swap.

```
RTVSWCHRDY ICGROUP( ) RDYSTATUS( ) STATUSINFO( ) ALWOOS( ) ALWSUS( )
```

The readiness check verifies the following conditions:

- There are no suspended or out-of-sync objects
- The replication latency does not exceed the user-defined thresholds
- There are no open commitment control cycles

If all of these conditions are satisfied, the group is ready for a switchover. As an option, suspended or out-of-sync objects can be ignored if they are not critical to a switchover or a role swap.

Failover and switchover is not supported for the following types of groups:

- Groups that are refresh-only.
- Groups that are local loopback.
- Groups that have a backup node library other than *PRIMARY.
- Groups that have an object specifier whose library is not *GROUP or *PRIMARY.

Input Parameters

ICGROUP

Specifies the name of the replication group. The group can be *ACTIVE or *INACTIVE.

- <name>—Specifies the group name. This is a required parameter.

RDYSTATUS

The variable to retrieve the ready status result for the specified group. This is a required parameter and must be a variable name of TYPE(*CHAR) LEN(1). The possible values returned are:

- 0—The group is ready for a switchover.
- 1—The group is not valid for a roleswitch. Additional details might be available in parameter **STATUSINFO**.
- 2—The group has suspended objects.
- 3—The group has out-of-sync objects.
- 4—The latency exceeded the user-specified thresholds.
- 5—Open commitment control cycles are present on the backup node.

STATUSINFO

The variable to retrieve the additional status information when the RDYSTATUS variable is returned with a value of 1. This is a required parameter and must be a variable name of TYPE(*CHAR) LEN(1). The possible values returned are:

- 0—No additional information is available.
- A—Not all group information could be retrieved. Look for previously listed messages in the joblog.
- B—The group uses local loop-back replication.
- C—The group or one of its object specifiers has a target library that is not set to *PRIMARY.
- D—The group is a refresh-only group.

ALWOOS

Specifies whether out-of-sync objects should be allowed for the specified group specified. This is a required

parameter.

The possible values are:

- *NO—Do not allow out-of-sync objects. If out-of-sync objects are detected, the group is not ready for a switchover. By default, this value is used.
- *YES—Allow out-of-sync objects.

ALWSUS

Specifies whether suspended objects should be allowed for the specified group. This is a required parameter.

The possible values are:

- *NO—Do not allow suspended objects. If suspended objects are detected, the group is not ready for a switchover. By default, this value is used.
- *YES—Allow suspended objects.

Example

```
RTVSWCHRDY ICGROUP(PAYROLL) RDYSTATUS(&GROUPSTAT) STATUSINFO(&ADDINFO)
ALWOOS(*NO) ALWSUS(*YES)
```

The RTVSWCHRDY verifies the following conditions for the group PAYROLL:

- There are no out-of-sync objects.
- Suspended objects are allowed.
- The replication latency does not exceed the user-defined thresholds.
- There are no open commitment control cycles.
- A status value is returned in CL variable &GROUPSTAT.
- If &GROUPSTAT returns a value of 1, additional status information is returned in CL variable &ADDINFO.

Restrictions

This command must be run from a CL program. The command is not available from the command line.

Minimum Authority Level

None.

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **48**

DMSTRRSS (Start Roleswitch Simulator command)

Use on a backup node to start a roleswitch simulation.

```
DMSTRRSS GROUP( ) ALWOOS( ) ALWSUS( ) ALWJRNERR( )
```

The roleswitch simulation ends all replication apply processes on the node and enables triggers and constraints so that this node can temporarily simulate the primary node for the roleswitch simulation without disrupting operations on the primary production node. After the roleswitch simulation is complete, run the DMENDRSS (End Roleswitch Simulator) command.

The following conditions must be met before the roleswitch simulation can start:

- Groups must be fully active
- Group type must be replication or MQSeries (Refresh-only and groups with backup node library redirection are not eligible)
- Group must have **JRNBACKUP** (Journal objects on backup) parameter set to *YES
- If group is *HADR, roleswitch simulation can only be done on the HA backup node, not the DR backup node.

The status of the groups that are prepared for the roleswitch simulation changes to *RSS for the duration of the roleswitch simulation. Groups whose apply processes are ended but are not part of the roleswitch simulation have the status *ENDAPY for the duration of the roleswitch simulation.

Input Parameters

GROUP

Specifies the group for roleswitch simulation. This is a required parameter.

- *BACKUP—Specifies all of the eligible groups that replicate to this backup node for a roleswitch simulation. By default, this value is used.
- <name>—Specifies the group name.

ALWOOS

Specifies whether out-of-sync objects should be allowed for the specified group. This is a required parameter.

The possible values are:

- *NO—Does not allow groups that are involved in the roleswitch simulation to have out-of-sync objects. By default, this value is used.
- *YES—Allows groups that are involved in the roleswitch simulation to have out-of-sync objects..

ALWSUS

Specifies whether suspended objects should be allowed for the specified group. This is a required parameter.

The possible values are:

- *NO—Does not allow groups that are involved in the roleswitch simulation to have suspended objects. By default, this value is used.
- *YES—Allow groups that are involved in the roleswitch simulation to have suspended objects.

ALWJRNERR

Specifies whether to allow journaling errors when the roleswitch simulation changes database file journaling to use both images on the files and data areas that are replicated by the groups involved in the roleswitch simulation.

The possible values are:

- *NO—Does not allow errors during journal changes. By default, this value is used.
- *YES—Allow errors during journal changes. The roleswitch simulation proceeds even when errors occur when journaling for database files and data areas is changed from *AFTER image to *BOTH images.

Example

```
DMSTRSS GROUP(PAYROLL) ALWOOS(*NO) ALWSUS(*YES) ALWJRNERR(*NO)
```

Runs the roleswitch simulation for the group **PAYROLL** and allows roleswitch simulation to proceed even if suspended objects are in the group.

Restrictions

None.

Minimum Authority Level

*ADMIN

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **49**

DMENDRSS (End Roleswitch Simulator command)

Use on a backup node to end a roleswitch simulation after you have run the DMSTRSS (Start Roleswitch Simulator) command and the roleswitch simulation is completed.

```
DMENDRSS ROLLBACK( ) SYNCCHK( )
```

Input Parameters

The command restores replication processes, database files, and objects to the state that they were in before the roleswitch simulation.

ROLLBACK

Specifies how to perform the rollback of operations that were recorded in database journals during the roleswitch simulation.

The possible values are:

- *OBJECT—Perform database journal rollback only for objects that are in replication scope for the groups in *RSS state.

Ending the roleswitch simulation with this option takes longer because each object's changes are rolled back individually. The database processes the journal receivers multiple times, once for each object.

Use this option if changes to objects that are not in replication scope, but reside in a library whose objects are involved in the role switch simulation, might take place during a roleswitch simulation.

By default, this value is used.

- *FAST—Perform database journal rollback at the library level. All changes reported to a journal for each library are rolled back at the library level for the period of the roleswitch simulation.

As a result, changes rolled back are processed by the database faster.

Do not use this option if *FILE or *DTAARA objects that are not in replication scope, but reside in a library that includes objects that are in replication scope, are changed during the roleswitch simulation. These changes are rolled back during the End Roleswitch Simulator processing.

SYNCHK

Specifies to run a sync check at the end of command processing. This is a required parameter.

The possible values are:

- *NO—Does not run a sync check. By default, this value is used.
- *YES—Runs a *FULL sync check when the command processing is complete. The REPAIR options *JRN and *AUTH are specified to correct any journaling or authority mismatches in the objects on the backup node.

Example

```
DMENDRSS SYNCHK(*YES)
```

Ends the switchover simulation and runs a *FULL sync check when the command processing is complete. The REPAIR options *JRN and *AUTH are specified to correct any journaling or authority mismatches in the objects on the backup node.

Restrictions

This command can be run only on the backup node after a roleswitch simulation has been started with the DMSTRRSS (Start Roleswitch Simulator) command. The following changes, if done during a roleswitch simulation, cannot be rolled back:

- ALTER TABLE
- Changes to *DTAQ objects
- CLRPFM of a physical database file (object type *FILE with attribute PF-DTA)
- Jounaled BSF content changes
- Changes to SQL objects

In addition, commitment cycles that are opened during the roleswitch simulation must be closed, otherwise they cannot be rolled back.

Minimum Authority Level

*ADMIN

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **50**

DMINZSWO (Initialize System Switchover command)

Use on the primary node to prepare for a full system switchover. Run this command to input the parameter values and display the **Initialize System Switchover** menu.

```
DMINZSWO PRMNODE( ) BKPNODE( ) PRIMIFC( ) MODE( ) ENDSBS( )
ENDSBSLST( ) STRSBS( ) STRSBSLST( ) USRTASK( )
```

Using the command to prepare for a system switchover

The parameters that you enter for this command are used when you select steps or press the PF9 key from the **Initialize System Switchover** menu.

Select the steps on the **Initialize System Switchover** menu in sequential order:

1. Save and send IBM job schedule entries to current backup
2. End printer writers (ENDWTR)
3. End application subsystems
4. End QINTER subsystem and primary IP interface
5. Display iCluster primary Status Monitor (WRKHASMON)
6. End iCluster replication
7. Roll journal receivers
8. Move the ownership of the iCluster to the new primary node

Pause here and move to the current backup node (the new primary node) and enter the DMFNZSWO (Finalize System Switchover) command. Return here after the DMFNZSWO command is completed. Select the following steps in order.

9. Start new backup node and application subsystems
10. Restore IBM job schedule entries from old to new backup
11. Start printer writers (STRPRTWTR)
12. Display iCluster backup Status Monitor (WRKHATMON)

Before you use the command, ensure that the following requirements are met:

- Only 2 nodes are in the cluster.
- The nodes must be active.
- Only one node is the primary node for all of the groups that are eligible for a roleswitch in the cluster.
- Eligible groups replicate from the primary node to the backup node. Local loopback is not supported. There is no library redirection of the objects that are replicated by the groups.
- Eligible group types are replication groups or MQSeries groups. Refresh-only groups are not eligible for role switching.

The command runs in two modes: *EXEC and *TEST. First, use *TEST mode to verify the command parameters and to become familiar with the Initialize System Switchover interface. *TEST mode does not run the steps that you select on the **Initialize System Switchover** menu. Use *EXEC mode to run the steps that you select from the **Initialize System Switchover** menu.

After the **Initialize System Switchover** menu is displayed, complete steps 1-8 on the current primary node. Pause. Move to a session on the current backup node and use the DMFNZSWO (Finalize System Switchover) command to complete the system switchover to change the current backup node to become the new primary node. After the Finalize System Switchover process is completed on the new primary node, return to this session and complete the

remaining steps on this node, which is now the backup node. This command runs immediately and interactively.

Input Parameters

PRMNODE

Specifies the name of the primary node for iCluster. This is a required parameter.

- <name>—Specifies the name of the current production system and primary node of the replication groups whose node roles will be switched.

BKPNODE

Specifies the name of the backup node for iCluster. This is a required parameter.

- <name>—Specifies the name of the backup node that will take over the role of primary node.

PRIMIFC

Specifies the primary IP address that is used for connection to this system by an interactive display. This IP address must not be the same IP address that iCluster uses for replication. This is a required parameter.

- <ipaddr>—Specifies the IP address as an IPv4 or IPV6 address in the form *nnn.nnn.nnn.nnn*, where *nnn* is a decimal number from 0 - 255.

This IP interface is ended when you run the **DMINZSWO** command in *EXEC mode..

MODE

Specifies the execution mode for the **DMINZSWO** command and menu.

The possible values are:

- *TEST—Specifies to run the command in test mode to verify the command parameters. Does not run the steps that you select on the **Initialize System Switchover** menu. The menu is displayed in test mode. By default, this value is used.
- *EXEC—Specifies to run the command in execution mode to run the steps that you select from the **Initialize System Switchover** menu.

Before you run the command in *EXEC mode, issue the TFRJOB JOBQ(QCTL) command to transfer the interactive session into subsystem QCTL and end the QINTER subsystem and the primary IP interface. You can set up additional interactive sessions under the QCTL subsystem that do not use the primary IP interface (PRIMIFC), or you can use the primary console as a precaution when you run the command in *EXEC mode to monitor the command progress.

- Before the **Initialize System Switchover** menu is displayed, this command verifies that the current job is running in the QCTL subsystem and that the IP address that is used by the current iCluster node for replication is different from the Primary IP Interface that is specified on this command.

ENDSBS

Specifies the set of application subsystems to be ended when you select the **End application subsystems** step from the **Initialize System Switchover** menu. By default, all active subsystems are ended. Critical subsystems, such as QCTL, remain active during switchover. The list of subsystems that are ended is sent to the new primary node. This list is used as the list of subsystems to start on the new primary node when *PRIMARY is specified for the **STRSBS** parameter of the DMFNZSWO (Finalize System Switchover) command.

The possible values are:

- *ACTIVE—Specifies to end all active subsystems, except for QCTL, QSYSWRK, and the iCluster subsystem (usually XDMCLUSTER). By default, this value is used.
- *NONE—Does not end subsystems. The *PRIMARY list of subsystems that are ended on the primary node is created with no entries and is sent to the backup node.

- *LIST—Creates a list of up to 50 subsystems to be ended.

Note: Only expert users who can control subsystem activity without impacting the iCluster replication environment should use this option.

ENDSBSLST

Specifies the names of up to 50 subsystems to end during the switchover. This parameter is required when *LIST is specified for the **ENDSBS** parameter.

- <name>—Specifies the qualified subsystem name. You must specify the library where the subsystem description resides to allow the new primary node to start the same list of subsystems when the STRSBS(*PRIMARY) parameter is specified on the DMFNZSWO (Finalize System Switchover) command.

STRSBS

Specifies a set of subsystems to start when this node becomes the backup node, and you select the Start new backup node and application subsystems step from the **Initialize System Switchover** menu.

The possible values are:

- *BACKUP—Specifies that the subsystems that were ended on the previous backup node will be started on the new backup node. By default, this value is used.
- *PRIMARY—Specifies to start the subsystems that were ended on this system when it was the primary node.
- *NONE—Specifies that no subsystems are started during switchover.
- *LIST—Specifies a list of up to 50 subsystems to start.

Note: This option should be used only by expert users who can control subsystem activity without impacting the iCluster replication environment.

STRSBSLST

Specifies the names of up to 50 subsystems to start. This parameter is required when *LIST is specified for the **STRSBS** parameter.

- <name>—Specifies the qualified subsystem name. You must also specify the library where the user task program resides.

USR TASK

Specifies the name and library of the user task program to invoke when you press the PF9 key from the **Initialize System Switchover** menu. You can invoke the user task at any time. This is an optional parameter.

- <name>—Specifies the name and library of the user task program. You must also specify the library where the user task program resides.
- *LIBL—Specifies the library list of the current thread. All libraries in the library list are searched until the first match is found.
- *CURLIB—Specifies that the current library for the thread is used to locate the program. If no library is specified as the current library for the thread, the QGPL library is used.

Example

```
DMINZSWO PRMNODE(PRODUCTN) BKPNODE(BACKUP) PRIMIFC('155.4.5.44') MODE(*TEST)
ENDSBS(*ACTIVE) STRSBS(*PRIMARY)
```

Runs test mode for the initialize switchover for the current primary node **PRODUCTN** and the current backup node **BACKUP** (that takes over the role of primary node). IP address 155.4.5.44 is used for communications to this node and will be ended by the **DMINZSWO** command.

All active subsystems are ended, except for **QCTL**, **QSYSWRK**, and the iCluster subsystem. After this node becomes the backup node, select the **Start new backup node and application subsystems** step from the **Initialize System Switchover** menu to start the subsystems that were ended on this system when it was the primary node.

Restrictions

This command can be used only in a two-node cluster where one node is the primary node and the other node is the backup node for at least one replication group or one *MQSERIES group that is eligible for switchover. The command must be invoked on the current primary node of the groups that will be switched over. After running the first 8 steps of the **DMINZSWO** menu, you must run the **DMFNZSWO** command on the current backup node.

Minimum Authority Level

*ADMIN

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option 51

DMFNZSWO (Finalize System Switchover command)

Use on the new primary node to complete the full system switchover. Run this command to input the parameter values and display the **Finalize System Switchover** menu.

```
DMFNZSWO NEWPRMNODE( ) NEWBKPNODE( ) PRIMIFC( ) MODE( ) ENDSBS( )
ENDSBSLST( ) STRSBS( ) STRSBSLST( ) USRTASK( )
```

Using the command to complete a system switchover

After you initialize the switchover with the **DMINZSWO** (Initialize System Switchover) command and complete the required steps on the previous primary node, use this command on the new primary node.

The parameters that you enter for this command are used when you select steps or press the PF9 key from the **Finalize System Switchover** menu.

Select the steps on the **Finalize System Switchover** menu in sequential order.

1. Verify that iCluster replication has ended
2. Save and send IBM job schedule entries to the new backup node
3. End printer writers (ENDWTR)
4. End application subsystems and the primary IP interface
5. Verify that the current node owns the iCluster metadata
6. Perform roleswitch for all groups
7. Display the iCluster event log to verify roleswitch completion
8. Verify node role change for each group (DMWRKNODE)
9. Start the IP interface and application subsystems
10. Restore job schedule entries from previous to new primary

11. Start printer writers (STRPRTWTR)

Perform the remaining steps only if the new backup node (previous primary node) is operational.

12. Start replication from the new primary node to the new backup node

13. Display the iCluster primary Status Monitor (WRKHASMON)

Before you use the command, ensure that the following requirements are met:

- Steps 1 - 8 on the **Initialize System Switchover** menu are completed on the previous primary node.
- There are only 2 nodes in the cluster.
- The current node (the new primary node) must be active; the previous primary node (the new backup node) must be inactive.
- Only one node is the primary node for all of the groups that are eligible for a roleswitch in the cluster.
- Eligible groups replicate from the primary node to the backup node. Local loopback is not supported.
- Eligible group types are replication groups or MQSeries groups. Refresh-only groups are not eligible for role switching.

The command runs in two modes: *EXEC and *TEST. First, use *TEST mode to verify the command parameters and to become familiar with the Finalize System Switchover interface. *TEST mode does not run the steps that you select on the **Finalize System Switchover** menu. Use *EXEC mode to run the steps that you select from the **Finalize System Switchover** menu.

This command runs immediately and interactively.

Input Parameters

NEWPRMNODE

Specifies the name of the new primary node for the roleswitch. This node is the backup node before the roleswitch. This parameter is required.

- <name>—Specifies the name of the current backup node that will become the primary node of the replication groups whose node roles will be switched.

NEWBKPNODE

Specifies the name of the new backup node for the groups that will be switched over. This node is the primary node before the switchover. This parameter is required.

- <name>—Specifies the name of the node that will take over the role of the backup node after the system switchover is completed.

PRIMIFC

Specifies the primary IP address that is used for connection to this system by an interactive display. This IP address must not be the same IP address that iCluster uses for replication. This parameter is required.

- <ipaddr>—Specifies the IP address as an IPv4 address in the form *nnn.nnn.nnn.nnn*, where *nnn* is a decimal number from 0 - 255, or in IPv6 notation.

This IP interface becomes the primary IP address for the new primary node when you run the Start the IP interface and application subsystems step from the **Finalize System Switchover** menu in *EXEC mode. You can use the same IP address that you specified for the Primary IP interface (PRIMIFC) on the DMINZSWO command. Optionally, specify a different Primary IP interface, particularly if the two nodes are on different subnets.

-  **Note:** This interface must be inactive when you run this command to prevent users from accessing the system before the node roles are changed. The interface was ended when the DMINZSWO

- !** command was run. If the interface has been activated this command will end. End the interface before you run the command again.

MODE

Specifies the execution mode for the DMFNZSWO command and menu.

The possible values are:

- *TEST—Specifies to run the command in test mode to verify the command parameters. Does not run the steps that you select on the **Finalize System Switchover** menu. The menu is displayed in test mode.
- *EXEC—Specifies to run the command in execution mode to run the steps that you select from the **Finalize System Switchover** menu.

Before you run the command in *EXEC mode, issue the TFRJOB JOBQ(QCTL) command to transfer the interactive session into subsystem QCTL and end the QINTER subsystem and the primary IP interface. You can set up additional interactive sessions under the QCTL subsystem that do not use the primary IP interface (PRIMIFC), or you can use the primary console as a precaution when you run the command in *EXEC mode to monitor the command progress.

Before the **Finalize System Switchover** menu is displayed, this command verifies that the current job is running in the QCTL subsystem and that the IP address that is used by the current iCluster node for replication is different from the Primary IP Interface that is specified on this command.

ENDSBS

Specifies a set of application subsystems to be ended when you select the End application subsystems and primary IP interface step from the **Finalize System Switchover** menu. By default, all active subsystems are ended. Critical subsystems, such as QCTL, remain active during switchover. The list of subsystems that are ended is the *BACKUP list that is used in step 9 of the **DMINZSWO** menu. This list is sent to the new backup node and can be used to start the same subsystems there.

The possible values are:

- *ACTIVE—Specifies to end all active subsystems, except for QCTL, QSYSWRK, and the iCluster subsystem (usually XDMCLUSTER). By default, this value is used.
- *NONE—Does not end subsystems. Creates the *BACKUP list of subsystems on the primary node. Sends the *BACKUP list of subsystems with no entries to the new backup node.
- *LIST—Creates a list of up to 50 subsystems to be ended.

! **Note:** Only expert users who can control subsystem activity without impacting the iCluster replication environment should use this option.

ENDSBSLST

Specifies the names of up to 50 subsystems to be ended when you select the end application subsystems and primary IP interface step from the **Finalize System Switchover** menu. This parameter is required when *LIST is specified for the **ENDSBS** parameter.

- <name>—Specifies the qualified subsystem name. You must specify the library where the subsystem description resides. Prefix the subsystem name with the name of the library where the subsystem description is located (for example, LIB1/MYSBSD).

STRSBS

Specifies a set of subsystems to start when you select the Start IP interface and application subsystems step from the **Finalize System Switchover** menu.

The possible values are:

- *BACKUP—Specifies that the subsystems that were ended on this system when it was the backup node are started again when it becomes the new primary node.
- *PRIMARY—Specifies that the subsystems that were ended on the previous primary node are started when this system becomes the new primary node. By default, this value is used.
- *NONE—Specifies that no subsystems are started.
- *LIST—Specifies a list of up to 50 subsystems to start after this system becomes the new primary node.

! **Note:** This option should be used only by expert users who can control subsystem activity without impacting the iCluster replication environment.

STRSBSLST

Specifies the names of up to 50 subsystems to start. This parameter is required when *LIST is specified for the **STRSBS** parameter.

- <name>—Specifies the qualified subsystem name. You must also specify the library where the subsystem description resides.

USR TASK

Specifies the name and library of the user task program to invoke when you press the PF9 key from the **Finalize System Switchover** menu. You can invoke the user task at any time. This parameter is optional.

- <name>—Specifies the name and library of the user task program. You must specify the library where the user task program resides. Prefix the program with the name of the library where the user task program is located (for example, LIB1/FNZSWOTASK).
- *LIBL—Specifies the library list of the current thread. All libraries in the library list are searched until the first match is found.
- *CURLIB—Specifies that the current library for the thread is used to locate the program. If no library is specified as the current library for the thread, the QGPL library is used.

Example

```
DMFNZSWO NEWPRMNODE(BACKUP) NEWBKPNODE(PRODUCTN) PRIMIFC('155.4.5.44')
MODE(*TEST) ENDSBS(*ACTIVE) STRSBS(*PRIMARY)
```

Runs the Finalize System Switchover command in test mode for the finalize switchover for the new primary node **BACKUP** and the new backup node **PRODUCTN**. IP address 155.4.5.44 is used for communications with the current node, is currently inactive, and is activated by this command.

All active subsystems are ended, except for **QCTL**, **QSYSWRK**, and the iCluster subsystem. After this system becomes the new primary node, select the **Start IP interface and application subsystems** step from the **Finalize System Switchover** menu to start the subsystems that were ended on the previous primary node.

Restrictions

This command can be used only in a two-node cluster where one node is the primary node and the other node is the backup node for at least one replication group or *MQSERIES group that is eligible for switchover. The command must be invoked on the new primary node (previously the backup node) of the groups that are being switched over. Before running this command you must run the first 8 steps of the **DMINZSWO** menu on the previous primary node of the groups that are being switched over.

Minimum Authority Level

*ADMIN

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **52**

DMGENEXC (Generate Command Exceptions command)

Generates an exception when an iCluster command fails or results in no action.

```
DMGENEXC GENEXC( )
```

For example, if you are running iCluster commands in a user exit program, this command allows you to detect command failures by generating an exception.



Note: This command generates an exception with an ID of CSI9954. For more information on what command caused this exception, examine the previous messages in the job log or in the event log for each node in the cluster.

For example, to generate an exception for an iCluster command failure, place the call to `DMGENEXC GENEXC(*YES)` before any calls to commands in your user exit program. When iCluster encounters the CSI9954 exception, role switch processing is stopped and declared incomplete.

Input Parameters

GENEXC

Indicates whether exception generation is enabled or disabled for iCluster commands that fail or result in no action.

The possible values are:

- *NO—Does not generate exceptions for iCluster commands that fail. By default, this value is used.
- *YES—Generates exceptions for iCluster commands that fail.

Example

```
DMGENEXC GENEXC(*YES)
```

iCluster generates an exception for iCluster commands that fail or result in no action.

Restrictions

The scope of this command is limited to the job in which it is run. Exception generation is disabled when the job ends.

Minimum Authority Level

*OPERATOR

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **55**

Replication operation commands

Use the replication operation commands to work with journals, manage apply processes, activate and suspend objects and files, and perform other replication tasks.

Related information

- [DMSETPOS \(Set Journal Start Position command\)](#)
- [DMMRKPOS \(Mark Journal Positions command\)](#)
- [DSPHAPOS \(Display Journal Information command\)](#)
- [RTVHAPOS \(Retrieve iCluster Journal Position command\)](#)
- [VFYHAJRN \(Verify Audit Journal command\)](#)
- [DMSTRAPY \(Start Replication Apply Process command\)](#)
- [DMENDAPY \(End Replication Apply Process command\)](#)
- [RTVAPYSTS \(Retrieve Apply Status command\)](#)
- [DMACTOBJ \(Activate Object command\)](#)
- [DMACTBSF \(Activate BSF Object command\)](#)
- [DMACTOOS \(Activate Out-of-Sync Objects command\)](#)
- [DMACTSUS \(Activate Suspended Objects command\)](#)
- [DMACTSQLO \(Activate SQL Object command\)](#)
- [DMSUSOBJ \(Suspend Object command\)](#)
- [DMSUSBSF \(Suspend BSF Object command\)](#)
- [CRTCFGOBJ \(Create Configuration Objects command\)](#)
- [INITHAOBJ \(Initialize Objects command\)](#)
- [DMLOGENT \(Log Journal Entry command\)](#)
- [DMACTSPLF \(Activate Suspended Spooled Files command\)](#)
- [DMWRKSSPLF \(Work with Suspended Spooled Files command\)](#)
- [RTVRCVPT \(Retrieve Recovery Checkpoint command\)](#)
- [RTVRCVPTR \(Retrieve Recovery Checkpoint \(CL Program\) command\)](#)
- [SYNCHATRG \(Synchronize Triggers and SQL Triggers command\)](#)
- [DMENDRFSH \(End Current Refresh command\)](#)

DMSETPOS (Set Journal Start Position command)

Positions iCluster to start replication at a specific entry in a database journal, system audit journal, or all journals scraped by a group.

```
DMSETPOS GROUP( ) JRN( ) JRNRCV( ) JRNPOSLRG( ) STRDATE( ) STRTIME( ) JRNPOS( )  
BACKUP( )
```

The specific journal entry can be entered directly or determined by the command if a date and time is entered.

- ! **Note:** To start mirroring after performing a save of replicated objects on the primary node and

- !** restoring them on the backup node, set the journal starting positions with the DMSETPOS (Set Journal Start Position) command or the DMMRKPOS (Mark Journal Positions). The **DMMRKPOS** command ensures that objects are audited and journaled properly for replication.

If you select SQL objects into the group and want to run the **DMSETPOS** command before starting the group, you must also explicitly run the **DMSETPOS** command on journal **QYS2/QSQJRN**.

Input Parameters

GROUP

The name of a replication group. The starting entry in the journal or journals is set on the primary node in the recovery domain for the group.

The group must be inactive when this command is invoked.

JRN

The name of a database or system audit journal that has its starting position set through this command.

The possible values are:

- <name>—Specifies the name of a database journal or system audit journal.
- *ALL—Sets starting positions for all journals that are being scraped by the replication group identified in the GROUP parameter of this command.

If you specify the *ALL option, the following conditions must be true:

- The **JRNRCV** parameter is set to *CURRENT or *CURCHAIN.
- A specific date and time is specified in the STRDATE/STRTIME parameters OR the *LASTAPY or *CURSRCPOS options are specified in the **JRNPOS** parameter.

If you specify a database journal, then the database and audit journal starting positions are set for objects journaled to this journal and selected for replication.

If you specify the system audit journal, then the audit journal starting position is set for all non-journaled objects that are selected for replication, , as well as for object level operations on journaled objects selected for replication.

If you do not specify *ALL, the library where the journal resides must be identified. Prefix the journal with the name of the library where the journal is located. For example:

LIB1/JRN1

JRNRCV

The name of a database or system audit journal receiver.

The possible values are:

- <name>—Specifies the name of a journal receiver.
- *CURRENT—Specifies the current journal receiver for the journal. By default, this value is used.
- *CURCHAIN—Specifies the journal receiver that is determined from the timestamp information of the starting entry that is provided through the **STRDATE** and **STRTIME** parameters.

If you specify *CURCHAIN, you cannot specify a starting sequence number with the **JRNPOS** parameter. In this case, only the **STRDATE** and **STRTIME** parameters can be set.

If you do not use the *CURCHAIN value, then you must specify the library where the journal receiver resides.

Prefix the journal receiver with the name of the library where the journal receiver is located. For example,

LIB1/JRNRCV1

Or, with the following value:

- *LIBL—Specifies the set of libraries in your library list (the libraries are searched in order for the first occurrence of the specified journal receiver). By default, this value is used.

JRNPOSLRG

The sequence number (up to 20 digits) of the entry in the journal from which iCluster starts processing when replication resumes. Any non-blank value in this parameter takes precedence over the **JRNPOS** parameter.

You cannot use a numeric value in this parameter with **STRDATE** or **STRTIME** or when **JRNRCV** is set to *CURCHAIN.

The possible values are:

- <number>—Specifies the starting sequence number.
- *LASTAPY—Starts journal processing at the last journal entry in the source journal that was applied to the backup node. You cannot use this value for a group that has never been replicated. Instead, refresh the group when you start it for the first time.
- *CURSRCPOS—Starts journal processing at the last journal entry in the source journal.
- *MRKPOS—Sets the starting positions to the positions found in the DMMRKPOS metadata for the group. This value is only permitted when the **JRN** value is *ALL.

STRDATE

The date of the entry that iCluster will start processing from when mirroring is restarted. The date format must adhere to the journal's date format, which can be determined by displaying attributes for the attached journal receiver on the node.

If you specify a starting date, the **JRNRCV** parameter can be set to a special value (*CURCHAIN).

STRTIME

The time of the entry that iCluster will start processing from when mirroring is restarted.

This parameter is only applicable when a value is specified for the **STRDATE** parameter.

If you specify a starting time, the **JRNRCV** parameter can be set to a special value (*CURCHAIN).

JRNPOS

The sequence number (up to 10 digits) of the entry in the journal that iCluster will start processing from when replication resumes.

You cannot use a numeric value in this parameter with **STRDATE** or **STRTIME** or if **JRNRCV** is set to *CURCHAIN.

The possible values are:

- <number>—Specifies the starting sequence number.
- *LASTAPY—Starts journal processing at the last journal entry in the source journal that was applied to the backup node. You cannot use this value for a group that has never been replicated. Instead, refresh the group when you start it for the first time.
- *CURSRCPOS—Starts journal processing at the last journal entry in the source journal.
- *MRKPOS—Starts journal processing at the journal entries of all journals that were recorded by the latest DMMRKPOS operation for the group. If you specify this value, you can only specify *ALL for the journal parameter(JRN).

BACKUP

The name or type of the backup node of the group whose journal positions will be set. The values *HABACKUP and *DRBACKUP are only valid if a group of type *HADR is named on the GROUP parameter.

The possible values are:

- *ALL—Specifies that starting journal positions will be set for all backup nodes for the group specified. This is the only valid option for setting journal positions for groups that are not *HADR groups.
- *HABACKUP—Specifies that starting journal positions will be set for the group but they will apply only to the HA backup node of the group. This value is valid only when an *HADR group is specified in the GROUP parameter.
- *DRBACKUP—Specifies that starting journal positions will be set for the group but they will apply only to the DR backup nodes of the group. This option is valid only when an *HADR group is specified in the GROUP parameter.
- <name>—Specifies the single backup node for which journal positions will be set for the specified groups. This option is valid only when an *HADR group is specified in the GROUP parameter.

Example 1

```
DMSETPOS GROUP(GRP1) JRN(QSYS/QAUDJRN) JRNRCV(QSYS/JRNRCV1)
JRNPOSLRG(25689653214569)
```

The starting journal position is set for group **GRP1**.

The starting position is set for journal **QAUDJRN** in library **QSYS**, and journal receiver **JRNRCV1** in the same library.

Journal processing starts at sequence number 25689653214569 in the journal when mirroring is restarted.

Example 2

```
DMSETPOS GROUP(GRP2) JRN(QSYS/QAUDJRN) JRNRCV(*CURRENT) JRNPOSLRG(*LASTAPY)
```

The starting journal position is set for group **GRP2**.

The starting position is set for journal **QAUDJRN** in library **QSYS** and the current journal receiver.

Journal processing starts at the last journal entry in the source journal that was applied to the backup node.

Example 3

```
DMSETPOS GROUP(GRP3) JRN(QSYS/QAUDJRN) JRNRCV(*CURCHAIN) STRDATE('10/01/14')
STRTIME('12:45:00')
```

The starting journal position is set for group **GRP3**.

The starting position is set for journal **QAUDJRN** in library **QSYS**.

The journal receiver is determined from the starting time and date information specified through the **STRDATE** and **STRTIME** parameters.

Journal processing starts at the entry that is time stamped October 1, 2014 at 12:45 p.m. in all journals when mirroring is restarted.

Example 4

```
DMSETPOS GROUP(GRP4) JRN(*ALL) JRNRCV(*CURCHAIN) STRDATE('10/01/14')
STRTIME('12:45:00')
```

The starting journal position is set for group **GRP4**.

The starting position is set for all journals scraped by the group **GRP4**.

The journal receiver is determined from the starting time and date information specified through the **STRDATE** and **STRTIME** parameters.

When mirroring is restarted, journal processing starts at the entry that is timestamped October 1, 2014 at 12:45 p.m. in all journals.

Restrictions

You must issue this command on an active node in the cluster when the replication group is inactive.

Minimum Authority Level

*OPERATOR

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **60**

DMMRKPOS (Mark Journal Positions command)

Establishes the starting positions for mirroring for a replication group.

```
DMMRKPOS GROUP( ) PRIMNODE( ) CHGOBJAUD( ) SAVOBJ( ) SAVFLIB( ) TGTRLS( ) BACKUP( )
```

```
SAVPVTAUT( )
```

The journal positions are recorded in the iCluster metadata on the primary node that is associated with the group. Marking a journal position takes a snapshot of the state of the primary node and the objects that match the object specifiers at the time the command is issued.

The marked positions are used in the DMSTRGRP (iCluster Start Group) and DMSTRAPP (iCluster Start Application) commands. This command is typically used to record starting points before starting replication. When using the DMSTRGRP command, you cannot start mirroring at the marked journal positions for the group when starting an MQ Series group for the first time.

This command journals objects that have not already been journaled that match object specifiers for the group to the default journal for the group. To journal objects to a journal other than the default journal, it is your responsibility to start journaling for the objects before you run this command.

This command optionally changes the object auditing value of all objects in replication scope on the primary node to *CHANGE so that changes to objects can be captured. iCluster requires that this value for each object be either *CHANGE or *ALL.

The command overwrites the previous results each time this command is issued.

Input Parameters

GROUP

The name of an existing replication group.

Specify the name of a group or resilient application, or one of the following values:

- *ALL—Marks the journal positions of all groups on all nodes, regardless of their current recovery domain.
- *PRIMNODE—Marks the journal positions of all groups with the primary node specified in the PRIMNODE parameter. If you specify this option, you must specify the name of the primary node in the PRIMNODE parameter in this command.

PRIMNODE

Indicates the name of the primary node for the groups being processed. This parameter is only applicable when the GROUP parameter in this command is set to *PRIMNODE.

CHGOBJAUD

Specifies whether to change the object auditing value to *CHANGE for all native and BSF objects in the replication scope of the group or resilient application. iCluster requires that all changes of the replicated objects be audited on the primary node. However, if you are certain that the objects already have object auditing value *CHANGE or *ALL, you can specify CHGOBJAUD(*NO) to speed up the DMMRKPOS processing.

The possible values are:

- *YES—Changes the object auditing value of all objects in the replication scope. By default, this value is used.
- *NO—Does not change the object auditing value of the objects in the replication scope.

SAVOBJ

Specifies whether to save objects in save files.

Note: This parameter only applies if you specify a specific group. It does not apply if you specify *ALL or *PRIMNODE on the **GROUP** parameter.

- Use this parameter if you wish to save the objects that are replicated by the group into one or more save files as part of this command. You are responsible for transporting the save files to the backup node and restoring the objects before starting replication.
- The save files are saved in the library that is specified by the **SAVFLIB** parameter on the primary node.
- The save processing uses the Object compression product system value that is configured with the DMSETVAL command.

The possible values are:

- *NO—Does not save the objects in the group's replication scope. By default, this value is used.
- *YES—Saves the objects in the group's replication scope.

The following naming conventions apply to the save files:

- Native objects are saved in save files with the same names of the libraries they are in.
- Non-DLO BSF objects are saved in a save file with the name **BSFSAVF**.
- DLO BSF objects are saved in a save file with the name **DLOSAVF**.

The save files are not saved if the combined length of the save file names, or the names of the excluded DLO folders, is more than 5000 characters.

SAVFLIB

Specifies the library name on the primary node. The save files are created in this library when the **SAVOBJ** parameter is *YES on the primary node.

The possible values are:

- *NONE—Specifies no library. By default, this value is used.
- <name>—The library name where the save files are created. Required when **SAVOBJ** *YES.

TGTRLS

Specifies the target release of the save files when the **SAVOBJ** parameter is *YES on the primary node. The target release is required so that the objects in the save files can be restored on the backup node. You cannot enter an operating system release that is later than the operating system release on the primary node.

The possible values are:

- *CURRENT—Specifies the target release of the save files is the same as the operating system release on the primary node. By default, this value is used.
- <charval>— If the backup node is at an earlier release of the operating system than the primary node, specify the OS release of the backup node. The operating system release in format VxRyMz, where:
 - Vx is the version
 - Ry is the release
 - Mz is the modification

SAVPVTAUT

Specifies whether to save private authorities when saving the native and non-DLO BSF objects in the group's replication scope. Saving private authorities will increase the amount of time it takes to save the objects, but it can simplify the synchronization of the objects by restoring the private authorities on the backup node when the objects themselves are restored.

The possible values are:

- *NO—No private authorities are saved. By default, this value is used.
- *YES—Private authorities are saved for each native or non-DLO BSF object that is saved. Note that Document Library objects (DLO) can only be saved without their private authorities, even when *YES is specified for the **SAVPVTAUT** parameter.

You must have save system (*SAVSYS) or all object (*ALLOBJ) special authority to specify this value.

If you specify *YES for the parameter, specify PVTAUT(*YES) in the RSTOBJ or RST command when you restore the objects on the backup node of the group.

If the save target release (TGTRLS) is lower than the OS release of the primary node, the objects in the group's replication scope will be saved without private authorities, even if you specify SAVPVTAUT(*YES).

If you specify *NO for the SAVPVTAUT parameter, you can run a sync check for the group with SCTYPE(*FULL) REPAIR(*AUTH) to refresh the private authorities of the objects on the backup node, after the group is started from the marked journal positions.

BACKUP

The name or type of the backup node of the group whose journal positions will be marked. The values *HABACKUP and *DRBACKUP are only valid if a group of type *HADR is named on the GROUP parameter.

The possible values are:

- *ALL—Specifies that starting journal positions will be marked for all backup nodes for the group specified. This is the only valid option for marking journal positions for multiple groups and for groups that are not

- *HADR groups.
- *HABACKUP—Specifies that starting journal positions will be marked for the group but they will apply only to the HA backup node of the group. This value is valid only when an *HADR group is specified in the GROUP parameter.
- *DRBACKUP—Specifies that starting journal positions will be marked for the group but they will apply only to the DR backup node of the group. This option is valid only when an *HADR group is specified in the GROUP parameter.
- <name>—Specifies the single backup node for which journal positions will be marked for the specified group. This option is valid only when an *HADR group is specified in the GROUP parameter.

Example 1

```
DMMRKPOS GROUP(GRP1)
```

Marks the current positions for journals that are used by group **GRP1** and changes the object auditing value to *CHANGE for all native and BSF objects in the replication scope of the group or resilient application. Validates that all objects that are replicated by the group are audited and journaled where necessary.

Example 2

```
DMMRKPOS GROUP(*PRIMNODE) PRIMNODE(MYPROD) CHGOBJAUD(*NO) SAVOBJ(*YES)
```

```
SAVFLIB(MYSAVLIB) TGTRL(*CURRENT)
```

Marks the current position of all groups using node MYPROD as their primary node. Object auditing is not changed as all objects are currently set properly (the group already exists). All objects will be saved into a savefile in library MYSAVLIB at the current release.

This savefile can now be sent to the backup node and restored to synchronize the group quickly.

Restrictions

You must issue this command on an active node in the cluster when the group is inactive.

Minimum Authority Level

*OPERATOR

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **61**

DSPHAPOS (Display Journal Information command)

Displays the earliest journal position that was fully processed for all replication groups that use a specified journal.

```
DSPHAPOS JOURNAL( )
```

The journal position is displayed including the journal sequence number, journal receiver and library. The journal

position also displays the replication group that is processing the journal at this earliest position.

Use this command to determine whether a journal receiver that is associated with a journal is being used by iCluster, or to find out which receivers can be deleted. Receivers in the chain before the returned journal receiver have been fully processed and can be deleted.

Input Parameters

JOURNAL

The name of a journal being used by iCluster.

Identify the library where the journal resides. Prefix the journal with the name of the library where the journal is located. For example:

LIB2/JRN1

Example

```
DSPHAPOS JOURNAL(LIB1/JRN1)
```

This command displays the group, the earliest unprocessed journal sequence number, the journal receiver name and the journal receiver library for **JRN1** (if it exists) in library **LIB1**.

The earliest journal position includes the journal sequence number, journal receiver and journal receiver library, of journal **JRN1** in library **LIB1**.

Restrictions

You must issue this command on the node where the journal resides. The node does not have to be active in the cluster.

Minimum Authority Level

Any user with access to the iCluster product library.

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **62**

RTVHAPOS (Retrieve iCluster Journal Position command)

Returns the journal receiver, the library of the journal receiver, the sequence number of the last fully processed entry for a specified journal, and the name of the group processing that entry.

```
RTVHAPOS JOURNAL( ) TARGET( ) GROUP( ) JRNENTLRG( ) JRNENTRY( ) JRNRCVNME( )
JRNRCLIB( ) RESULT( ) SINGLEGRP( ) GRPNAME( )
```

The backup node and replication group that are associated with the journal entry are also returned.

Use this command to determine whether a journal receiver that is associated with a journal is being used by iCluster. Any receivers in the chain prior to the returned journal receiver have been fully processed and can be deleted.

You can optionally retrieve the last applied journal position for a single group and a specific journal.

- ! **Note:** Since this command uses return variables, it must be executed from a CL program. Use the DSPHAPOS (Display Journal Information) command to obtain the same information from the command line.

Input and Output Parameters

JOURNAL

The name of a journal. This is an input parameter.

Identify the library where the journal resides. Prefix the journal with the name of the library where the journal is located. For example:

LIB2/JRN1

TARGET

The name of a 10-character CL variable into which the name of the backup node for the group associated with the last fully processed entry for the journal is copied. This is an output parameter.

GROUP

The name of a 10-character CL variable into which the name of the replication group associated with the last fully processed journal entry for the journal is copied. This is an output parameter.

JRNENTLRG

The name of the CL program variable into which the sequence number of the last fully processed journal entry is copied. The variable must be a character variable with a length of 20. This is an output parameter.

Both the **JRNENTLRG** and **JRNENTRY** parameters return the last applied journal sequence number if the journal sequence number is less than 10000000000. However, if the journal sequence number is 10000000000 or greater, only the **JRNENTLRG** parameter is returned, and the **JRNENTRY** parameter will return the decimal value -1.

JRNENTRY

The name of the CL program variable into which the sequence number of the last fully processed journal entry is copied. The variable must be a decimal variable that has a length of 10 positions with no decimal positions. This is an output parameter.

JRNENTRY returns the last applied journal sequence number if the journal sequence number is less than 10000000000, and returns -1 if the journal sequence number is 10000000000 or greater.

JRNRCVNME

The name of a 10-character CL variable into which the name of the journal receiver that contains the last fully processed journal entry for the journal will be copied. This is an output parameter.

JRNRCVLIB

The name of a 10-character CL variable into which the name of the library that contains the journal receiver that is identified through the **JRNRCVNME** parameter is copied. This is an output parameter.

RESULT

The name of a 10-character CL variable where the result of the command is copied. This is an output parameter.

If the command runs successfully, the result variable contains the string '*SUCCESS'. Otherwise, the result variable contains the string '*ERROR'.

SINGLEGRP

Specifies whether to retrieve the last applied position for a single group or for all groups that are using the journal on this node. This is an input parameter.

The possible values are:

- *NO—Does not retrieve the last applied position for a single group. By default, this value is used.
- *YES—Retrieves the last applied journal position for the group that is specified with the **GRPNAM** parameter.

GRPNAM

Specifies the name of the replication group for which the last applied journal position is retrieved. This parameter requires SINGLEGRP(*YES). This is an input parameter.

- <name>—The name of the replication group.

Example 1

```
RTVHAPOS JOURNAL(LIB1/JRN1) TARGET(&NODE) GROUP(&GRPVAR)
JRNENTRY(&JRNEVAR) JRNRCVNME(&JRNRVAR) JRNRCVLIB(&JRNLLVAR)
RESULT(&RESVAR) SINGLEGRP(*YES) GRPNAME(PAYROLL)
```

This command retrieves the journal receiver, the library of the journal receiver, and the sequence number of the last fully processed journal entry for journal **JRN1** in library **LIB1** for the group **PAYROLL**. This information is received through the variables **&JRNRVAR**, **&JRNLLVAR**, and **&JRNEVAR**, respectively.

The replication group that is associated with the sequence number is returned through the variable **&GRPVAR**. In this example, the returned value should be **PAYROLL**.

The backup node that is associated with the group is returned through the variable **&NODE**.

Example 2

```
RTVHAPOS JOURNAL(LIB1/JRN1) TARGET(&NODE) GROUP(&GRPVAR) JRNENTRY(&JRNEVAR)
JRNRCVNME(&JRNRVAR) JRNRCVLIB(&JRNLLVAR) RESULT(&RESVAR)
```

This command retrieves the journal receiver, the library of the journal receiver, and the sequence number of the last fully processed journal entry for journal **JRN1** in library **LIB1** for all groups. This information is received through the variables **&JRNRVAR**, **&JRNLLVAR**, and **&JRNEVAR**, respectively.

The replication group that is associated with the sequence number is returned through the variable **&GRPVAR**.

The backup node that is associated with the group is returned through the variable **&NODE**.



Note: It is recommended that you use the CL naming convention of an ampersand (&) in front of the variable names.

Restrictions

This command must be executed from a CL program. It is not available from the menus. You must issue this command on the node where the journal resides. The node does not have to be active in the cluster.

Minimum Authority Level

Any user with access to the iCluster product library.

Menu Access

None.

VFYHAJRN (Verify Audit Journal command)

Verify that the audit journal exists in the library **QSYS** and the audit related IBM i system values are set appropriately for iCluster. Issue this command after installation and before starting iCluster.

```
VFYHAJRN AUDQTEMP( ) AUDSPLF( )
```

The user profile that runs the command must have *ALLOBJ or *SECOFR authority to set the level of security auditing appropriately.

- ! **Note:** This command sets two IBM i system values (QAUDLVL, QAUDCTL) that are required for replication. Other system values that have been set for QAUDLVL and QAUDCTL are maintained and are not affected by this command.

The VFYHAJRN (Verify Audit Journal) command verifies that the audit journal exists in the library **QSYS**. The required system values for parameter **QAUDLVL** (security auditing level) are as follows:

- *CREATE
- *DELETE
- *OBJMGT
- *SAVRST
- *SECURITY
- *SPLFDTA (if the **AUDSPLF** parameter is set to *YES).
- *PRTDTA (if the **AUDSPLF** parameter is set to *YES).

This command also starts object-level auditing before using iCluster. The required system values for parameter QAUDCTL (auditing control) are as follows:

- *OBJAUD
- *AUDLVL
- *NOQTEMP (if the **AUDQTEMP** parameter is set to *NO).

Specifying *OBJAUD means that objects that are selected for audit by using the IBM i CHGOBJAUD command are audited. *AUDLVL ensures that object auditing changes that are controlled by the QAUDLVL system value are performed.

This command will also ensure that the new object auditing level is set correctly for iCluster.

Input Parameters

AUDQTEMP

Specifies whether you want to audit objects in the library QTEMP.

The possible values are:

- *NO—Does not audit objects in the library QTEMP by specifying *NOQTEMP for QAUDCTL. By default, this

value is used.

! **CAUTION:** *NO is the recommended setting to avoid auditing of objects in QTEMP.

- *YES—Audits objects in the library QTEMP by not specifying *NOQTEMP for QAUDCTL.

AUDSPLF

Specifies whether you want to audit spooled file functions (create spooled file, delete spooled file, change spooled file, and so on).

The possible values are:

- *NO—Does not audit spooled file functions. By default, this value is used.
- *YES—Audits spooled file functions.

! **Note:** If *SPLFDTA and *PRTDTA are not included in the QAUDLVL of the IBM i system value, iCluster adds them to the QAUDLVL when AUDSPLF is equal to *YES.

Example

```
VFYHAJRN AUDQTEMP(*NO) AUDSPLF(*NO)
```

Verifies that the audit journal exists and correct values are set for the operating system values QAUDLVL and QAUDCTL.

Objects in the library QTEMP are not audited (the system value *NOQTEMP is set for QAUDCTL).

Spooled file functions are not audited.

Restrictions

You must issue this command on the node where the operations are performed. The node does not have to be active in the cluster.

Minimum Authority Level

Any user with access to the iCluster product library.

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **64**

DMSTRAPY (Start Replication Apply Process command)

Starts the apply processes on a backup node if they are not running. Starting apply processes allows journal entries that are placed in the staging store by receive processes to be applied.

```
DMSTRAPY GROUP( ) BACKUP( ) OVRSTOP( ) FRCDRN( )
```

The apply processes handle each journal entry in the staging store in chronological order. They remain active until the stop date and time that is specified through the previous or subsequent DMENDAPY (End Replication Apply Process) command. The apply processes also end when the staging store has been drained and the send/receive processes are not running.

Input Parameters

GROUP

The name of the replication group that will have its apply processes started.

The possible values are:

- <name>—Specifies the name of the group.
- *ALL—Starts the apply processes on the backup node for all groups in the cluster.
- *BACKUP—Starts the apply processes for all groups on the specified backup node. If you specify this option, you must specify the name of the backup node in the **BACKUP** parameter.

BACKUP

Indicates the name of the backup node that starts the apply processes for all groups that have the node as their backup node. A specific backup node can only be specified if the **GROUP** parameter is set to *BACKUP.

The possible values are:

- <name>—Specifies the name of the backup node.
- *ALL—Starts the apply processes for all groups on all backup nodes in the cluster. By default, this value is used.
- *HABACKUP—Specifies that the apply processes will only be started for the HA backup node of the group. Valid only when starting a single *HADR group.
- *DRBACKUP—Specifies that the apply processes will only be started for the DR backup node of the group. Valid only when starting a single *HADR group.

OVRSTOP

Overrides a previously set stop date and time for the apply processes that was specified through the DMENDAPY (End Replication Apply Process) command.

The possible values are:

- *YES—Disregards the previous stop date and time. The apply processes will continue until a new DMENDAPY (End Replication Apply Process) command is issued. By default, this value is used.
- *NO—Starts the apply processes, but the previous stop date and time that was specified through the DMENDAPY (End Replication Apply Process) command is still valid. Consequently, the processes will stop at that date/time unless it is reset by a subsequent invocation of the DMENDAPY (End Replication Apply Process) command.

FRCDRN

Specifies whether the staging store is force drained.

The apply processes merge entries from the audit and database journals. Depending on the selection, the apply processes either continue draining the staging store even when the staging store channel for one of the journals is empty (known as force draining), or it stops draining when the staging store channel for one of the journals is empty.

Force draining stops after it reaches the stop date/time that is specified through the DMENDAPY (End Replication Apply Process) command (if one is specified). Otherwise, it stops after the staging store channels of all journals to the specified group or groups are empty.

The possible values are:

- *NO—Does not force drain the staging store. The apply processes merge entries from the audit and database journals and apply them in sequence. When the staging store channel for one journal becomes empty, the apply process stops regardless of any entries remaining in the staging store channel for the other journal. By default, this value is used.
- *YES—Force drains the staging store. The apply processes merge entries from the audit and database journals. After it reaches the last entry of the staging store channel with fewer entries, the merge stops, and the apply process drains the staging store channel for the other journal until it is empty.

Example 1

```
DMSTRAPY GROUP(GRP1) OVRSTOP(*YES) FRCDRN(*YES)
```

Starts the apply processes on the backup node for replication group **GRP1**.

Overrides the stop date and stop time that was set by the previous **DMENDAPY** command.

The staging store is force drained.

Example 2

```
DMSTRAPY GROUP(*BACKUP) BACKUP(SYSIBU) OVRSTOP(*YES) FRCDRN(*YES)
```

Starts the apply processes for all groups that have the **SYSIBU** backup node.

The staging store is force drained.

Restrictions

You can issue this command on any node in the cluster.

Minimum Authority Level

*OPERATOR

Menu Access

Main Menu - Option 16

iCluster Commands menu - F22 (Shift + F10) - Option **66**

DMENDAPY (End Replication Apply Process command)

Ends applying journal entries from the staging store on a backup node.

```
DMENDAPY GROUP( ) BACKUP( ) OPTION( ) ENDDATE( ) ENDTIME( )
```

Even though this command stops the apply processes, iCluster continues to scrape and send journal entries for the replication group if it is active. The journal entries remain in the staging store until the apply processes are started through the **DMSTRAPY** (Start Replication Apply Process) command.

Input Parameters

GROUP

The name of the replication group that contains the backup node that has its apply processes stopped by this command.

The possible values are:

- <name>—Specifies the name of the group.
- *ALL—Ends the apply processes on the backup node for all groups in the cluster.
- *BACKUP—Ends the apply processes for all groups that have the specified node as their backup node. If you specify this option, you must specify the name of the backup node in the **BACKUP** parameter.

BACKUP

Indicates the name of the backup node that will end the apply processes for all groups. A specific backup node can only be specified if the **GROUP** parameter is set to *BACKUP.

The possible values are:

- <name>—Specifies the name of the backup node.
- *ALL—Ends the apply processes for all groups on all backup nodes in the cluster. By default, this value is used.
- *HABACKUP—Specifies that the apply processes will only be ended for the HA backup node of the group. Valid only when starting a single *HADR group.
- *DRBACKUP—Specifies that the apply processes will only be ended for the DR backup node of the group. Valid only when starting a single *HADR group.

OPTION

Indicates how to stop the apply processes.

The possible values are:

- *CTRLD—Indicates a controlled end of the apply processes, allowing the apply processes to complete their current processing. It is recommended to perform a controlled end when possible. By default, this value is used.
- *INVLD—Invalidates the pending end date/time for the apply processes. Apply processes continue.
- *DATETIME—Indicates the latest date and time for a journal entry to be applied. Journal entries that are deposited after this date and time are not processed. You must specify values for the **ENDDATE** and **ENDTIME** parameters.

Indicates the latest date and time for a journal entry to be applied. When the apply process encounters an entry deposited at later date/time it ends without processing that entry.

- *IMMED—Indicates that the apply processes stop immediately.

Note: If either *CTRLD or *DATETIME is specified, the apply processes stop after the current operation has been completed. The amount of time for the apply processes stop depends on the operation (for example, restoring a save file or committing a commitment control cycle with a large number of transactions).

ENDDATE

The date to stop the apply processes. The date format must adhere to the system date format.

This parameter is applicable only when the **OPTION** parameter is set to *DATETIME.

ENDTIME

The time to stop the apply processes. Time is based on the journal transaction time stamp, not the system date and time.

This parameter is only applicable if the **OPTION** parameter is set to *DATETIME.

Example 1

```
DMENDAPY GROUP(GRP1) OPTION(*CTRLD)
```

Ends the apply processes on the backup node in group **GRP1**.

The apply processes are stopped in a controlled manner after the current operation completes.

Example 2

```
DMENDAPY GROUP(*ALL) OPTION(*DATETIME) ENDDATE(12/31/14) ENDTIME(235959)
```

Ends the apply processes for all groups on all backup nodes in the cluster.

Stops the apply processes on the specified date (December 31, 2014) and the specified end time (11:59 p.m. and 59 seconds).

Restrictions

You can issue this command on any active node in the cluster.

Minimum Authority Level

*OPERATOR

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **67**

RTVAPYSTS (Retrieve Apply Status command)

In a program, retrieves the Apply status for the specified group or groups. The command returns a character value that represents the status of the apply processes for the specified iCluster replication groups.

```
RTVAPYSTS GROUP( ) APYSTS( )
```

The apply processes handle each journal entry in the staging store in chronological order. The apply processes remain active until the specified stop date and time. Stop date and time are specified through the previous or subsequent DMENDAPY (End Replication Apply Process) command. The apply processes also end when the staging store has been drained and the send/receive processes are not running, or if the group is ended with the DMENDGRP (iCluster End Group) command.

This command returns one of the following single characters that represent the apply process status:

A

Active. All apply processes are active.

I

Inactive. All apply process are inactive or the set of selected groups is empty.

X

In doubt. Some apply processes are active and some apply processes are inactive.

U

Unknown. An error occurred while processing this request.

Input Parameters

GROUP

The replication group or groups for which the apply status is being retrieved.

The possible values are:

- <name>—Specifies the name of a group.
- *ALL—All groups in the cluster that have the local node as their backup node.
- *PRIMNODE—All groups with the primary node that is specified in the **PRIMNODE** parameter. Includes only the groups that have the local node as the backup node.

PRIMNODE

Specifies the name of the primary node for the group or groups for which the apply status is being retrieved. This parameter applies only when the **GROUP** parameter is set to *PRIMNODE.

The required value is:

- <name>—The name of the primary node.

APYSTS

Specifies the name of the variable to receive the 1-character value that represents the apply process status.

The required value is:

- <variable>—The name of the CL program variable.

Example

```
RTVAPYSTS GROUP(GRP1) APYSTS(&STS)
```

Returns a character value that represents the status of the apply processes for replication group **GRP1**, and puts the single-character value in the CL program variable **&STS**.

Restrictions

You must issue this command in a program. This command cannot be issued on the command line.

Minimum Authority Level

*OPERATOR

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **67**

DMACTOBJ (Activate Object command)

Activates an object that is currently suspended from replication and enables replication of objects that are in replication scope.

```
DMACTOBJ GROUP( ) OBJ( ) OBJTYPE( ) MEMBER( ) RFSH( ) PFRFSHMTD( ) BACKUP( )
```

After activating an object, mirroring of the object starts when the replication group is active. If the group is inactive, mirroring starts when the group is activated with the DMSTRGRP (iCluster Start Group) command.

Any changes made to an object while it is suspended are not mirrored when the object is activated with this command. Specify *YES for the **RFSH** parameter to ensure that the object on the backup node is synchronized with the object on the primary node when activating the object for mirroring.

Input Parameters

GROUP

The name of the replication group to which the object is selected.

OBJ

The object name component of the object to activate.

The possible values are:

- <name>—Specifies the object name.
- <generic*>—Specifies a generic object name, to identify multiple objects in a library.
- *ALL—Specifies all objects in a library.

Note: If you specify a generic name for this parameter, only *ALL is allowed for the **MEMBER** parameter. If you specify a specific name for this parameter, both *ALL and a specific name are allowed for the **MEMBER** parameter.

OBJTYPE

The object type component of the specifier to select.

The possible values are:

- <type>—Specifies the object type. Press F4 for a list of all values.
- *ALL—Specifies all object types.

*ALL cannot be used for object types that can reside only in the QSYS library, that is, *LIB (library), *USRPRF (user profile) and the configuration object types *CNNL, *COSD, *CTLD, *DEVD, *IPXD, *LIND, *MODD, *NTBD, *NWID, *NWSID.

MEMBER

This parameter is available only if you entered *FILE in the **OBJTYPE** parameter and the file is a specific source physical file or a database file.

Specifies whether you want to activate a specific member of a source physical file or a database file.

The possible values are:

- <name>—Specifies the member name.
- *ALL—Specifies all members for the file. By default, this value is used.

RFSH

Specifies to refresh the object to the backup node when the object is activated. This option allows you to activate an object without having to refresh the entire object.

The possible values are:

- *NO—Does not refresh the object to the backup node when the object is activated. In this case, you are responsible for refreshing the object and ensuring that it is synchronized at the time the activation is performed.
- *YES—Refreshes the object to the backup node when the object is activated. By default, this value is used.
- *FIXREC—Refreshes only the records that caused the record-level errors (RLE) based on relative record number instead of the entire file. Applies only to files that are listed in ACTOOS state with the RLE reason code when the **MAXRLE** parameter is set to *FIX in the system value or the DMSELOBJ (Select Objects to Group) or DMCHGOBJSL (Change Object Selection) commands.

PFRFSHMTD

Specifies the method of physical file refresh when the object is activated. This parameter applies only when the **RFSH** parameter is set to *YES.

The possible values are:

- *RBR—Record-by-record refresh method. By default, this value is used.
- *SAVRST—Save and restore refresh method. Saves the files with the data and restores the file on the backup node. Improves performance for large files.

Note: The *SAVRST method of physical file refresh will not succeed if there are any other update or exclusive locks on the file being refreshed at the time of refresh, or if the file is under commitment control. A file's refresh by *SAVRST will also fail if the group's "Save active" parameter evaluates to *NO and the file is open or locked for update by another application. In these cases, you should specify the *RBR refresh method.

- *OBJ—Refresh by the method specified for the object, as determined by the object's specifier, group, and the product's system values.

BACKUP

Specifies the name of the backup node for which the objects will be activated.

If the group is not an *HADR group, you can use the special value *ONLY or you can specify the name of the group's backup node. If the group is an *HADR group, you must name one of the group's backup nodes.

The possible values are:

- <name>—Specifies the name of the group's backup node.
- *ONLY—Specifies that the group's only backup node is the backup node for the object activation. This value is not valid for *HADR groups.

Example 1

```
DMACTOBJ GROUP(GRP1) OBJ(LIB1/OBJ1) OBJTYPE(*FILE) RFSH(*YES) BACKUP(*ONLY)
```

Activates the *FILE object named **OBJ1** in library **LIB1** that is selected for replication within replication group **GRP1**.

Before mirroring is restarted, the **LIB1/OBJ1** object is refreshed to the backup node using the default record-to-record refresh method.

Specifies that the group's only backup node is the backup node for the object activation.

Example 2

```
DMACTOBJ GROUP(GRP1) OBJ(LIB1/OBJ1) OBJTYPE(*FILE) MEMBER(MEM1) BACKUP(*ONLY)
```

Activates the member **MEM1** of the file named **OBJ1** that is located in the library **LIB1** and is part of the group **GRP1**. **MEM1** is refreshed to the backup node because the default value of *YES is used for the **RFSH** parameter.

Specifies that the group's only backup node is the backup node for the object activation.

Example 3

```
DMACTOBJ GROUP(GRP1) OBJ(LIB1/A*) OBJTYPE(*ALL) MEMBER(*ALL) BACKUP(*ONLY)
```

Activates all objects located in library **LIB1** that have 'A' as their first letter. These objects are also in the replication scope of **GRP1**. The objects are refreshed to the backup node because the default value of *YES is used for the **RFSH** parameter.

Specifies that the group's only backup node is the backup node for the object activation.

Example 4

```
DMACTOBJ GROUP(GRP1) OBJ(LIB1/*ALL) OBJTYPE(*ALL) MEMBER(*ALL) RFSH(*YES)  
PFRFSHMTD(*SAVRST) BACKUP(*ONLY)
```

Activates and refreshes all objects in library **LIB1** that are also in the replication scope of **GRP1**.

Uses the save and restore physical file refresh method.

Specifies that the group's only backup node is the backup node for the object activation.

Restrictions

You can issue this command on any active node in the cluster.

Minimum Authority Level

*OPERATOR

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **68**

Related information

[**Suspended objects**](#)

[**DMSELOBJ \(Select Objects to Group command\)**](#)

[**DMACTSQLO \(Activate SQL Object command\)**](#)

DMACTBSF (Activate BSF Object command)

Activates a BSF object in the Integrated File System (IFS) or refreshes a BSF object through the iCluster replication engine.

```
DMACTBSF GROUP( ) PATH( ) RFSH( ) BACKUP( )
```

When a BSF object is activated, it can be refreshed to the backup node in the recovery domain when mirroring is active. Mirroring of the object starts if the replication group is active. If the group is inactive, mirroring starts when the group is started with the DMSTRGRP (iCluster Start Group) command.

Input Parameters

GROUP

The name of the replication group to which the BSF object is selected.

PATH

The path object specifier that identifies the location of the BSF object that is activated through this command.

Enclose the path in single quotation marks ('') and start with a forward slash (/), for example '/Dir3/Dir4/file'. The path can be 2 - 5000 characters.

Generic path names of the form '/mydir*' are supported, where the generic indicator '*' is the final character of the path name. When using generic path names, all sub-directories are included recursively.

Generic file name extensions after the asterisk (*) are supported, where the file name extension follows the period (.) and is 1 - 8 alphabetic characters.

For example, '/mydir/*.txt' and '/mydir/prod*.txt'. File name extension support does not apply to /QDLS pathnames.

- ! **Note:** Only objects that match the generic path specifier and are replicated by the group are activated. For example, matching objects that are excluded from replication by the group are not activated.

RFSH

Specifies whether to refresh the path object to the backup node when the object is activated. This option enables path object activation without having to save and restore the object.

The possible values are:

- *NO—Does not refresh the object to the backup node when the object is activated. In this case, you must manually refresh the object to ensure that it is synchronized at activation time. Specify *NO to remove the object from the suspended object list, if the object is still suspended but was deleted on the primary node.
- *YES—Refreshes the path object to the backup node when the object is activated. Verifies that the object on the backup node is correctly synchronized with the object on the primary node. Fully refreshes the object to ensure that changes made to a suspended object are mirrored when the object is activated with this command. By default, this value is used.

BACKUP

Specifies the name of the backup node for which the objects will be activated.

If the group is not an *HADR group, you can use the special value *ONLY or you can specify the name of the group's backup node. If the group is an *HADR group, you must name one of the group's backup nodes.

The possible values are:

- <name>—Specifies the name of the group's backup node.
- *ONLY—Specifies that the group's only backup node is the backup node for the object activation. This value is not valid for *HADR groups.

Example 1

```
DMACTBSF GROUP(GRP1) PATH('/DIR1/DIR2/DIR3/FILEA') BACKUP(*ONLY)
```

Activates the BSF object named **FILEA** in **/DIR1/DIR2/DIR3** that is selected for replication within group **GRP1**.

FILEA in directory **/QDLS/DIR1/DIR2** is refreshed to the backup node in the group.

Specifies that the group's only backup node is the backup node for the object activation.

Example 2

```
DMACTBSF GROUP(GRP2) PATH('/QDLS/DIR1/DIR2/*') BACKUP(*ONLY)
```

Activates all Document Library Objects (DLO) in directory **/QDLS/DIR1/DIR2** that are selected for replication within group **GRP2**.

The objects are refreshed to the backup node in the recovery domain.

Specifies that the group's only backup node is the backup node for the object activation.

Restrictions

You can issue this command on any active node in the cluster.

Minimum Authority Level

*OPERATOR

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **69**

Related information

[Path object specifiers](#)

[Replicating BSF objects](#)

[Suspended objects](#)

DMACTOOS (Activate Out-of-Sync Objects command)

Activates out-of-sync objects for the specified group.

```
DMACTOOS GROUP( ) RFSHTYPE( ) BACKUP( ) MAXACTSIZ( ) MAXACTNUM( )
```

This command does not activate out-of-sync objects that are suspended with one of the following reason codes that prevent automatic reactivation.

Reason Codes that Prevent Automatic Reactivation

EJF

The object was suspended on the primary node because journaling ended on the primary node for the object.

JPF

A logical file was suspended on the primary node because the associated physical file could not be journaled.

LNK

The BSF object is a hard link. Replication of hard links is not supported.

MRR

The file was suspended on the primary node. The file should be refreshed manually but it has yet to be activated.

NGP

A logical file was suspended on the primary node because the associated physical file was not replicated in the same group as the logical file.

NSO

A physical file contains fields of an unsupported type (datalink or LOB fields with MINENTDTA).

RBC

The file was part of a canceled rollback operation on the primary node.

RGF

The file was suspended because a member could not be reorganized or cleared on the backup node.

RRD

Record-by-record refresh of a file failed due to a record read failure.

SBU

An object was explicitly suspended on the primary node as a result of issuing the DMSUSOBJ or DMSUSBSF request.

SIZ

An object was suspended on the primary node. A refresh of the object was required, but the size of the object is greater than the value specified by the **Maximum Refresh Size** system value.

SPF

A logical file was suspended on the primary node because the associated physical file was suspended.

SPL

An *OUTQ object has suspended spooled files. The *OUTQ object itself is not suspended.

TNE

The BSF object does not exist on the backup node.

TNS

The BSF object is a type of object that cannot be replicated.

UKM

The file is suspended on the backup node. The file object specifier was added with **PFKEY (*AUTO)** but a unique key could not be found for the file on the backup node.

Input Parameters

GROUP

Specifies the name of the group whose out-of-sync objects will be refreshed.

Example

```
DMACTOOS GROUP(GRP1) BACKUP(*ONLY)
```

Activates the out-of-sync objects in the replication group named **GRP1**.

The objects are refreshed because of the default value ***FULL** for parameter **RFSHTYPE**.

Specifies that the group's only backup node is the backup node for the object activation.

RFSHTYPE

Specifies the type of refresh to perform for the out-of-sync objects for the group.

The possible values are:

- ***FULL**—Fully activates and refreshes the out-of-sync objects for the group and removes the objects from the list of out-of-sync objects. By default, this value is used.
- ***FIXREC**—Refreshes only the physical file records that are found to be out of sync with a ***CHECKSUM** sync check. The files are activated and removed from the list of out-of-sync objects for the group if the files are not listed as being out of sync for another reason.
- ***AUTH**—Refreshes only the private authorities and ownership of objects that are listed as out of sync with the AUT (authority differences), NAU (number of authorized users), OWN (object owner), or PGP (primary group) reason code. The objects are removed from the list of out-of-sync objects for the group if the files are not listed as being out of sync for another reason besides the authority difference. Suspended objects are not activated, only their authorities are refreshed.
- ***JRN**—Refreshes only the journal information of objects that are out of sync with the JRN (journaling on target) reason code. The objects are removed from the list of out-of-sync objects for the group if they are not listed as being out of sync for another reason.

BACKUP

Specifies the name of the backup node for which the objects will be activated.

If the group is not an ***HADR** group, you can use the special value ***ONLY** or you can specify the name of the group's backup node. If the group is an ***HADR** group, you must name one of the group's backup nodes.

The possible values are:

- **<name>**—Specifies the name of the group's backup node.
- ***ONLY**—Specifies that the group's only backup node is the backup node for the object activation. This value is not valid for ***HADR** groups.

MAXACTSIZ

Specifies the largest object size to include in the activation. The size of an object can affect network performance and introduce mirroring latency.

Possible values include:

- **max-size**—Specify the maximum size in kilobytes (KB) of an object. The largest value that you can enter into this field is 52,428,800.
- ***NOMAX**—Specify that objects of all sizes are included in the activation.

Note: The *NOMAX value might have serious performance issues on the primary node, if there are very large objects to process.

MAXACTNUM

Specifies the maximum number of objects to include in the activation. The number of objects can affect network performance and introduce mirroring latency.

Possible values include:

- maximum-number-of-objects—Specify the maximum number of objects. The largest value that you can enter into this field is 10,000.
- *NOMAX—Specify that all objects are included in the activation.

Note: The *NOMAX value might have serious performance issues on the primary node, if there are very large objects to process.

DMACTOOS examples

To activate all out-of-sync objects, enter the applicable values for DMACTOOS and press Enter. For example:

```
Group . . . . . . . . . . Name
Backup node . . . . . . . . *ONLY Name, *ONLY
Refresh type . . . . . . . . *FULL *FULL, *FIXREC, *AUTH, *JRN
Max. activation size (KB) . . . 2097152 1-52428800, *NOMAX
```

```
Group . . . . . . . . . . Name
Backup node . . . . . . . . *ONLY Name, *ONLY
Refresh type . . . . . . . . *FULL *FULL, *FIXREC, *AUTH, *JRN
Max. activation size (KB) . . . 2097152 1-52428800, *NOMAX
Max. # of activation objects . . 1000 1-10000, *NOMAX
```

Restrictions

None.

Minimum Authority Level

*OPERATOR

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **81** - Option **13**

DMACTSUS (Activate Suspended Objects command)

Activates certain suspended objects for the group.

```
DMACTSUS GROUP( ) REASON( ) BACKUP( )
```

The suspension reason codes that are eligible for activation with this command are:

- EJF—Object suspended because journaling for the object was ended on the primary node.
- SBU—Object suspended by user.

 **Note:** Objects suspended with the EJF or SBU reason code are not eligible for auto-reactivation.

Input Parameters

GROUP

The name of an existing replication group.

REASON

Specifies the suspension reason code for the objects that will be activated.

The possible values are:

- *ALL—All objects for the group that are suspended for the reason codes EJF or SBU will be activated with a full refresh. By default, this value is used.
- EJF—Objects for the group that are suspended with reason code EJF (object suspended on the primary node because journaling was ended for the object) will be activated with a full refresh.
- SBU—Objects for the group that are suspended with reason code SBU (object suspended by user) will be activated with a full refresh.

BACKUP

Specifies the name of the backup node for which the objects will be activated.

If the group is not an *HADR group, you can use the special value *ONLY or you can specify the name of the group's backup node. If the group is an *HADR group, you must name one of the group's backup nodes.

The possible values are:

- <name>—Specifies the name of the group's backup node.
- *ONLY—Specifies that the group's only backup node is the backup node for the object activation. This value is not valid for *HADR groups.

Example

```
DMACTSUS GROUP(GRP1) REASON(EJF)
```

Activates with a full refresh those objects within replication group **GRP1** that have been suspended with reason code **EJF**.

Specifies that the group's only backup node is the backup node for the object activation.

Restrictions

You can issue this command on any active node in the cluster for any group.

Minimum Authority Level

*OPERATOR

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **68**

Related information

[DMACTOBJ \(Activate Object command\)](#)

DMACTSLO (Activate SQL Object command)

Activates one or more SQL objects with refresh through the replication engine.

```
DMACTSLO GROUP( ) OBJ( ) OBJTYPE( ) BACKUP( )
```

After activating a SQL object, the SQL object will be synchronized on the backup node if the replication group is active. If the group is inactive, the SQL object will be synchronized on the backup node when the group is started through the DMSTRGRP (iCluster Start Group) command.

Input Parameters

GROUP

The name of the replication group to which the SQL object to be activated is selected.

OBJ

The object name component of the SQL object to activate. The name of a SQL object can be up to 128 characters long. Enter a specific or generic SQL object name to identify one or multiple SQL objects in a library.

The possible values are:

- <name>—Specifies the SQL object name.
- <generic*>—Specifies a generic SQL object name, to identify multiple SQL objects in a library.
- *ALL—Specifies all SQL objects in a library.

! **Note:** The library where the SQL object resides must be identified. Prefix the SQL object with the name of the library where the object is located (for example, LIB1/SQLOBJ1).

OBJTYPE

The type component of the SQL object that you want to activate or *ALL.

BACKUP

Specifies the name of the backup node for which the objects will be activated.

If the group is not an *HADR group, you can use the special value *ONLY or you can specify the name of the group's backup node. If the group is an *HADR group, you must name one of the group's backup nodes.

The possible values are:

- <name>—Specifies the name of the group's backup node.
- *ONLY—Specifies that the group's only backup node is the backup node for the object activation. This value is not valid for *HADR groups.

Example 1

```
DMACTSQLO GROUP(GRP1) OBJ(LIB1/SQLOBJ1) OBJTYPE(*SQLFUNC) BACKUP(*ONLY)
```

Activates the SQL function named **SQLOBJ1** in library **LIB1** that is selected for replication within replication group **GRP1**.

Specifies that the group's only backup node is the backup node for the object activation.

Example 2

```
DMACTSQLO GROUP(GRP1) OBJ(LIB1/A*) OBJTYPE(*SQLFUNC) BACKUP(*ONLY)
```

Activates all SQL functions located in library **LIB1** that have 'A' as their first letter. These SQL functions are also in the replication scope of **GRP1**.

Specifies that the group's only backup node is the backup node for the object activation.

Restrictions

You can issue this command on any active node in the cluster.

Minimum Authority Level

*OPERATOR

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **78**

Related information

[Replicating SQL objects](#)

DMSUSOBJ (Suspend Object command)

Suspends an object and prevents it from being replicated to a backup node.

```
DMSUSOBJ GROUP( ) OBJ( ) OBJTYPE( ) MEMBER( ) BACKUP( )
```

The object is suspended with the SBU reason code. Journal entries that were created before an object was suspended will still be processed, sent and applied on the backup node.

To activate an object that is suspended through this command, use the DMACTOBJ (Activate Object) command. To activate all objects in a group that are suspended with the SBU (suspended by user) reason code, you can use the DMACTSUS (Activate Suspended Objects) command.



Note: Any changes made to an object while it is suspended will not be mirrored when it is activated, unless it is activated with a refresh of the object.

Input Parameters

GROUP

The name of the replication group to which the object is selected.

OBJ

The name of a valid object.

The library where the object resides must be identified. Prefix the object with the name of the library where the object is located. For example:

LIB1/OBJ1

MEMBER

Specifies whether you want to suspend a specific member of a source physical file.

This parameter is available only if you entered *FILE in the **OBJTYPE** parameter.

The possible values are:

- <name>—Specifies the name of a single member of the file that is to be suspended.
- *ALL—Specifies all members for the specified source physical file.

BACKUP

Specifies the name of the backup node for which the objects will be suspended.

If the group is not an *HADR group, you can use the special value *ONLY or you can specify the name of the group's backup node. If the group is an *HADR group, you must name one of the group's backup nodes.

The possible values are:

- <name>—Specifies the name of the group's backup node.
- *ONLY—Specifies that the group's only backup node is the backup node for the object suspension. This value is not valid for *HADR groups.

Example

```
DMSUSOBJ GROUP(GRP1) OBJ(LIB1/OBJ1) OBJTYPE(*FILE) MEMBER(MEM1) BACKUP(*ONLY)
```

Suspends the member **MEM1** of the *FILE object named **OBJ1** in library **LIB1** that is selected for replication within group **GRP1**.

Specifies that the group's only backup node is the backup node for the object suspension.

Restrictions

You must issue this command on an active node in the cluster.

Minimum Authority Level

*OPERATOR

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **70**

Related information

[Suspended objects](#)

[DMACTSUS \(Activate Suspended Objects command\)](#)

DMSUSBSF (Suspend BSF Object command)

Prevents a Byte Stream File (BSF) object from being replicated to a backup node.

```
DMSUSBSF GROUP( ) PATH( ) BACKUP( )
```

The object will appear suspended with the SBU reason code. It applies only to BSF objects residing in the Integrated File System (IFS).

Journal entries created before the object was suspended will still be processed, sent to, and applied on the backup node.

To activate a BSF object that is suspended through this command, use the DMACTBSF (Activate BSF Object) command. To activate all BSF objects for the group that are suspended with the SBU (suspended by user) reason code, use the DMACTSUS (Activate Suspended Objects) command.



Note: This command supports the suspension of directories and folders with the exception of those directories that match a BSF specifier with matching files that are journaled.

Input Parameters

GROUP

The name of the replication group to which the BSF objects are selected.

PATH

The path must be enclosed in single quotes and start with a '/' (forward slash) character ('/Dir3/Dir4/file').

The path can be a maximum of 5000 characters, and you must enter at least two characters for the path.

Generic paths are not supported.

BACKUP

Specifies the name of the backup node for which the objects will be suspended.

If the group is not an *HADR group, you can use the special value *ONLY or you can specify the name of the group's backup node. If the group is an *HADR group, you must name one of the group's backup nodes.

The possible values are:

- <name>—Specifies the name of the group's backup node.
- *ONLY—Specifies that the group's only backup node is the backup node for the object suspension. This value is not valid for *HADR groups.

Example

```
DMSUSBSF GROUP(GRP1) PATH(' /DIR1/DIR2/DIR3/FILEA') BACKUP(*ONLY)
```

Suspends the BSF object named FILEA in /DIR1/DIR2/DIR3 that is selected for replication within group GRP1.

Specifies that the group's only backup node is the backup node for the object suspension.

Restrictions

You must issue this command on an active node in the cluster.

Minimum Authority Level

*OPERATOR

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **71**

Related information

[Path object specifiers](#)

[Replicating BSF objects](#)

[Suspended objects](#)

[DMACTSUS \(Activate Suspended Objects command\)](#)

[DMADDBSF \(Add Path Specifier to Group command\)](#)

CRTCFGOBJ (Create Configuration Objects command)

Creates configuration objects that are replicated on a backup node.

```
CRTCFGOBJ OBJ( ) OBJTYPE( ) OWNER( )
```

iCluster can automatically create configuration objects immediately after the data required for creating them is received on the backup node. Alternatively, the configuration objects can be created later. If the latter approach is adopted, this command can be issued on the backup node to create the configuration objects.

Input Parameters

OBJ

The name of the configuration object to create.

The possible values are:

- <name>—Specifies a configuration object name.
- <generic*>—Specifies a generic object name, to identify multiple configuration objects.
- *ALL—Specifies all configuration objects whose data is being stored on the backup node.

OBJTYPE

The type of configuration object to create.

The possible values are:

- *ALL - Specifies all configuration object types.
- *CNNL—Connection list.
- *COSD—Class-of-service description.
- *CTLD—Controller description.

- *DEVD—Device description.
- *IPXD—Internet package exchange description.
- *LIND—Line description.
- *MODD—Mode description.
- *NTBD—NetBIOS description.
- *NWID—Network interface description.
- *NWSD—Network server description.

OWNER

This parameter is not used by iCluster any more. The object authority information is stored with the other parameters that are required to create the objects in the data on the backup node.

Example 1

```
CRTCFGOBJ OBJ(OBJ1) OBJTYPE(*MODD)
```

Creates the mode description **OBJ1**.

Example 2

```
CRTCFGOBJ OBJ(*ALL) OBJTYPE(*DEVD)
```

Creates all device descriptions whose data is being stored on the backup node.

Example 3

```
CRTCFGOBJ OBJ(*ALL) OBJTYPE(*ALL)
```

Creates all configuration objects whose data is being stored on the backup node.

Restrictions

If the configuration object being created already exists on the backup node, this command can be issued only when the object is not in use.

You must issue this command on the backup node where the configuration objects are created. However, the backup node does not have to be active.

Minimum Authority Level

Any user with access to the iCluster product library.

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **73**

Related information

[Replicating configuration objects](#)

INITHAOBJ (Initialize Objects command)

Prepares objects for mirroring so that any changes made to the objects that are selected to the groups are audited.

```
INITHAOBJ TARGET( ) GROUP( )
```

This command changes the IBM i **Object Auditing Value** of the objects to *CHANGE.

Initializing objects should occur automatically when you select the objects for replication with the DMSELOBJ (Select Objects to Group) command or the DMADDBSF (Add Path Specifier to Group) command. However, you might need to issue this command in the following situations:

- If the system auditing levels (QAUDLVL, and QAUDCTL) have been changed so that they no longer meet the required values for iCluster.
- If iCluster could not start auditing on one or more objects at the time the object specifier was added or modified. In this case a warning message is logged in the job log to remind you to issue the INITHAOBJ (Initialize Objects) command.

Any objects that are replicated by a group must have object auditing set to *CHANGE. Auditing should begin whenever an included specifier (BSF or native) is added to the group or whenever a specifier is changed to include an object or BSF.

The following restrictions apply:

- Excluded BSF objects are not respected when there is a more generic include. For example, /home/* *INC supersedes /home/dir1 *EXC or /home/dir1/* *EXC. In this example, all of the objects in /home/* are audited, including the objects in /home/dir1.
- The IBM i **Object Auditing Value** for BSF objects will always be set to *CHANGE.

Input Parameters

TARGET

The name of the backup node in the recovery domain for the replication group identified through the **GROUP** parameter.

GROUP

The name of the replication group that will have its selected objects initialized through this command.

Example

```
INITHAOBJ TARGET(NODE1) GROUP(GRP1)
```

Initializes all objects selected to the group **GRP1** that has **NODE1** as a backup node.

Restrictions

You must issue this command only on a primary node.

Minimum Authority Level

Any user with access to the iCluster product library.

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **74**

DMLOGENT (Log Journal Entry command)

Inserts one journal entry in each of the journals that are scraped by a group on the primary node to track the processing of journal entries from the primary node to the apply jobs on the backup node.

```
DMLOGENT GROUP( ) ENTTYP( ) USRDTA( ) DTAARA( )
```

When the entries for each journal have been processed by the group's apply processes on the backup node, a record for each entry is placed into the **DMLOG** database file on the backup node. Each record in the file contains the time that the journal entry was scraped and the time that it was applied, as well as the name of the group, the journal, and the journal receiver that contains the entry. This journal entry allows you to gauge the performance of groups and replication processes in your cluster. You can also use this command to gain insight into the progress of replication activities during batch runs or troubleshooting.

This command creates the **DMLOG** database file on the group's backup node if it does not already exist. The log file has the following format:

- GROUP (char 10)—Name of the group
- ENTTYPE (4 bytes)—Entry type
- JRN (char 10)—The name of the journal
- JRNLIB (char 10)—The journal library
- JOURNAL (char 20)—Journal with library
- RCV (char 10)—The name of the journal receiver that received the journal entry
- RCVLIB (char 10)—The journal receiver library
- JRNPOS (numeric 20)—The sequence number of the journal entry in the journal on the primary node. To determine the time that the journal entry was deposited into the journal on the primary node, issue the DSPJRN command for the journal sequence number and journal.
- SCRTIME (timestamp)—The date and time when the journal entry was processed by the journal scraper.
- APYTIME (timestamp)—The date and time when the journal entry was processed by the apply job
- USRDTA (char 400)—The user data entered on the DMLOGENT command.
- EXTDTA (char 2000)—The contents of the data area specified on the DMLOGENT command, if not *NONE. This content is filled in by the journal scraper.

Input Parameters

GROUP

The name of the group for which logging information is deposited and recorded.

Note: Only *REPL groups are supported. *REFRESH, *MQSERIES, and *SWDEV groups are not supported by the DMLOGENT (Log Journal Entry) command.

ENTTYP

Indicates the type of logging information. You can choose from a predefined type or create a user defined type.

The possible values are:

- <number>—Specifies an integer value from 1–2000.
- *INFO—Indicates an informational entry. By default, this value is used.
- *STRRUN—Indicates the beginning of mirroring.
- *ENDRUN—Indicates the end of mirroring.

USRDTA

A user defined field of up to 400 characters that is optional.

DTAARA

The name of a data area that is recorded in the log file. The data area cannot exceed 2000 bytes of character data.

The possible values are:

- <name>—Specifies the name of the data area.
- *NONE—Indicates that no data area is specified. By default, this value is used.

The library where the data area resides must be identified if you do not specify *NONE. Prefix the data area name with the name of the library where the data area is located . For example:

LIB2/DTAARA1

Or, the following value:

- *PRODLIB—Specifies your iCluster installation library.

Example

```
DMLOGENT GROUP(GRP1) ENTTYP(20) DTAARA(LIB3/DTAARA2)
```

A logging journal entry is placed in each of the journals that is scraped by **GRP1**. The logging entry type is 20.

The contents of **DTAARA2** in library **LIB3** are recorded in the log file.

Restrictions

None

Minimum Authority Level

*OPERATOR

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **75**

DMACTSPLF (Activate Suspended Spooled Files command)

Activates a spooled file that is currently suspended from replication.

```
DMACTSPLF GROUP( ) OUTQ( ) SPLF( ) SPLFNBR( ) JOBNR( ) JOBNAME( ) JOBUER( )
```

```
SRCSYSTEM( ) TGTLIB( ) BACKUP( )
```

After activating a spooled file, mirroring of the spooled file will start if the replication group is active.

If the group is inactive, mirroring starts when the group is started through the DMSTRGRP (iCluster Start Group)

- ! **Note:** The spooled file is refreshed when it is activated.

Input Parameters

GROUP

Specifies the name of the replication group to which the output queue of the spooled file is selected.

OUTQ

The name of the output queue that contains the suspended spooled file to activate.

The library where the output queue resides must be identified. Prefix the output queue with the name of the library where the output queue is located. For example:

LIB2/OUTQ1

SPLF

Specifies the name of the suspended spooled file to activate.

SPLFNBR

Specifies the number of the suspended spooled file to activate.

JOBNAME

Specifies the name of the job that created the suspended spooled file to activate.

JOBUSER

Specifies the user of the job that created the suspended spooled file to activate.

JOBNBR

Specifies the number of the job that created the suspended spooled file to activate.

SRCSYSTEM

Specifies the machine name on which the suspended spooled file to activate was created.

TGTLIB

Specifies the library of the output queue on the backup node to which the suspended spooled file is refreshed when activated.

BACKUP

Specifies the name of the backup node for which the objects will be activated.

If the group is not an *HADR group, you can use the special value *ONLY or you can specify the name of the group's backup node. If the group is an *HADR group, you must name one of the group's backup nodes.

The possible values are:

- <name>—Specifies the name of the group's backup node.
- *ONLY—Specifies that the group's only backup node is the backup node for the object activation. This value is not valid for *HADR groups.

Example

```
DMACTSPLF GROUP(GRP2) OUTQ(LIB2/OUTQ2) SPLF(QPJOBLOG) SPLFNBR(000001)
JOBNAME(MYJOB) JOBUSER(USER1) JOBNBR(654321) SRCSYSTEM(MACHINEA) TGTLIB(LIB3) BACKUP(*ONLY)
```

Activates the specified spooled file for group **GRP2**. The spooled file is in output queue **QUTQ2** in library **LIB2** on the primary node. The spooled file name is **QPJOBLOG** and its spooled file number is **000001**. The spooled file is generated by job with job name **MYJOB**, job user **USER1**, and job number **654321**. The spooled file is generated on machine **MACHINEA** and is refreshed to the output queue in library **LIB3** on the backup node.

Specifies that the group's only backup node is the backup node for the object activation.

Restrictions

This command must be invoked on an active node in the cluster.

Minimum Authority Level

*OPERATOR

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **76**

WRKHASMON, **WRKHATMON**, and **WRKCSMON** menus - Option **8** (Work with Object Status) - Option **10** Work with Suspended Spooled files, then Option **1** to activate the spooled file

DMWRKSSPLF (Work with Suspended Spooled Files command)

Displays the list of suspended spooled files for a group, or a group and a specific output queue.

```
DMWRKSSPLF GROUP( ) OUTQ( ) BACKUP( )
```

After the list of suspended spooled files is displayed, you can then activate selected spooled files.

Input Parameters

GROUP

Specifies the name of an existing replication group.

OUTQ

Specifies the name of an existing output queue that is in the replication scope of the group.

The possible values are:

- <value>—Specifies the name of the output queue whose suspended spooled files are to be displayed.
- *ALL—Specifies that suspended spooled files to be displayed can be in any output queue that is replicated by the group.

The library where the output queue resides must be identified, unless the special value *ALL is specified for the output queue. Prefix the output queue with the name of the library where the output queue is located. For example:

LIB2/OUTQ1

BACKUP

Specifies the name of the backup node for which the suspended spooled files are listed.

If the group is not an *HADR group, you can use the special value *ONLY or you can specify the name of the group's backup node. If the group is an *HADR group, you must name one of the group's backup nodes.

The possible values are:

- <name>—Specifies the name of the group's backup node.
- *ONLY—Specifies that the group's only backup node is the backup node for the suspended spooled file list. This value is not valid for *HADR groups.

Example

```
DMWRKSSPLF GROUP(GRP1) OUTQ(LIB1/OUTQ1) BACKUP(*ONLY)
```

Displays the list of suspended spooled files for group **GRP1** and output queue **OUTQ1** in library **LIB1**.

Specifies that the group's only backup node is the backup node for the suspended spooled file list.

Restrictions

None

Minimum Authority Level

*OPERATOR

Menu Access

WRKHASMON, **WRKHATMON**, and **WRKCSMON** menus - Option **8** (Work with Object Status) - Option **10** Work with Suspended Spooled files

RTVRCVPT (Retrieve Recovery Checkpoint command)

Retrieves the recovery checkpoint position from the journal on the backup node.

```
RTVRCVPT JRNKEY( ) JRN( ) OLDRCV( ) OLDPOS( ) OLDPOSLRG( )
```

You can use the recovery checkpoint to find the approximate backup node journal position corresponding to a given primary node journal position. This journal position can be used as the starting journal position in a recovery situation.



Note: Issue this command only from the command line.

Input Parameters

JRNKEY

Specifies the key of a recovery checkpoint that was specified for the **JRNKEY** parameter of either the DMADDGRP (Add Group) or DMCHGGRP (Change Group) command.

JRN

Specifies the name of the journal on the primary node.

Identify the library where the journal resides. Prefix the journal with the name of the library where the journal is located. For example:

LIB2/JRN1

OLDRCV

Specifies the name of the journal receiver on the primary node.

You need to identify the library where the journal receiver resides. Prefix the journal receiver with the name of the library where the journal receiver is located. For example:

LIB1/JRNRCV1

OLDPOS

Specifies the journal position on the primary node.

OLDPOSLRG

Specifies the journal position of up to 20 digits on the primary node.

Output

The journal receiver and journal position of the recovery checkpoint on the backup system is displayed in the job log.

Example 1

```
RTVRCVPT JRNKEY(RCVRYID1) JRN(LIB1/JRN1) OLDRCV(LIB2/JRNRCV2) OLDPOSLRG(242)
```

Retrieves the recovery checkpoint based on the key **RCVRYID1**, journal **JRN1** in library **LIB1**, journal receiver **JRNRCV2** in library **LIB2**, and journal position 242.

Example 2

```
RTVRCVPT JRNKEY(RCVRYID1) JRN(LIB1/JRN1) OLDRCV(LIB2/JRNRCV2)
OLDPOSLRG(2424546454533345)
```

Retrieves the recovery checkpoint based on the key **RCVRYID1**, journal **JRN1** in library **LIB1**, journal receiver **JRNRCV2** in library **LIB2**, and journal position 2424546454533345.

Restrictions

You issue this command on the backup system as part of a recovery situation; for example, after performing a switchover.

Minimum Authority Level

Any user with access to the iCluster product library.

Menu Access

This command is not available from the iCluster menus.

Related information

[RTVRCVPTR \(Retrieve Recovery Checkpoint \(CL Program\) command\)](#)

RTVRCVPTR (Retrieve Recovery Checkpoint (CL Program) command)

Retrieves the recovery checkpoint position from the journal on the backup node given a journal key, journal, receiver, and position from the primary node.

```
RTVRCVPTR JRNKEY( ) JRN( ) OLDRCV( ) OLDPOS( ) OLDPOSLRG( ) RTNCDE( ) RTNRCVNME( )
RTVRCVLIB( ) RTNPOS( ) RTNPOSLRG( )
```

The journal key (**JRNKEY**) is specified in the DMADDGRP (Add Group) or DMCHGGRP (Change Group) command.

You can use the recovery checkpoint to find the approximate backup node journal position that corresponds to a given primary node journal position. This journal position can be used as the starting journal position in a recovery situation, such as after a switchover.

Input Parameters

JRNKEY

Specifies the key of a recovery checkpoint that was specified for the **JRNKEY** parameter of either a DMADDGRP (Add Group) or DMCHGGRP (Change Group) command.

JRN

Specifies the name of the journal on the primary node.

Identify the library where the journal resides. Prefix the journal with the name of the library where the journal is located. For example:

OLDRCV

Indicates the name of the journal receiver on the primary node.

You need to identify the library where the journal receiver resides. Prefix the journal receiver with the name of the library where the journal receiver is located. For example:

LIB1/JRN1

OLDPOS

Indicates the journal position on the primary node.

OLDPOSLRG

Indicates the journal position of up to 20 digits on the primary node.

Output Parameters

RTNCDE

Indicates if the command succeeded or failed in retrieving a recovery checkpoint.

One of the following values is returned:

- *OK—Indicates that the command completed successfully.
- *ERROR—Indicates that the command did not complete successfully.

RTNRCVNME

Indicates the name of the journal receiver on recovery machine.

RTVRCVLIB

Indicates the library where the journal receiver is located.

RTNPOS

Indicates the recovery checkpoint journal position on the backup machine. This is the journal position that is used in the DMSETPOS (Set Journal Start Position) command.

RTNPOSLRG

Indicates the recovery checkpoint journal position of up to 20 digits on the backup machine. This is the journal position that is used in the DMSETPOS (Set Journal Start Position) command.

Example 1

```
RTVRCVPTR JRNKEY(RCVRYID1) JRN(LIB1/JRN1) OLDRCV(LIB2/JRNRCV2) OLDPOS(262)  
RTNCDE(&RTNVAR) RTNRCVNME(&RCVNMEVAR) RTVRCVLIB(&RCVLIBVAR) RTNPOS(&POSVAR)
```

Retrieves the recovery checkpoint position, based on key RCVRYID1, journal **JRN1** in library **LIB1**, journal receiver **JRNRCV2** in library **LIB2**, and journal position 262.

 **Note:** &RTNVAR, &RCVNMEVAR, &RCVLIBVAR and &POSVAR are CL variables.

Example 2

```
RTVRCVPTR JRNKEY(RCVRYID2) JRN(LIB2/JRN2) OLDRCV(LIB2/JRNRCV2)  
OLDPOSLRG(5612364789525458) RTNCDE(&RTNVAR) RTNRCVNME(&RCVNMEVAR)  
RTVRCVLIB(&RCVLIBVAR) RTNPOSLRG(&POSVARLRG)
```

Retrieves the recovery checkpoint position based on the key RCVRYID2, journal **JRN2** in library **LIB2**, journal receiver **JRNRCV2** in library **LIB2**, and journal position 5612364789525458.

 **Note:** &RTNVAR, &RCVNMEVAR, &RCVLIBVAR and &POSVARLRG are CL variables.

Restrictions

You need to issue this command from a CL program on the backup system as part of a recovery situation, such as after performing a switchover.

Minimum Authority Level

Any user with access to the iCluster product library.

Menu Access

This command must be run from a CL program. It is not available from the menus.

SYNCHATRG (Synchronize Triggers and SQL Triggers command)

Refreshes the native and SQL triggers on one or more database files from this node to one or more backup nodes.

```
SYNCHATRG TARGET( ) GROUP( ) FILE( ) FILELIB( )
```

All existing triggers are removed from the database files on the backup node before the triggers are synchronized and added. This command synchronizes native database file triggers and SQL triggers. This command will disable all synchronized triggers on the backup node or nodes.

Input Parameters

TARGET

The name of the backup node for the files whose triggers are going to be synchronized.

The possible values are:

- <name>—Specifies the name of a valid backup node.
- *ALL—Uses all backup nodes for groups that are replicating from the current node. The **GROUP** parameter is ignored.

GROUP

The name of the group whose triggers are going to be synchronized.

The possible values are:

- <name>—Specify the name of a valid group that replicates from the current node to the specified backup node.
- *ALL—Specifies all groups that are replicating from the current node to the specified backup node.

FILE

The database files whose triggers will be synchronized. These files must be within the replication scope of the group.

The possible values are:

- <name>—Specifies a file name.
- <generic*>—Specifies a generic file name.
- *ALL—Specifies all of the database files that are selected for replication by the group or groups.

FILELIB

The library for the files whose triggers will be synchronized.

The possible values are:

- <name>—The library name.
- *ALL—All libraries within replication scope of the specified group or groups. The **FILE** parameter must be *ALL.

Output

Relevant messages are sent to the event and job logs.

Example 1

```
SYNCHATRG TARGET(*ALL) GROUP(*ALL) FILE(*ALL) FILELIB(*ALL)
```

Synchronizes triggers for all files in any library within the replication scope for all groups that replicate from the current node to any backup node.

Example 2

```
SYNCHATRG TARGET(SITGT) GROUP(*ALL) FILE(*ALL) FILELIB(MYLIB1)
```

Synchronizes triggers for all files in library **MYLIB1** that are replicated by all groups with the current node as their primary node and whose backup node is **SITGT**.

Restrictions

This command must be used on the primary node of the groups. The groups can be active or inactive when issuing the command, but the actual synchronization occurs when the group is active.

Minimum Authority Level

Any user with access to the iCluster product library.

Menu Access

This command is not available from the iCluster menus.

SYNCHACST (Synchronize constraints command)

Use the **SYNCHACST** command to refresh the constraints on one or more database files from this node to one or more backup nodes.

```
SYNCHACST TARGET( ) GROUP( ) FILE( ) FILELIB( ) CSTTYPE( )
```

Depending on the constraint type, the existing constraints are removed from the database files on the backup node before the constraints are synchronized and re-added to the backup node.



Important:

1. This command only synchronizes the constraints themselves other than the objects in a dependency relationship. For referential constraint, please verify that all dependent files and parent files are applicable.
2. This command disables all synchronized referential integrity constraints on the backup nodes.

Input parameters

TARGET

The name of the backup node for the files in which constraints are synchronized. The possible values are:

- <name>—Specifies the name of a valid backup node.
- *ALL—Uses all backup nodes for groups that are replicating from the current node. The GROUP parameter is ignored.

GROUP

The name of the group for which constraints are synchronized. The possible values are:

- <NAME>—Specifies the name of a valid group that replicates from the current node to the specified backup node.
- *ALL—Specifies all groups that are replicating from the current node to the specified backup node.

FILE

The database files for which constraints are synchronized. These files must be within the replication scope of the group. The possible values are:

- <name>—Specifies a file name.
- <generic*>—Specifies a generic file name.
- *ALL—Specifies all of the database files that are selected for replication by the group or groups.

FILELIB

The library for the files for which constraints are synchronized. The possible values are:

- <name>—The library name.
- *ALL—All libraries within replication scope of the specified group or groups. The FILE parameter must be *ALL.

CSTTYPE

Specifies the type of constraint being synchronized to the backup node. The possible values are:

- *ALL—Specifies all kinds of constraints are synchronized to the backup node.
- *REFCST—Only referential constraint is synchronized to the backup node.
- *UNQCST—Only unique constraint is synchronized to the backup node.
- *PRIKEY—Only primary key constraint is synchronized to the backup node.
- *CHKCST—Only check constraint is synchronized to the backup node.

Output

Relevant messages are sent to the event and job logs.

Example 1

```
SYNCHATRG TARGET(*ALL) GROUP(*ALL) FILE(*ALL) FILELIB(*ALL) CSTTYPE(*ALL)
```

Synchronizes all constraints for all files in any library within the replication scope for all groups that replicate from the current node to any backup node.

Example 2

```
SYNCHATRG TARGET(SITGT) GROUP(*ALL) FILE(*ALL) FILELIB(MYLIB1) CSTTYPE(*REFCST)
```

Synchronizes only referential constraints for all files in library MYLIB1 that are replicated by all groups with the current node as their primary node and whose backup node is SITGT.

Restrictions

This command must be used on the primary node of the groups. The groups can be active or inactive when issuing the command, but the actual synchronization occurs when the group is active.

Minimum authority level

Any user with access to the iCluster product library.

Menu access

This command is not available from the iCluster menus.

DMENDRFSH (End Current Refresh command)

Ends the current refresh processing for a database or byte-stream file on the primary or backup nodes of the specified replication group for the specified journals.

```
DMENDRFSH GROUP( ) JRN( ) LOCATION( ) BACKUP( )
```

The objects whose refresh is stopped are suspended with the SBU - Suspended by user - reason code. Replication continues after the terminated refresh.

If the refresh that is stopped is part of a group refresh, the group refresh continues with the next object to be refreshed. If the refresh that is stopped was initiated by an object activation with RFSH(*YES), replication continues with the next journal entry to process.

You can use the DMENDRFSH command to interrupt the refresh of a physical database file (PF-DTA) or byte-stream file (BSF) that is consuming excess CPU and disk resources.



Note: The DMENDRFSH command cannot stop the refresh of a database file being refreshed with the *SAVRST (save-restore) method. Only files being refreshed with the *RBR (record-by-record) method can have their refresh stopped with the DMENDRFSH command. *RBR is the default database file refresh method.

DLO documents are not supported by the DMENDRFSH command.

This command must be invoked on an active node in the cluster.

Input Parameters

GROUP

The name of the replication group for which the object currently being refreshed has to be suspended and its refresh stopped.

JRN

Specifies the name of the database journal that journals the object whose refresh has to be stopped, or *ALL journals. The special value *NONJRN indicates that the object whose refresh has to be stopped is a non-journalized object.

The possible values are:

- <journal-name>—Specifies the name of the database journal. The library where the journal resides must be identified if you do not specify *ALL or *NONJRN. Prefix the journal with the name of the library where the journal is located (for example, LIB1/JRN1).
- *ALL—Ends the current object refresh by all processes for the replication group identified in the GROUP parameter of this command. If no object is currently being refreshed by a journal's processes for the group, the request is ignored for that journal.
- *NONJRN—Ends the current refresh of a non-journalized byte-stream file by the replication group identified in the GROUP parameter of this command.

LOCATION

Specifies whether the current refresh processing should be ended on the primary or backup node of the group identified in the GROUP parameter of this command for the journals identified in the JRN parameter of this command.

The possible values are:

- *APPLY—Specifies that the apply processes on the backup node of the specified group will stop applying staging store entries for the objects currently being refreshed and suspend the objects with the SBU reason code. All further refresh entries for the objects will be discarded, up to the point where processing starts for another object. By default, this value is used.
- *SOURCE—Specifies that the journal scrape processes on the primary node of the specified group will stop sending object refresh entries for the objects currently being refreshed to the backup node. The receive processes on the backup node of the specified group will stop adding refresh entries for the objects to the staging store. The objects will be suspended with the SBU reason code.
- *BOTH—Specifies that all processes working with the objects currently being refreshed will stop the refresh and suspend the objects with the SBU reason code.

BACKUP

Specifies the backup node or nodes on which to end the current refresh.

The possible values are:

- <name>—Specifies the name of the backup node on which to end the current refresh.
- *ONLY— This value is only valid for non-HADR groups. By default, this value is used.
- *BOTH—Specifies that the current object refresh will be ended for both backup nodes. This value is only valid for *HADR groups.

Example

```
DMENDRFSH GROUP(GRP1) JRN(LIB1/JRN1) LOCATION(*SOURCE)
```

Ends the refresh processes for the journal LIB1/JRN1 on the primary node in group **GRP1**.

The refresh processes are suspended with the SBU - Suspended by user - reason code. Replication continues after the terminated refresh.

Restrictions

You can issue this command on any active node in the cluster.

Minimum Authority Level

*OPERATOR

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **67**

Monitor commands

Use the monitor commands to monitor replication, and view and manage monitor history.

Related information

- [WRKCSMON \(Work with Full Cluster Monitor command\)](#)
- [WRKHASMON \(Work with Status Monitor on Primary Node command\)](#)
- [WRKHATMON \(Work with Status Monitor on Backup Node command\)](#)
- [DSPHASMON \(Display Source Monitor command\)](#)
- [CHGHASMON \(Change History Monitor on Primary Node command\)](#)
- [PRGHASMON \(Purge History Monitor on Primary Node command\)](#)
- [WRKHAOBJST \(Work with Object Status command\)](#)
- [WRKHABSFST \(Work with BSF Status command\)](#)
- [DMWRKOBJST \(Work with Object Status by Group command\)](#)

WRKCSMON (Work with Full Cluster Monitor command)

Use on any active node to display the Full Cluster Status Monitor to view the replication status, latency, and replication throughput of all replication groups in the cluster or for a specific named group.

```
WRKCSMON OUTPUT( ) GROUP( )
```

You can view and manage group and object status from the Full Cluster Status Monitor. You can also start and end groups and run sync checks from the Full Cluster Status Monitor. The monitor reporting is based on group and journal combinations and provides real time status of active replication processes and historical inquiries of past activity.

Input Parameters

OUTPUT

Specifies where to send the output from the command.

The possible values are:

- *—Display the output. Includes access to lists of journaled objects that are actively being replicated by a replication group and are not suspended. By default, this value is used.
- *PRINT—Writes the current group status report only to a spooled file. Does not include lists of active, suspended, and out-of-sync objects.

GROUP

The group or groups to be displayed in the monitor output. The possible values are:

- *ALL—Displays the monitor results for all groups in the cluster. By default, this value is used.
- group-name—Displays the monitor results for the specified group.

Example 1

```
WRKCSMON
```

Displays the Full Cluster Status Monitor.

Example 2

```
WRKCSMON GROUP(MYGROUP)
```

Displays the Full Cluster Status Monitor only for the group MYGROUP.

Restrictions

None.

Minimum Authority Level

Any user with access to the iCluster product library.

Menu Access

Main Menu - Option 8

iCluster Commands menu - F22 (Shift + F10) - Option **85**

WRKHASMON (Work with Status Monitor on Primary Node command)

Use on a primary node to display the Primary Status Monitor, where you can view the replication status of replication groups.

```
WRKHASMON
```

Check the object status and open the communication links from the Primary Status Monitor. The command also allows you to start and end groups and run sync checks.

The monitor reporting is based on group and journal combinations, which provide for both real time inquiry of active replication processes and historical inquiry of past activity.

The following statuses and statistics are displayed on the Real Time Overall Latency (Primary) screen:

- Replication status for both active and inactive groups.
- Replication latency on the backup node.
- Primary and backup node journal positions.
- Run time totals for both primary and backup node replication processes.

- Throughput in transactions per hour for the backup node.

Input Parameters

None.

Example

```
WRKHASMON
```

Displays the Primary Status Monitor on the primary node of the groups

Restrictions

You must issue this command on the primary node of the groups you want to monitor.

Minimum Authority Level

Any user with access to the iCluster product library.

Menu Access

Main Menu - Option 80

iCluster Commands menu - F22 (Shift + F10) - Option **80**

Related information

[Monitoring overview](#)

WRKHATMON (Work with Status Monitor on Backup Node command)

Displays the Backup Monitor on a backup node to view the replication status of primary and backup nodes.

```
WRKHATMON
```

The monitor reporting is based on group and journal combinations to provide for real time inquiry of active replication processes and historical inquiry of past activity.

Using this command, you can check the object's status.

The following statuses and statistics may be viewed using the Backup Monitor:

- Replication status for both active and inactive groups.
- Replication latency on the backup node.
- Primary and backup node journal positions.
- Run time totals for both primary and backup node replication processes.
- Throughput in transactions per hour for the backup node.

Input Parameters

None.

Example

```
WRKHATMON
```

Displays the Backup Monitor, allowing the replication status of groups as well as transaction information to be viewed.

Restrictions

You must issue this command on the backup node of the groups to monitor.

Minimum Authority Level

Any user with access to the iCluster product library.

Menu Access

Main Menu - Option **81**

iCluster Commands menu - F22 (Shift + F10) - Option **81**

Related information

[Monitoring overview](#)

DSPHASMON (Display Source Monitor command)

Displays a simplified version of the Status Monitor on the primary system.

```
DSPHASMON
```

This simplified Status Monitor allows you to view only the details and the status of objects for latency, throughput, object position, and journal information. No operational control of iCluster is provided from the resulting screen.

For additional functionality, use the WRKHASMON (Work with Status Monitor on Primary Node) command.

Input Parameters

None.

Output

Relevant messages to the job log.

Example

```
DSPHASMON
```

Displays the simplified Status Monitor.

Restrictions

You can issue this command at any time from the primary node.

Minimum Authority Level

Any user with access to the iCluster product library.

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **82**

CHGHASMON (Change History Monitor on Primary Node command)

Changes the Historical Latency collection process on a primary node. Modify the collection process to capture replication information for different groups with different collection intervals.

```
CHGHASMON TARGET( ) GROUP( ) STRDTE( ) STRTIM( ) POLINT( ) ENDDTE( ) ENDTIM( )
```

Input Parameters

TARGET

The name of the backup node of the group.

GROUP

Specifies the group or groups for replication monitoring.

The primary node in the group must be the system where this command is invoked.

The possible values are:

- <name>—Specifies the group name for replication monitoring.
- *ALL—Specifies all groups replicating to the backup node specified in the TARGET parameter.

STRDTE

The starting date for replication monitoring. If the date precedes the current date, monitoring starts at the current date.

The possible values are:

- <date>—The date to start monitoring in local system date format.
- *TODAY—Starts monitoring on the current date. By default, this value is used.
- *SAME—Keeps the current setting for this parameter.

STRTIM

The starting time for monitoring replication.

The possible values are:

- <time>—A time value from 000010 - 235959 expressed in HHMMSS format.
- *IMMED—Monitoring starts immediately if the starting date is set to *TODAY or an earlier date. If the starting date is set to a later date, monitoring starts at 000000 on the specified date. By default, this value

is used.

- *SAME—Keeps the current setting for this parameter.

If the date specified in the parameter **STRDTE** precedes the current date, monitoring will start immediately.

POLINT

The polling interval for monitoring replication. The monitoring process uses this time interval to take a snapshot of replication activity. You set the interval by specifying the number of hours (first two digits), minutes (middle two digits) and seconds (last two digits) between consecutive polls.

The possible values are:

- <time>—A time value from 000010 - 235959 expressed in *HHMMSS* format. The default value is 001500 (15 minutes).
- *SAME—Keeps the current setting for this parameter.

ENDDTE

The ending date for monitoring replication.

The possible values are:

- <date>—The date to end monitoring in local system date format.
- *TODAY—Ends monitoring on the current date.
- *NONE—Ends monitoring when all replication processes end. By default, this value is used.
- *SAME—Keeps the current setting for this parameter.

Note: The Historical Latency collection process is always active when replication is active.

If the date that is specified precedes the current date, an error message prompts the user to correct the input.

ENDTIM

The ending time for replication monitoring.

The possible values are:

- <time>—A time value from 000000 - 235959 expressed in *HHMMSS* format.
- *NONE—Ends monitoring when all replication processes end or when the CHGHASMON (Change History Monitor on Primary Node) command is invoked. If the end date is set to *TODAY or a date in the future, monitoring will end at 23:59:59 (11:59:59 PM) on that date. By default, this value is used.
- *SAME—Keeps the current setting for this parameter.

Note: The Historical Latency collection process is always active when replication is active.

If the specified time specified precedes the current time, an error message prompts the user to correct the input. However, the time that is specified may precede the current time if the end date is set to a date in the future.

Example

```
CHGHASMON TARGET(NODE2) GROUP(GRP1) STRDTE(*TODAY) STRTIM(163000)
POLINT(013000) ENDDTE('12/23/14') ENDIM(175959)
```

The monitor facility will collect information pertaining to the replication status of group **GRP1** whose backup node is **NODE2**.

Replicated objects are monitored between 4:30 PM on the day the command is issued, and 6:00 PM on December 23, 2014.

Replication activity is captured every 90 minutes within this time interval.

Restrictions

You must issue this command on the primary node of the groups you want to monitor. Do not issue this command on the backup node.

Minimum Authority Level

Any user with access to the iCluster product library.

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **83**

Related information

[Monitoring overview](#)

PRGHASMON (Purge History Monitor on Primary Node command)

Deletes the historical information from the Status Monitor history database on the primary node.

```
PRGHASMON TARGET( ) ENDDTE( ) ENDTIM( )
```

Input Parameters

TARGET

The name of the backup node for which records are removed.

The possible values are:

- <name>—Specifies the name of a backup node.
- *ALL—Specifies that records related to all backup nodes are deleted.

This field requires an entry.

ENDDTE

The latest date for records to be deleted. All records that are timestamped before the specified date and time are purged.

If the date specified follows the current date, all the historical information is deleted.

The possible values are:

- <date>—Specifies the date to end monitoring. The date format must be specified in the user's job attributes.
- *TODAY—Deletes history log records for all dates. By default, this value is used.

ENDTIM

The latest time for records to be deleted. All records that are timestamped before the specified date and time are purged.

The possible values are:

- <time>—A time value from 000000 - 235959 expressed in HHMMSS format.
- *NOW—Specifies the current time as the ending time for records to be deleted. By default, this value is used.

Example

```
PRGHASMON TARGET(NODE2) ENDDTE('12/31/13') ENDTIM(235959)
```

Purges the monitor of iCluster historical records that were written before the end of 2013 and whose backup node is NODE2.

Restrictions

You must issue this command on the primary node of the groups to monitor.

Minimum Authority Level

Any user with access to the iCluster product library.

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **84**

Related information

[Monitoring overview](#)

WRKHAOBJST (Work with Object Status command)

Displays the **Work with Object Status** screen to view the status of objects that are replicated in the group and associated with a particular journal.

```
WRKHAOBJST TARGET( ) GROUP( ) JOURNAL( ) LOCATION( ) SOURCE( )
```

Input Parameters

TARGET

Specifies the backup node of the group.

GROUP

Specifies the name of an existing replication group.

JOURNAL

Specifies the name of an existing journal.

Identify the library where the journal resides. Prefix the journal with the name of the library where the journal is located. For example:

LIB2/JRN1

The possible values are:

- <name>—Specifies the name of the library where the journal is located.
- *PRODLIB—Specifies your iCluster installation library.
- *UNKNOWN—Specifies a library-journal combination with the values *UNKNOWN/*UNKNOWN.

You would specify *UNKNOWN when you want to check whether there are suspended objects associated with unknown journals and libraries.

LOCATION

Specifies whether you want to work with object status on the primary or backup node.

The possible values are:

- *SRC—Indicates to work with object status on the primary node.
- *TGT—Indicates to work with object status on the backup node.

SOURCE

Specifies the name of the primary node where the objects to view the status are located.

This command uses the value for this parameter only when the **LOCATION** parameter is *SRC.

Example

```
WRKHAOBJST TARGET(TGT1) GROUP(GRP1) JOURNAL(LIB1/JRN1)
```

Displays the list of suspended objects for the journal **JRN1**, located in the library **LIB1**, are replicated by the group **GRP1** to backup node **TGT1**. Press F16 to view alternate lists of the active journaled objects and the full list of journaled objects, both active and suspended, for this journal and group.

When the **WRKHAOBJST** command is run on the primary node, the object's source library is displayed. When the command is run on nodes other than the primary node, the object's target library is displayed.

Restrictions

None

Minimum Authority Level

Any user with access to the iCluster product library.

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **80** - Option **8**

Related information

[Monitoring overview](#)

WRKHABSFST (Work with BSF Status command)

Displays the **Work with BSF Status** screen to view the path objects that are replicated by the group and associated

with a particular journal.

```
WRKHABSFST TARGET( ) GROUP( ) JOURNAL( ) LOCATION( ) SOURCE( )
```

Input Parameters

TARGET

Specifies the backup node of the group.

GROUP

Specifies the name of an existing replication group.

JOURNAL

Specifies the name of an existing journal.

Identify the library where the journal resides. Prefix the journal with the name of the library where the journal is located. For example:

LIB2/JRN1

The possible values are:

- *PRODLIB—Specifies your iCluster installation library.
- *UNKNOWN—Specifies a library-journal combination with the values *UNKNOWN/*UNKNOWN.

You would specify *UNKNOWN when you want to check whether there are suspended objects associated with unknown journals and libraries.

- library-name—Specifies the name of the library where the journal is located.

LOCATION

Specifies whether you want to work with object status on the primary or backup node.

The possible values are:

- *SRC—Indicates to work with object status on the primary node. By default, this value is used.
- *TGT—Indicates to work with object status on the backup node.

SOURCE

Specifies the name of the primary node where the objects you want to view the status are located.

This command only uses the value for this parameter when **LOCATION** is *SRC.

Example

```
WRKHABSFST TARGET(TGT1) GROUP(GRP1) JOURNAL(LIB1/JRN1)
```

Displays the list of suspended path objects that are journaled to the journal **JRN**, located in library **LIB1**, and are replicated by the group **GRP1** to backup node **TGT1**. To view the list of active journaled BSF objects for this journal and group, press F16. To see the list of active journaled BSF, the system value **Keep list of active BSF** must be set to *YES and the list of active journaled BSF objects must be populated.

Note: Type 5 in the Opt. column to display BSF pathnames that are greater than 28 characters.

Restrictions

None.

Minimum Authority Level

Any user with access to the iCluster product library.

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **80** - Option **7**

Related information

[Work with BSF/SQL Status views](#)

DMWRKOBJST (Work with Object Status by Group command)

Displays the **Work with Object Status by Group** screen to view suspended and out-of-sync objects for a group.

```
DMWRKOBJST GROUP( ) STRWITH( ) BACKUP( )
```

Use this command on any node in the cluster, but the objects in the list are displayed with their status on the backup node. Native objects that have target-library redirection are displayed with their target library.

Input Parameters

GROUP

The name of the group you want to monitor.

STRWITH

Specifies which view you want to open this screen with. After running this command, you can change views by pressing **F16**.

The possible values are:

- *NTV—Starts with the native object view. By default, this value is used.
- *BSF—Starts with the BSF object view.

BACKUP

Specifies the name of the backup node to which the objects in the group are replicated.

If the group is not an *HADR group, you can use the special value *ONLY or you can specify the name of the group's backup node. If the group is an *HADR group, you must name one of the group's backup nodes.

The possible values are:

- <name>—Specifies the name of the group's backup node.
- *ONLY—Specifies that the group's only backup node is the backup node for the object activation. This value is not valid for *HADR groups.

Example

```
DMWRKOBJST GROUP(GRP1) STRWITH(*NTV) BACKUP(*ONLY)
```

Displays the suspended and out-of-sync objects in the **GRP1** group. When the screen first appears, it shows native objects. To view the status of BSF objects in the **GRP1** group, press F16 (Shift + F4).

- ! **Note:** Type **5** in the Opt. column to display BSF pathnames that are greater than 28 characters.

Specifies that the group's only backup node is the backup node for the object activation.

Restrictions

None.

Minimum Authority Level

Any user with access to the iCluster product library.

Menu Access

Enter Option **8** on a group entry in the WRKHATMON, WRKHASMON, or WRKCSMON display.

Sync check commands

Use the sync check commands to configure and run sync checks. A sync check verifies that objects on the backup node are the same as the objects on the primary node within a replication group.

Related information

[SELSCATTR \(Select Sync Check Attributes command\)](#)

[DMSCATTR \(Select Sync Check Attributes for a Group command\)](#)

[DMSTRSC \(Start Group Sync Check command\)](#)

[DMSTRSCUSR \(Start User-specified Sync Check command\)](#)

[STRCNSC \(Start Continuous Sync Check command\)](#)

[DMSCRPT \(Sync Check Report command\)](#)

[DSPHASC \(Display Sync Check Results command\)](#)

[ENDHASC \(End Sync Check command\)](#)

[PRGHASC \(Purge Sync Check Results command\)](#)

[DMSYNCACT \(Sync Check and Activate command\)](#)

[STRHASC \(Start Group Sync Check command\)](#)

[STRHASCU \(Start User-Specified Sync Check command\)](#)

SELSCATTR (Select Sync Check Attributes command)

Specifies which attributes are checked in a sync check operation.

```
SELSCATTR USEDFT( )
```

Type *NO for the **USEDFT** parameter and then press the **PgDn** key to view and select attributes in the following categories:

- File attributes—These attributes relate to each database file.
- Member attributes—These attributes relate to each member within a database file.
- Record attributes—These attributes relate to the record formats that are contained in each database file.
- Field attributes—These attributes relate to the fields of records that are contained in each database file.
- Object-level attributes—These attributes relate to common object attributes such as creation date and time, ownership and authorities, as well as object-level attributes specific to certain object types. User profile attributes can also be selected or deselected for sync check.

Follow these guidelines for selecting attributes to include in a sync check:

- Select at least one file attribute at the file level.
- If you select attributes at the member level, then the File Member Name attribute is automatically selected for the sync check.
- The number of attributes that you select can affect the performance of the sync check. The more attributes you select, the longer the sync check takes to complete.
- If you are not sure which attributes to select, use the default selections.

You must specify the attributes to include in a sync check before you start the sync check program. The selected attributes affect all sync checks. The list of attributes that are used by the sync check is stored in the database file **SCATTRFILE** in the ICLUSTER product library.

The attributes that are selected by the command are used for the following commands:

- DMSTRSC (Start group sync check)
- DMSTRSCUSR (Start user-specified sync check)
- STRHASC (Start group sync check)
- STRHASCUSR (Start user-specified sync check)
- STRCNSC—Start continuous sync check

Input Parameters

USEDFT

Specifies to use the default attribute selections.

The possible values are:

- *YES—Uses the default attribute selections. Ignores the selected attributes.
- *NO—Uses the attribute selections that are specified on this command. Does not use the default attribute selections.

Press **F9** and then press the **PgDn** key to view the attributes that you can select for a sync check.

Some attributes are always included in a sync check and cannot be deselected. The attributes that are always included are: journaling (for all types of journaled objects), number of based-on files, the list of based-on files, and number of triggers for physical database files. Default attribute selections are indicated in the following list. Select from the following sync check attributes:

File Level Attributes

- Access path indicator
- Access path journal indicator

-
- Access path maintenance (default selection)
 - Access path recovery
 - Access path size indicator
 - Access path type
 - Allocate & Contiguous storage
 - Allow delete indicator
 - Allow read indicator
 - Allow update indicator
 - Allow write indicator
 - Alternate sequence table
 - Auxiliary storage pool
 - Country identifier
 - CCSID for select/omit constant
 - CCSID for text description
 - CCSID for the table
 - Date/time data indicator
 - Dynamic select/omit indicator
 - DBCS Or graphic data
 - DBCS Or graphic literals
 - File generic key field count (default selection)
 - File generic key length (default selection)
 - File level identifier
 - Text description
 - Floating point key indicator
 - Force keyed access path
 - Graphic data indicator
 - Join file type
 - Join logical file indicator
 - Language identifier
 - Logical file indicator (default selection)
 - Maximum delete records
 - Maximum file wait time
 - Maximum key length (default selection)
 - Maximum members (default selection)
 - Maximum number of fields (default selection)
 - Maximum record length (default selection)
 - Maximum record wait time
 - Multiple CCSID indicator
 - Null value data indicator (default selection)
 - Null value duplicate indicator (default selection)
 - Null value key indicator
 - Number of constraints (default selection)
 - Number of data members
 - Number of key fields (default selection)
 - Number of select and omit (default selection)

- Preferred storage unit
- Primary key indicator (default selection)
- Program described indicator
- Public authority at creation
- Record format level check (default selection)
- Record format selector program
- Records to force a write
- Reuse deleted record indicator (default selection)
- Select/omit indicator
- Sort sequence table name
- Source file member name
- Source file name
- Source physical file indicator (default selection)
- Substitute character indicator
- SQL index indicator
- SQL table indicator
- SQL view indicator
- SST weight indicator
- Total number of members (default selection)
- Total number of record formats (default selection)
- Total of Member Sizes(LF only)

The Total of Member Sizes (LF only) attribute only applies to logical files (view and index).

Note: It is normal for a restored logical file to vary from a saved file in this attribute. This behavior is determined by the "how to estimate the maximum size of a database file" strategy used by IBM i for SAVE and RESTORE. Therefore, the restored logical file cannot be synchronized by activation.

IBM i reserves the file size for future growth and to help with performance. The number of bytes in a file does not determine if the size is equal from one system to another. The byte size between systems following a SAVE and RESTORE can vary.

For more information go to the [IBM Knowledge Center](#) and search with the key words, "Database file sizes".

- Trigger data
- Unique constraint indicator (default selection)
- UCS2 sort sequence table CCSID
- UCS2 sort sequence table length
- Variable length data indicator
- Version release and mod level
- With check option
- 1st supported version release

Member Level Attributes

- Current number of records (default selection)

-
- Expiration date
 - File attribute
 - File member name (default selection)
 - Join member
 - Member text description
 - Member text description CCSID
 - Number of based-on members
 - Number of constraint indexes
 - Number of deleted records (default selection)
 - Open data path sharing
 - Remote file indicator
 - Source type
 - SQL file type
 - Use reset date

Record Level Attributes

- Number of fields
- Record format ID
- Record format name
- Record length
- Record text description
- Record text description CCSID

Field Level Attributes

- Alternative field name
- Column heading 1
- Column heading 2
- Column heading 3
- Data or object type
- Date and time fields indicator
- Date and time format
- Date and time separator
- Decimal position
- Edit code
- Edit word
- Field column heading CCSID
- Data CCSID
- Field edit word CCSID
- Field length in bytes
- Field name
- Field text description
- Field text description CCSID
- Field usage
- Graphic fields indicator

- Input buffer position
- Internal field name
- Null values allowed
- Null-capable fields indicator
- Number of digits
- Number of DBCS characters
- Output buffer position
- UCS2 displayed field length
- Variable field indicator
- Variable length indicator

Object Level Attributes

The object level attributes are all selected by default.

- Creation date and time (applies only to native library objects that are replicated by save-restore)
- Created by user (PFDTA only)

The "Created by user" attribute is only checked for *FILE objects with the PF-DTA extended attribute, and only if the Physical file refresh method for the object, as determined by the product system value, the group, and the object specifier for the object, evaluates to *SAVRST (save-restore).

- Data area contents
- Data area decimal position
- Data area length
- Data area type
- DLO sensitivity level
- Extended attribute
- IFS object CCSID
- IFS object codepage
- IFS object data size
- IFS object mode
- IFS object type
- Job description

The following Job description attributes are checked if the Job description attribute is set to *YES:

- Initial library list
 - Routing data
 - Request data
- Journaling

The "Journaling" attribute is only checked for journaled objects, including journaled IFS objects, for groups that require the replicated objects to be journaled on the backup node.



Note: The "Journaling" attribute is only used by the *OBJATTR sync check type. The journal of a database file is always checked with *FILEATTR and *FULL sync check, if the "Journal objects on backup" parameter for the file evaluates to *YES. The Journal images and Omitted journal entries attributes are included during replication. These attributes are included in the Journaling part of

-  the sync check for journaled objects whose Journal on backup group attribute is *YES.

- Number of authorized users
- Number of spooled files in an output queue when spooled file replication is enabled
- Object owner
- Primary group
- Printer device settings

The following printer device settings are checked if the Printer device settings attribute is set to *YES:

- Printer device class
 - Printer device type and model
 - LAN attachment
 - LAN remote adapter address
 - Adapter type
 - Adapter connection type
 - Port number
 - Attached non-switchable controller
 - Local location name
 - Remote network identifier
 - Remote location name (SNA or IP)
 - Remote location type (SNA or IP)
 - Network protocol
 - Network protocol address
 - Network IP address
 - Server protocol
 - Server protocol address
 - Server IP address
 - Validation list name and library
- Private authorities
 - SQL object attributes

The attributes of SQL functions, SQL procedures, and SQL variables that are checked if this value is set to *YES:

- Whether a SQL procedure or function is defined in SQL language or by an external routine
- External program or service program name
- External program or service program library
- Naming convention
- Date format
- Date separator
- Time format
- Time separator
- Result sets (only for SQL procedure)
- In parameters (only for SQL procedure)
- Out parameters (only for SQL procedure)
- In and out parameters (only for SQL procedure)

The attributes of SQL sequences that are checked if this value is set to *YES:

-
- Maximum value
 - Minimum value
 - Increment value
 - Order
 - Cycle option
 - Cache option
 - Data type
 - Numeric precision
 - Starting value
- Text description
 - User profile (*USRPRF) status
 - User profile (*USRPRF) security settings
 - The attributes that are checked if the value is set to *YES:
 - No password indicator (for example, is the password *NONE?)
 - Password expiry interval
 - Block password change
 - Local password management
 - Password expiration interval
 - User class
 - Special authorities
 - Group profile
 - Owner of created objects
 - Group authority
 - Group authority type
 - Number of supplemental groups
 - Supplemental groups
 - User expiration date
 - User expiration interval
 - User expiration action
 - User profile (*USRPRF) environmental settings
 - The attributes that are checked if the value is set to *YES:
 - Assistance level
 - Current library
 - Initial menu
 - Limit capabilities
 - Display signon information
 - Limit device sessions
 - Keyboard buffering
 - Maximum allowed storage
 - Scheduling priority
 - Accounting code
 - Message queue delivery
 - Message severity filter
 - Printer device
 - Special environment

- Attention program
- Sort sequence
- Language identifier
- Country identifier
- Coded character set identifier
- Character identifier control
- Locale job attributes
- Locale
- User options
- User entitlement required
- User ID number
- Group ID number
- Home directory

User profile detailed attributes

The following user profile settings are checked if the **User profile environmental settings** value is set to *YES:

- Job description
- Message queue
- Output queue
- Initial program

 **Note:**

- If the *USRPRF environmental settings attribute is set to YES but one or more of the detailed attributes is set to NO, the product will not check the ones that are set to NO.
- If the *USRPRF environmental settings attribute is set to NO but one or more of the detailed attributes is set to YES, the product will check the ones that are set to YES.
- iCluster will never replicate the RBTUSER, RBTADMIN, or RBTNETPT user profiles.

Example 1

```
SELSCATTR USEDFT(*YES)
```

The default values are used in the sync check.

Example 2

```
SELSCATTR USEDFT(*NO)
```

Your selected attributes are applied in the sync check.

Restrictions

You can issue this command on any active node in the cluster.

Minimum Authority Level

Any user with access to the iCluster product library.

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **90**

DMSCATTR (Select Sync Check Attributes for a Group command)

Specifies which attributes are checked in a sync check operation for a single group.

```
DMSCATTR GROUP( ) USEDFT( )
```

Type *NO for the **USEDFT** parameter and press PgDn to view and select attributes in four categories:

- File attributes—These attributes relate to each database file.
- Member attributes—These attributes relate to each member within a database file.
- Record attributes—These attributes relate to the record formats that are contained in each database file.
- Field attributes—These attributes relate to the fields of records that are contained in each database file.
- Object-level attributes—These attributes relate to common object attributes such as creation date and time, ownership and authorities, as well as attributes specific to certain object types.

Follow these guidelines for selecting attributes to include in a sync check:

- Select at least one file attribute at the file level.
- If you select attributes at the member level, then the File Member Name attribute is automatically selected for the sync check.
- The number of attributes that you select can affect the performance of the sync check. The more attributes you select, then the longer the sync check takes to complete.
- If you are not sure which attributes to select, use the default selections.

You must specify the attributes to include in a sync check before you start the sync check program. The selected attributes affect all sync checks for the group. The list of attributes that are used by the sync check is stored in the database file **SCATTRFILE** in the ICLUSTER product library.

The attributes that are selected by the command are used for the DMSTRSC (Start group sync check), DMSTRSCUSR (Start user-specified sync check), STRHASC (Start group sync check), and STRHASCUSR (Start user-specified sync check) commands for the specified group, as well as by continuous sync checks for the group.

Input Parameters

GROUP

The name of a replication group.

USEDFT

Specifies to use the default value for each sync check attribute.

The possible values are:

- *YES—Uses the default attribute selections that are specified on this command. Ignores the selected attributes.

-
- *NO—Uses the attribute selections. Does not use the default attribute selections.
- Press F9 and then press PgDn key to view the attributes that you can select for a sync check.

Some attributes are always included by default in a sync check and cannot be deselected. These attributes are: journaling (for all types of journaled objects), number of based-on files, list of based-on files and number of triggers, for physical database files. Select from the following sync check attributes:

File Level Attributes

- Access path indicator
- Access path journal indicator
- Access path maintenance (default selection)
- Access path recovery
- Access path size indicator
- Access path type
- Allocate & Contiguous storage
- Allow delete indicator
- Allow read indicator
- Allow update indicator
- Allow write indicator
- Alternate sequence table
- Auxiliary storage pool
- Country identifier
- CCSID for select/omit constant
- CCSID for text description
- CCSID for the table
- Date/time data indicator
- Dynamic select/omit indicator
- DBCS Or graphic data
- DBCS Or graphic literals
- File generic key field count (default selection)
- File generic key length (default selection)
- File level identifier
- Object text description
- Floating point key indicator
- Force keyed access path
- Graphic data indicator
- Increment number of records
- Increment number of records
- Initial number of records
- Join file type
- Join logical file indicator
- Language identifier
- Logical file indicator
- Maximum delete records
- Maximum file wait time
- Maximum key length (default selection)

- Maximum members
- Maximum number of fields
- Maximum record length (default selection)
- Maximum record wait time
- Multiple CCSID indicator
- Null value data indicator (default selection)
- Null value duplicate indicator (default selection)
- Null value key indicator
- Number of constraints (default selection)
- Number of data members
- Number of key fields (default selection)
- Number of select and omit
- Preferred storage unit
- Primary key indicator (default selection)
- Program described indicator
- Public authority at creation (default selection)
- Record format level check (default selection)
- Record format selector program
- Records to force a write
- Reuse deleted record indicator (default selection)
- Select/omit indicator
- Sort sequence table name
- Source file member name
- Source file name
- Source physical file indicator (default selection)
- Substitute character indicator
- SQL index indicator
- SQL table indicator
- SQL view indicator
- SST weight indicator
- Total number of members (default selection)
- Total number of record formats (default selection)
- Total of Member Sizes(LF only)

The Total of Member Sizes (LF only) attribute only applies to logical files (view and index).

 **Note:** It is normal for a restored logical file to vary from a saved file in this attribute. This behavior is determined by the "how to estimate the maximum size of a database file" strategy used by IBM i for SAVE and RESTORE. Therefore, the restored logical file cannot be synchronized by activation.

IBM i reserves the file size for future growth and to help with performance. The number of bytes in a file does not determine if the size is equal from one system to another. The byte size between systems following a SAVE and RESTORE can vary.

For more information go to the [IBM Knowledge Center](#) and search with the key words, "Database file sizes".

-
- Trigger data
 - Unique constraint indicator (default selection)
 - UCS2 sort sequence table CCSID
 - UCS2 sort sequence table length
 - Variable length data indicator
 - Version release and mod level
 - With check option
 - 1st supported version release

Member Level Attributes

- Current number of records (default selection)
- Expiration date
- Extended attribute
- File member name (default selection)
- Join member
- Member text description
- Member text description CCSID
- Number of based-on members
- Number of constraint indexes
- Number of deleted records (default selection)
- Open data path sharing
- Remote file indicator
- Source type
- SQL file type
- Use reset date

Record Level Attributes

- Number of fields
- Record format ID
- Record format name
- Record length
- Record text description
- Record text description CCSID

Field Level Attributes

- Alternative field name
- Column heading 1
- Column heading 2
- Column heading 3
- Data or object type
- Date and time fields indicator
- Date and time format
- Date and time separator

-
- Decimal position
 - Edit code
 - Edit word
 - Field column heading CCSID
 - Data CCSID
 - Field edit word CCSID
 - Field length in bytes
 - Field name
 - Field text description
 - Field text description CCSID
 - Field usage
 - Graphic fields indicator
 - Input buffer position
 - Internal field name
 - Null values allowed
 - Null-capable fields indicator
 - Number of digits
 - Number of DBCS characters
 - Output buffer position
 - UCS2 displayed field length
 - Variable field indicator
 - Variable length indicator

Object Level Attributes

The object level attributes are all selected by default.

- Creation date and time (applies only to native library objects that are replicated by save-restore)
- Created by user (PFDTA only)

The "Created by user" attribute is only checked for *FILE objects with the PF-DTA extended attribute, and only if the Physical file refresh method for the object, as determined by the product system value, the group, and the object specifier for the object, evaluates to *SAVRST (save-restore).

- Data area contents
- Data area decimal position
- Data area length
- DLO sensitivity level
- Extended attribute
- IFS object CCSID
- IFS object codepage
- IFS object data size
- IFS object mode
- IFS object type
- Job description

The following Job description attributes are checked if the Job description attribute is set to *YES:

- Initial library list

- Routing data
 - Request data
- Journaling

The "Journaling" attribute is only checked for journaled objects, including journaled IFS objects, for groups that require the replicated objects to be journaled on the backup node.

Note: The "Journaling" attribute is only used by the *OBJATTR sync check type. The journal of a database file is always checked with *FILEATTR and *FULL sync check, if the "Journal objects on backup" parameter for the file evaluates to *YES. The Journal images and Omitted journal entries attributes are included during replication. These attributes are included in the Journaling part of the sync check for journaled objects whose Journal on backup group attribute is *YES.

- Number of authorized users
- Number of spooled files in an *OUTQ object
- Object owner
- Primary group
- Printer device settings

The following printer device settings are checked if the Printer device settings attribute is set to *YES:

- Printer device class
 - Printer device type and model
 - LAN attachment
 - LAN remote adapter address
 - Adapter type
 - Adapter connection type
 - Port number
 - Attached non-switchable controller
 - Local location name
 - Remote network identifier
 - Remote location name (SNA or IP)
 - Remote location type (SNA or IP)
 - Network protocol
 - Network protocol address
 - Network IP address
 - Server protocol
 - Server protocol address
 - Server IP address
 - Validation list name and library
- Private authorities
 - SQL object attributes
 - The attributes of SQL functions, SQL procedures, and SQL variables that are checked if this value is set to *YES:
 - Whether a SQL procedure or function is defined in SQL language or by an external routine
 - External program or service program name
 - External program or service program library
 - Naming convention

- Date format
- Date separator
- Time format
- Time separator
- Result sets (only for SQL procedure)
- In parameters (only for SQL procedure)
- Out parameters (only for SQL procedure)
- In and out parameters (only for SQL procedure)

The attributes of SQL sequences that are checked if this value is set to *YES:

- Maximum value
- Minimum value
- Increment value
- Order
- Cycle option
- Cache option
- Data type
- Numeric precision
- Starting value

- Text description
- User profile (*USRPRF) status
- User profile (*USRPRF) security settings

The attributes that are checked if the value is set to *YES:

- No password indicator (for example, is the password *NONE?)
- Password expiry interval
- Block password change
- Local password management
- Password expiration interval
- User class
- Special authorities
- Group profile
- Owner of created objects
- Group authority
- Group authority type
- Number of supplemental groups
- Supplemental groups
- User expiration date
- User expiration interval
- User expiration action

- User profile (*USRPRF) environmental settings

The attributes that are checked if the value is set to *YES:

- Assistance level
- Current library
- Initial menu
- Limit capabilities
- Display signon information
- Limit device sessions

- Keyboard buffering
- Maximum allowed storage
- Scheduling priority
- Accounting code
- Message queue delivery
- Message severity filter
- Printer device
- Special environment
- Attention program
- Sort sequence
- Language identifier
- Country identifier
- Coded character set identifier
- Character identifier control
- Locale job attributes
- Locale
- User options
- User entitlement required
- User ID number
- Group ID number
- Home directory

User profile detailed attributes

The following user profile settings are checked if the **User profile environmental settings** value is set to *YES:

- Job description
- Message queue
- Output queue
- Initial program



Note:

- If the *USRPRF environmental settings attribute is set to YES but one or more of the detailed attributes is set to NO, the product will not check the ones that are set to NO.
- If the *USRPRF environmental settings attribute is set to NO but one or more of the detailed attributes is set to YES, the product will check the ones that are set to YES.
- iCluster will never replicate the RBTUSER, RBTADMIN, or RBTNETPT user profiles.

Example 1

```
DMSCATTR GROUP(GRP1) USEDFT(*YES)
```

The default values are used in the sync check for the **GRP1** group.

Example 2

```
DMSCATTR GROUP(GRP1) USEDFT(*NO)
```

Your selected attributes are applied in the sync check for the **GRP1** group.

Restrictions

You can issue this command on any active node of the cluster.

Minimum Authority Level

*USER

Menu Access

iCluster Commands menu - Option **91**

DMSTRSC (Start Group Sync Check command)

Starts the sync check program to verify that objects in the replication group on the backup node are the same as the objects on the primary node.

```
DMSTRSC GROUP( ) SCTYPE( ) OUTPUT( ) EMAILRPT( ) ACTOOS( ) RFSHTYPE( ) MAXACTSIZ( ) MAXACTNUM( )
LOCK( ) SBMJOB( )
DLTOBSOBJ( ) SENDTO( ) RUNCHKSUM( ) RUNCHKOBSL( ) REPAIR( ) CHKSUS( )
```

All of the objects in the replication group are checked. By comparison, the DMSTRSCUSR (Start user-specified sync check) command performs a sync check on a subset of the objects that are replicated in the group.

Perform the following types of sync checks:

- The object attribute (*OBJATTR) sync check compares the object attributes.
- The file attribute (*FILEATTR) sync check compares replicated *FILE objects and the attributes of the primary and backup node *FILE objects.
- The full (*FULL) sync check compares both the object attributes and the file attributes and does a *CHKOBsolete sync check.
- The file contents (*CHECKSUM) sync check compares the contents of physical database files, IFS, and DLO objects.
- The obsolete object (*CHKOBsolete) sync check looks for objects that exist on the backup node but do not exist on the primary node.

How often you perform a sync check depends on how frequently you want to confirm that the primary and backup node objects are the same. By using the DMSCATTR (Select Sync Check Attributes for a Group) or the SELSCATTR (Select Sync Check Attributes) commands, you can indicate the attributes to include in a sync check. The more attributes that you include in the sync check, the longer the sync check takes.

Input Parameters

GROUP (Group name)

The name of the group for which sync check is to be performed or the special value *ALLSEQ.

***ALLSEQ**—Indicates that a sync check is run sequentially for all of the groups whose primary node is the node where the DMSTRSC command is invoked. The sync checks for the individual groups are run one after another. A group's sync check is started when the previous group's sync check is complete. The delay specified by the 'Group job start delay' parameter of the iCluster system values is used before starting the next group's sync check. This option cannot be used when another sync check is currently active.

SCTYPE (Sync check type)

Specifies the type of sync check.

The possible values are:

- ***FULL**—Performs an *OBJATTR sync check on all objects that are replicated by the group and a *FILEATTR sync check on database file objects. If the selected group is not active, only an *OBJATTR sync check is run and a message is issued. A file contents sync check and obsolete object sync check can also be run optionally as part of the *FULL sync check if the group is active. By default, this value is used.
- ***FILEATTR**—Checks whether the *FILE objects on the primary node exist on the backup node. It also checks that the attributes of the primary node *FILE objects match the attributes of the backup node *FILE objects.

*FILEATTR sync check requires that the files being checked are journaled and that all of the required database journal scrape and apply processes are active. Source physical files can be sync checked with *FILEATTR sync check if either of the following two requirements are met:

- The file is journaled and there is a scrape and apply process for the journal that is active for the group that is being sync checked.
- The file is not journaled but there is a scrape and apply process active for the default database journal for the group that is being sync checked.

! Notes:

- *FILEATTR sync check will check objects of attribute type PF-SRC only if the group the object belongs to replicates at least one object that is being journaled. For example, having at least one PF-DTA object in the group ensures that *FILEATTR sync checks include the PF-SRC objects.
- *FILEATTR sync checks do not report object contents mismatches when MIRRCNTS(*NO) in the DMSELOBJ (Select Objects to Group) command to specify not to refresh or mirror the object contents with .

- ***OBJATTR**—Checks whether objects exist on the backup node. Also checks their authorities and journaling attributes (for journaled objects). BSF and native objects are checked, as well as certain attributes that are specific for an object type; for example, data area contents.
- ***CHECKSUM**—Checks whether the contents of database *FILE objects, IFS *STMF, and Document Library Object (DLO) documents matching the group's object specifiers and replicated by the group are identical on primary and backup, based on MD5 checksum calculations. The objects on the backup node are first checked to ensure they have the same basic attributes as the objects on the primary node. For database *FILE objects, the basic attributes do not include record counts. For IFS *STMF and DLO, the basic attributes include type and CCSID. For each database *FILE object, IFS *STMF, or DLO document that exists on the backup node and has the same basic attributes as the object on the primary node, iCluster then verifies the contents.

For database *FILE objects, *CHECKSUM sync check only checks files that are selected into a group with MIRRCNTS(*YES) and PFUPDMTD(*RRN) in the DMSELOBJ (Select Objects to Group) command. Requires that the files being checked are journaled, and that all of the group's required database journal scrape and apply processes are active.

For journaled IFS *STMF objects, *CHECKSUM sync check requires that the objects being checked are

journalized, and that all of the group's required database journal scrape and apply processes are active. For non-journalized IFS *STMF and DLO documents, *CHECKSUM sync check requires that the replication group is active and the apply job is active.

- *CHKOBSLETE—Checks for objects that exist on the backup node but do not exist on the primary node.

OUTPUT (Output sync check report)

Specifies whether to generate a sync check report.

The possible values are:

- *MISMATCH—Lists the out-of-sync objects, but does not list the details of the attributes that are mismatched. To view sync check results, use the DSPHASC (Display Sync Check Results)/ command.
- *ALL—Lists the out-of-sync objects, and the details of the attributes that are mismatched. To view sync check results, use the DSPHASC (Display Sync Check Results)/ command.
- *NONE—Does not generate a spooled file report for the sync check. By default, this value is used. The output results are recorded in the iCluster metadata on the backup node. To view the sync check results, use the DMSCRIPT (Sync Check Report) command.

EMAILRPT (Email summary report)

Specifies whether to send an email containing a summary sync check report.

The possible values are:

- *YES—Send a summary report of the sync check's results in an email note to the email recipients listed in the DMWRKALR panel. The summary report displays the following results:
 - date
 - time
 - duration
 - group name
 - sync check type
 - specifier
 - number of objects checked
 - number of objects failed
 - number of objects not available (without listing individual objects or reason codes)
- *NO—Do not send an email note with the summary report. By default, this value is used.

ACTOOS (Activate all out-of-sync objects)

Specifies whether to activate the out-of-sync (OOS) objects for the group with the specified refresh type after a sync check, unless the objects are suspended with a reason code that prevents auto-reactivation. For example, the DMACTOOS (activate all OOS objects) command is called after a sync check, if this parameter is set to *YES.

Note: This parameter is only used if the **GROUP** parameter is set to a specific group name and not set as *ALLSEQ.

The possible values are:

- *YES Runs the DMACTOOS command when the sync check processing completes.
- *NO: Does not run the DMACTOOS command at the end of sync check processing. By default, this value is used.

RFSHTYPE (Refresh type)

Indicates the type of refresh used for the group's OOS objects.

- ! **Note:** This parameter is only used if the **ACTOOS** parameter is set to *YES.

The possible values are:

- *FULL: Fully activates and refreshes the group's OOS objects to the backup node and removes them from the list of out of sync objects for the group. Physical database files are refreshed using the refresh method used for the objects as part of a group refresh. In other words, the physical file refresh method as determined from the object specifier, group, and product system values.
- *FIXREC: Refreshes only physical file records that are OOS with a *CHECKSUM sync check. The files are activated and removed from the list of OOS objects for the group, provided they are not listed as OOS for another reason.
- *AUTH: Refreshes only the private authorities and ownership of objects that are listed as OOS with the authority differences (AUT), number of authorized users (NAU), object owner (OWN), or primary group (PG) reason code.

The objects are removed from the list of OOS objects for the group, provided the objects are not listed as being out of OOS for another reason besides the authority difference. Objects that are currently suspended are not activated, although their authorities are refreshed.

- *JRN: Refreshes only the journal information of objects that are OOS with the journaling on target (JRN) reason code. The objects are removed from the list of OOS objects for the group, provided they are not listed as being OOS for another reason.

MAXACTSIZ (Maximum activation size)

Specifies the largest object size to include in the activation. The size of an object can affect network performance and introduce mirroring latency.

- ! **Note:** This parameter is only used if the **ACTOOS** parameter (activate all OOS objects) is set to *YES.

The possible values are:

- *max-size*: Specify the maximum size (in KB) of an object. The largest value that you can enter for this field is 52,428,800.
- *NOMAX: Includes objects of all sizes in the activation. This value can have serious performance issues on the primary node if there are very large objects to process.

MAXACTNUM (Maximum number of activation objects)

Specifies the maximum number of objects to include in the activation. The number of objects can affect network performance and introduce mirroring latency.

- ! **Note:** This parameter is only used if the **ACTOOS** parameter is set to *YES.

The possible values are:

- *maximum-number-of-objects*: Specify the maximum number of objects. The largest value that you can enter for this field is 10,000.
- *NOMAX: Includes all objects in the activation. This value can have serious performance issues on the primary node, if there are a very large number of objects to process.

LOCK (Lock files)

Specifies whether to lock the files that are involved in the sync check during a file attribute or file contents sync check. Locking ensures that an accurate sync check takes place between the primary and backup nodes, but it increases the amount of time required to complete the operation.

- Note:** If objects are locked during a sync check, do not initiate a sync check when the objects in question are being heavily used.

It is recommended that the DMSETSVAL (Set Cluster System Values) parameter **LOCKTGT** is set to *YES to ensure that the backup node files are not changed.

The possible values are:

- *YES—Locks files during the sync check. If a file cannot be locked, then the file is not involved in the sync check, and a message is logged in a spooled file. It will be listed in the sync check report under the Out of Sync or In Use Object list with reason code LCK. Because each object is locked while the attributes of the object are retrieved, this option produces a slower sync check as each object is individually locked and then unlocked. However, each object is locked for a short period of time, which means that access to the object is possible while other objects are being checked. Locking files increases the time it takes to complete the sync check.
- *NO—Does not lock files during the sync check. You must ensure that the files are not modified during the sync check. By default, this value is used.

SBMJOB (Submit sync check job)

Specifies to submit a job to the **XDMLCLUSTER** subsystem to run the sync check.

The possible values are:

- *YES—Runs the sync check by a submitted job in the **XDMLCLUSTER** subsystem. By default, this value is used.
- *NO—Runs the sync check in the job where the DMSTRSC (Start group sync check) command is invoked.

DLTOBSOBJ (Delete obsolete objects)

Specifies whether to automatically delete obsolete objects during obsolete object (*CHKOBSSLETE) sync checks. Obsolete objects exist on the backup node but do not exist on the primary node. The *YES option on the DLTOBSOBJ parameter is only available for the *CHKOBSSLETE sync check type, and the *FULL sync check type when the RUNCHKOBSL parameter is set to *YES.

The possible values are:

- *NO—Does not delete obsolete objects on the backup node. By default, this value is used.
- *YES—Deletes obsolete objects on the backup node as part of *CHKOBSSLETE sync check, or *FULL sync check when the RUNCHKOBSL parameter is set to *YES.

SENDTO (Send output to)

Specifies where to send the spooled file results of the sync check. This parameter is ignored if the OUTPUT parameter is *NONE.

- Note:** To send a report to print on a node, that node's LPD port must be available. For example, to send the spooled file results of a sync check from the backup node to the primary node, the LPD port on the primary node must be open. See "Adding the dmcluster entry to the TCP/IP service table" in the *Rocket iCluster Installation Guide*, or check with your network administrator.

The possible values are:

- *ALL—Sends the spooled file results of the sync check from the backup node to all active nodes in the cluster. By default, this value is used.
- *PRIMARY—Sends the spooled file results of the sync check from the backup node to the primary node, as long as the primary node is active.
- *BACKUP—Does not send the spooled file results of the sync check to any other node. The spooled file results exist only on the backup node. .

RUNCHKSUM (Check file contents with)

Specifies whether to check the contents of physical database files, IFS *STMF objects, and DLO (Document Library Objects) that are replicated by the group as part of the full (*FULL) sync check. This parameter applies only to *FULL sync checks.

The possible values are:

- *CLUSTER—Uses the cluster system value assigned for this option in the DMSETVAL command. By default, this value is used.
- *NO—Does not check the contents of physical database files, IFS *STMF objects and DLO (Document Library Objects).
- *YES—Compares the contents of the physical database files, IFS *STMF objects and DLO (Document Library Objects) using the same method that is used for *CHECKSUM sync checks.

RUNCHKOBSL (Check obsolete objects with)

Specifies whether to check for obsolete objects as part of the *FULL sync check. This parameter applies only to full (*FULL) sync checks.

The possible values are:

- *CLUSTER—Uses the cluster system value assigned for this option in the DMSETVAL command. By default, this value is used.
- *NO—Does not check for obsolete objects.
- *YES—Checks for obsolete objects during a *FULL sync check.

REPAIR (Auto-repair)

Specifies whether to automatically repair mismatched objects. Automatic repair occurs only when the replication group is active during sync check.

The possible values are:

- *YES—Repairs mismatched *FILE objects, IFS *STMF objects and DLO (Document Library Objects) after a *CHECKSUM sync check.
- *NO—Does not repair mismatched objects. By default, this value is used.
- *AUTH—Refreshes private authorities when a *FULL or *OBJATTR sync check discovers that authorities (including ownership and primary group) are found to be mismatched.
- *JRN—Automatically corrects journal objects whose journaling is found to be mismatched by a *FULL, *FILEATTR, or *OBJATTR sync check. The repaired object on the backup node is journaled to the same journal as the object on the primary node.
- *CBU—Indicates that database *FILE objects whose "Created by user" attribute is found to be mismatched by a *FULL or *OBJATTR sync check will be refreshed as part of the sync check processing.
- *CRD—Refreshes library (native) objects when their "Creation date and time" attribute is found to be mismatched by a *FULL or *OBJATTR sync check.

CHKSUS (CHECKSUM on suspended objects)

Specifies whether to perform a *CHECKSUM sync check on suspended objects. The *CHECKSUM sync check verifies that database *FILE objects, IFS *STMF objects and DLO (Document Library Objects) match the group specifiers and compares attributes (excluding record counts) and results of MD5 checksum calculations on the

object contents.

The possible values are:

- *YES—Checks suspended *FILE objects, IFS *STMF objects and DLO (Document Library Objects) during a *CHECKSUM sync check.
- *NO—Does not check suspended *FILE objects, IFS *STMF objects and DLO (Document Library Objects). By default, this value is used.

Example

```
DMSTRSC GROUP(GRP1) SCTYPE(*OBJATTR) OUTPUT(*MISMATCH) LOCK(*NO)
SENDTO(*PRIMARY) REPAIR(*JRN)
```

Indicates for an object attribute sync check to begin at the current date and time for the backup node **TGT1** and group **GRP1** for selected objects. The report displays only the objects that are not synchronized and does not lock the files.

The spooled file results of the sync check are sent from the backup node to the primary node.

The journaling of objects on the backup node is automatically repaired so that the objects on the backup node are journaled to the same journal as the objects on the primary node.

Restrictions

None. You can issue this command on any node in the cluster.

Minimum Authority Level

*USER

Menu Access

iCluster Commands menu - **F22 (Shift + F10)** - Option **92**

DMSTRSCUSR (Start User-specified Sync Check command)

Use on any node of a replication group to check if a set of user-specified objects is synchronized between the primary and backup nodes in a group.

```
DMSTRSCUSR GROUP( ) SCTYPE( ) OBJSPEC( ) PATH( )
OUTPUT( ) EMAILRPT( ) LOCK( ) SBMJOB( ) SENDTO( ) REPAIR( )
CHKSUS( ) SELREC( ) STRRRN( ) ENDRRN( ) PERCENT( )
```

This command performs a synchronization check on a single object or a subset of objects that are replicated within the replication group. This user-specified object selection ensures synchronization for a single object that is referenced by that object specifier when a group is replicating an object specifier with the *ALL setting. By comparison, the DMSTRSC (Start group sync check) and STRHASC (Start group sync check) commands perform a sync check on all objects that are selected to the group.

Specify the attributes to include in a sync check with the SELSCATTR (Select Sync Check Attributes) command. The sync check takes longer when more attributes are specified.

Input Parameters

GROUP

The name of a replication group.

SCTYPE

The type of sync check to perform.

The possible values are:

- *FULL—Performs an *OBJATTR sync check on all objects that match the object filter that are replicated by the group and a *FILEATTR sync check on database file objects. If the selected group is not active, only an *OBJATTR sync check is run and a message is issued. By default, this value is used.
- *FILEATTR—Checks whether the *FILE objects on the primary node exist on the backup node. It also checks that the attributes of the primary node *FILE objects match the attributes of the backup node *FILE objects.

*FILEATTR sync check requires that the files being checked are journaled, and that all of the group's required database journal scrape and apply processes are active. Source physical files can be sync checked with *FILEATTR sync check if either of the following two requirements are met:

- If the file is journaled and there is a scrape and apply process for the file's journal active for the group that is being sync checked.
- If the file is not journaled but there is a scrape and apply process active for the default database journal for the group that is being sync checked.

Note: *FILEATTR sync check will check objects of attribute type PF-SRC only if the group the object belongs to replicates at least one object that is being journaled. For example, having at least one PF-DTA object in the group ensures *FILEATTR sync checks will include the PF-SRC objects.

- *OBJATTR—Checks whether objects exist on the backup node. Also checks their authorities and journaling attributes (for journaled objects). BSF and native objects are checked.
- *BSF—Performs an *OBJATTR sync check.
- *DTAARA—Performs an *OBJATTR sync check.
- *CHKJRN—Performs an *OBJATTR sync check.
- *LIST—Performs an *OBJATTR sync check.
- *CHECKSUM—Checks whether the contents of database *FILE objects, IFS *STMF, and Document Library Object (DLO) documents matching the OBJSPEC or PATH parameter and replicated by the group, are identical on primary and backup, based on MD5 checksum calculations. The objects on the backup node are first checked to ensure they have the same basic attributes as the objects on the primary node. For database *FILE objects, the basic attributes do not include record counts. For IFS *STMF and DLO, the basic attributes include type and CCSID. For each database *FILE object, IFS *STMF, or DLO document that exists on the backup node and has the same basic attributes as the object on the primary node, iCluster then verifies the contents.

For database *FILE objects, *CHECKSUM sync check only checks files that are selected into a group with MIRRCNTS(*YES) and PFUPDMTD(*RRN). *CHECKSUM sync check requires that the files being checked are journaled, and that all of the group's required database journal scrape and apply processes are active.

For journaled IFS *STMF objects, *CHECKSUM sync check requires that the objects being checked are journaled, and that all of the group's required database journal scrape and apply processes are active. For non-journaled IFS *STMF and DLO documents, *CHECKSUM sync check requires that the replication group is active and the apply job is active.

OBJSPEC

The set of parameters that filters the native IBM i objects to check.

Object

The name of the objects to include in the sync check.

The possible values are:

- *ALL—Includes all object names in the sync check. By default, this value is used.
- <name>—The name of the object.
- <generic*>—A generic name that identifies one or more objects.

Library

<name>—The name of the library where the objects are located.

Object type

<type>—The type of objects to include in the sync check.

The possible values are:

- <name>—Specifies the object type name.
- *ALL—Includes all object types in the sync check. By default, this value is used.

Object attribute

The attribute of the objects to include in the sync check. Applies only when *FILE or *DEVD is specified for the **Object type** parameter.

The possible values are:

- <attribute>—The object attribute name. Enter any value that conforms to the IBM i naming conventions.

The following values have special meaning to iCluster: PFSRC (source physical file), PFDTA (data physical file), and PF (any physical file). If PF is used, the object specifier will match PFSRC or PFDTA files. Other values listed are standard IBM i file subtypes.

The object specifier that is formed from the name, library, type, and extended attribute components must exactly match an object specifier that is selected to the group, or must be a subset of an object specifier that is selected to the group.

- *ALL—Includes all object attributes in the sync check. By default, this value is used.

PATH

Identifies the location of the BSF objects to include in the sync check for BSF objects.

Enclose the path in single quotation marks ('') and start with a forward slash (/), for example '/Dir3/Dir4/file'. The path can be 2 - 5000 characters.

Generic path names of the form '/mydir*' are supported, where the generic indicator '*' is the final character of the path name. When using generic path names, all subdirectories are included recursively.

Generic file name extensions after the asterisk (*) are supported, where the file name extension follows the period (.) and is 1 - 8 alphabetic characters.

For example, '/mydir/*.txt' and '/mydir/prod*.txt'. File name extension support does not apply to /QDLS pathnames.

OUTPUT

The type of sync check report to generate.

The possible values are:

- *MISMATCH—Lists the out-of-sync objects, but does not list the details of the attributes that are mismatched. To view sync check results, use the DSPHASC (Display Sync Check Results) command.
- *ALL—Lists the out-of-sync objects, and the details of the attributes that are mismatched. To view sync check results, use the DSPHASC (Display Sync Check Results) command.
- *NONE—Does not generate a spooled file report for the sync check. The results of the sync check are recorded in a set of database files on the backup node. Use the DMSCRPT (Sync Check Report) command to view the sync check results in the database files. By default, this value is used.

EMAILRPT

Specifies whether to send an email containing a summary sync check report.

The possible values are:

- *YES—Send a summary report of the sync check's results in an email note to the email recipients listed in the DMWRKALR panel. The summary report displays the following results:
 - date
 - time
 - duration
 - group name
 - sync check type
 - specifier
 - number of objects checked
 - number of objects failed
 - number of objects not available (without listing individual objects or reason codes)
- *NO—Do not send an email note with the summary report. By default, this value is used.

LOCK

Indicates whether to lock the files that are involved in the sync check during a file attribute and file contents sync check. Locking ensures that an accurate sync check takes place between the primary and backup nodes, but increases the time to complete the operation.

-  **Note:** If objects are locked during a sync check, it is recommended that you do not initiate a sync check when the objects in question are being heavily used.

It is recommended that the DMSETSVAL (Set Cluster System Values) parameter **LOCKTGT** is set to *YES to ensure that the backup node files are not changed.

The possible values are:

- *YES—Locks the files during the sync check. If a file cannot be locked, then the file will not be involved in the sync check, and a message is logged in a spooled file. It will be listed in the sync check report under the Out of Sync or In Use Object list with reason code LCK. Because each object is locked while the attributes of the object are retrieved, this option produces a slower sync check as each object is individually locked and then unlocked. However, each object is locked for a short period of time, which means that access to the object is possible while other objects are being checked. Locking files increases the time it takes to complete the sync check.
- *NO—Does not lock files during the sync check. You must ensure that the files are not modified during the sync check. By default, this value is used.

SBMJOB

Specifies to submit a job to the **XDMCLUSTER** subsystem to run the sync check.

The possible values are:

- *YES—Runs the sync check by a submitted job in the **XDMCLUSTER** subsystem. By default, this value is used.
- *NO—Runs the sync check in the job where the **DMSTRSCUSR** (Start user-specified sync check) command is invoked.

SENDTO

Specifies where to send the spooled file results of the sync check. This parameter is ignored if the value of the **OUTPUT** parameter is *NONE.

- ! **Note:** To send a report to print on a node, that node's LPD port must be available. For example, to send the spooled file results of a sync check from the backup node to the primary node, the LPD port on the primary node must be open. See "Adding the **dmcluster** entry to the TCP/IP service table" in the *Rocket iCluster Installation Guide*, or check with your network administrator.

The possible values are:

- *ALL—Send the spooled file results of the sync check from the backup node to all active nodes in the cluster. By default, this value is used.
- *PRIMARY—Send the spooled file results of the sync check from the backup node to the primary node, as long as the primary node is active.
- *BACKUP—Do not send the spooled file results of the sync check to any other node. The spooled file results exist on the backup node only.

REPAIR

Specifies to automatically repair mismatched objects. Automatic repair occurs only when the replication group is active during sync check.

The possible values are:

- *YES—Repairs mismatched *FILE objects, IFS, and DLO objects after a *CHECKSUM sync check.
- *NO—Does not repair mismatched objects. By default, this value is used.
- *AUTH—Refreshes private authorities when authorities (including ownership and primary group) are found to be mismatched by a *FULL or *OBJATTR sync check.
- *JRN—Automatically corrects journal objects whose journaling is found to be mismatched by a *FULL, *FILEATTR, or *OBJATTR sync check. The repaired object on the backup node is journaled to the same journal as the object on the primary node.
- *CBU—Indicates that database *FILE objects whose "Created by user" attribute is found to be mismatched by a *FULL or *OBJATTR sync check will be refreshed as part of the sync check processing.
- *CRD—Refreshes library (native) objects when their "Creation date and time" attribute is found to be mismatched by a *FULL or *OBJATTR sync check.

CHKSUS

Specifies whether to perform a *CHECKSUM sync check on suspended objects.

The possible values are:

- *YES—Checks suspended *FILE objects, IFS, and DLO objects during a *CHECKSUM sync check.
- *NO—Does not check suspended *FILE objects, IFS, and DLO objects. By default, this value is used.

SELREC

Specifies to check a subset of records or a percentage of records of a single database file during a *CHECKSUM sync check.

The possible values are:

- *RRNRNG—Checks a range of records by relative record number. Specify the range of records with the **STRRRN** and the **ENDRRN** parameters. By default, this value is used.
- *PERCENT—Checks a percentage of the records of the file. Specify the percentage with the **PERCENT** parameter.

STRRRN

Specifies the starting relative record number for a *CHECKSUM sync check when the **SELREC** parameter value is *RRNRNG.

The possible values are:

- *FIRST—The first record of the file (relative record number 1). By default, this value is used.
- <number>—The relative record number of the file.

ENDRRN

Specifies the ending relative record number for a *CHECKSUM sync check when the **SELREC** parameter value is *RRNRNG.

The possible values are:

- *LAST—The last record of the file. By default, this value is used.
- <number>—The relative record number of the file.

PERCENT

Specifies a starting point for the sync check and a percentage of a file to check during a *CHECKSUM sync check when the **SELREC** parameter value is *PERCENT.

The possible values are:

- *FIRST—The first record of the file (relative record number 1). By default, this value is used.
- *LAST—The last record of the file.
- <number>—A number between 1 - 100 that specifies the percentage of the records of a file. The default value is

Example

```
DMSTRSCUSR GROUP(GRP1) SCTYPE(*FILEATTR)
OBJSPEC(*ALL LIB1 *FILE PF)
OUTPUT(*MISMATCH) LOCK(*NO)
```

The sync check begins for the group **GRP1** and checks the selected attributes for all *FILE objects in the **LIB1** library. The sync check report lists only the objects that are not synchronized.

Restrictions

You can issue this command on any node of the replication group.

Minimum Authority Level

*USER

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **93**

STRCNSC (Start Continuous Sync Check command)

Starts the continuous sync check program for a currently active replication group.

```
STRCNSC TARGET( ) GROUP( ) SCTYPE( ) DELAY( )
REPAIR( )
```

Input Parameters

TARGET

The name of the backup node for the group.

GROUP

The name of an active replication group.

SCTYPE

The type of continuous sync check to perform.

The possible values are:

- *FILEATTR—Checks whether the *FILE objects on the primary node exist on the backup node. It also checks that the attributes of the primary node *FILE objects match the attributes of the backup node *FILE objects. Objects are not locked during the sync check.
*FILEATTR sync check requires that the files being checked are journaled, and that all of the group's required database journal scrape and apply processes are active. Source physical files can be sync checked with *FILEATTR sync check if either of the following two requirements are met:
 - If the file is journaled and there is a scrape and apply process for the file's journal active for the group that is being sync checked.
 - If the file is not journaled but there is a scrape and apply process active for the default database journal for the group that is being sync checked.

Notes:

- *FILEATTR sync check will check objects of attribute type PF-SRC only if the group the object belongs to replicates at least one object that is being journaled. For example, having at least one PF-DTA object in the group will ensure *FILEATTR sync checks will include the PF-SRC objects.
- *FILEATTR sync checks will not report object contents mismatches if you decide not to refresh or mirror the object contents with MIRRCNTS(*NO) in the DMSELOBJ (Select Objects to Group) command.

- *OBJATTR—Checks whether objects exist on the backup node. Also checks their authorities and journaling attributes (for journaled objects). BSF and native objects are checked.
- *FULL—Performs an *OBJATTR sync check on all objects that are replicated by the group and a *FILEATTR sync check on database file objects. By default, this value is used.

You can optionally run a checksum sync check as part of the continuous *FULL sync check by specifying YES for the Check file contents element of the **FULLSCOPTS** (*FULL sync check options) parameter of the **DMSETSVAL** command.

Similarly, you can optionally run an obsolete object sync check by specifying YES for the Check obsolete objects element of the **FULLSCOPTS** (*FULL sync check options) parameter of the **DMSETSVAL** command.

DELAY

Specifies the delay time in milliseconds between object checks.

This parameter is intended to minimize the interference from sync check processing on regular object replication when the sync check is done through replication jobs. The benefit of doing sync check through the replication jobs is a more accurate sync check, since the sync check is synchronous with object replication. However, sync check can be a CPU-intensive process and might cause unnecessary delays in regular replication processing if there is not a delay between the attribute checking of individual objects.

The possible values are:

- *CLUSTER—Uses the system value that is assigned in the **DMSETSVAL**—Set Cluster System Values command. This is the default value.
- *GROUP—Uses the value that is assigned to the group in the **DMADDGRP** (Add Group) command
- <time>—A value between 0 - 10000 milliseconds.

REPAIR

Specifies to automatically repair objects that are found to be mismatched by a full (*FULL) or object attribute (*OBJATTR) sync check.

The possible values are:

- *NO—Do not automatically repair objects. By default, this value is used.
- *AUTH—Automatically repair the private authorities, ownership, and primary group of objects whose authorities (including ownership and primary group).
- *JRN—Automatically repair the journaling of objects on the backup node so that the objects on the backup node are journaled to the same journal as the objects on the primary node.
- *CBU—Indicates that database *FILE objects whose "Created by user" attribute is found to be mismatched by a *FULL or *OBJATTR sync check, are refreshed as part of the sync check processing.
- *CRD—Refreshes library (native) objects when their "Creation date and time" attribute is found to be mismatched by a *FULL or *OBJATTR sync check.

Example

```
STRCNSC TARGET(TGT1) GROUP(GRP1) SCTYPE(*FULL)
DELAY(*CLUSTER) REPAIR(*JRN)
```

A *FULL sync check will be started to run continuously for the backup node **TGT1** and group **GRP1** for selected objects.

The spooled file results of the sync check are sent from the backup node to the primary node.

The journaling of objects on the backup node is automatically repaired so that the objects on the backup node are journaled to the same journal as the objects on the primary node.

Restrictions

You must issue this command on the primary node of the replication group.

Minimum Authority Level

*USER

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **94**

DMSCRPT (Sync Check Report command)

Use on the backup node in the cluster to produce the iCluster Sync Check Report.

```
DMSCRPT GROUP( ) SORTBY( ) OUTPUT( ) SENDTO( ) DETAILS( ) BACKUP( )
```

The node can be active or inactive. The report lists sync check results only for the backup node where it is issued and for the groups that you specify.

Input Parameters

GROUP

Specifies the group or groups to include in the report. The groups must have been sync checked.

The possible values are:

- <name>—The name of a single group when you run this command on the backup node for the group.
- *LOCAL—All groups that have the current node as their backup node. By default, this value is used.

SORTBY

Specifies how to sort the output of the sync check report.

SORTBY is valid only if the value of the **GROUP** parameter is set to *LOCAL.

The possible values are:

- *GROUP—Sort the sync check results alphabetically by group name. By default, this value is used.
- *OBJECT—Sort the sync check results alphabetically by object name.

OUTPUT

Specifies where to write the report output.

The possible values are:

- *—Displays the report and writes it to a spooled file, when the command is run in interactive mode. Write the report only to a spooled file, when the command is run in batch mode. By default, this value is used.
- *PRINT—Writes the report only to a spooled file.

SENDTO

Specifies where to send the spooled file results of the sync check. This parameter is ignored if the value of the **GROUP** parameter is *LOCAL.

- Note:** To send a report to print on a node, that node's LPD port must be available. For example, to send the spooled file results of a sync check from the backup node to the primary node, the LPD port on the primary node must be open. See "Adding the dmcluster entry to the TCP/IP service table" in the *Rocket iCluster Installation Guide*, or check with your network administrator.

The possible values are:

- *ALL—Send the spooled file results of the sync check from the backup node to all active nodes in the cluster. By default, this value is used.
- *PRIMARY—Send the spooled file results from the backup node to the primary node, as long as the primary node is active.
- *BACKUP—Do not send the spooled file results of the sync check to any other node. The spooled file results exist only on the backup node of the specified group.

- Note:** The sync check report is sent to the output queue that is listed in the CSJOB job description in the ICLUSTER library on the group's backup node. If the output queue listed in the CSJOB job description on the backup node does not exist on the node to which the report is sent, the report is put into the output queue specified for spooled files received through a network (use the DSPNETA command to see the name of this output queue).

DETAILS

Specifies whether to list the details of the mismatched attributes of the out-of-sync objects.

The possible values are:

- *YES—Includes details of the mismatched attributes.
- *NO—Does not include details. By default, this value is used.

BACKUP

The name of the backup node of the group or groups listed in the report.

- Note:** If the backup node named in this parameter is not the current node or *CURRENT, the report will only be available as a spooled file. It will not be displayed on the screen even if the OUTPUT parameter is set to *.

The possible values are:

- Backup node name
- *CURRENT

The sync check report will list the sync check results for the current node as the backup node.

Example

```
DMSCRPT GROUP(*LOCAL) SORTBY(*GROUP) SENDTO(*PRIMARY) OUTPUT(*) BACKUP(*CURRENT)
```

All the groups that have the current node as the backup node are used in the sync check report.

The output of the report is sorted alphabetically by group name.

The report is sent to the primary node.

The output of the report is displayed and written to a spooled file.

Restrictions

The command must be used on the backup node for the groups that you specify in this command.

Minimum Authority Level

*USER

Menu Access

Main Menu - F22 (Shift + F10) - Option **9** - **iCluster Reports** menu - Option **4**

DSPHASC (Display Sync Check Results command)

Accesses the results of the sync check. The command displays a list of spooled files that contain iCluster sync check results.

```
DSPHASC
```

You can display the results of a single sync check by entering option **5** in the field beside a selected spooled file.

Input Parameters

None.

Output

This command displays the list of spooled files that were generated by the sync check.

Example

```
DSPHASC
```

Displays all sync check spooled files on the **Work with All Spooled Files** screen.

Minimum Authority Level

Any user with access to the iCluster product library.

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **96**

ENDHASC (End Sync Check command)

Use the **ENDHASC** command to end any type of sync check program.

```
ENDHASC GROUP( )
```

Input Parameters

GROUP

The name of a replication group.

Example

```
ENDHASC GROUP(GRP1)
```

Indicates that a sync check will end for the group **GRP1**.

Restrictions

This command must be issued on the primary node of the specified group.

Minimum Authority Level

*USER

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **97**

PRGHASC (Purge Sync Check Results command)

Use on the backup node to clear the sync check results database files of obsolete records.

```
PRGHASC PURGE( )
```

Obsolete records are records for groups that no longer exist and objects that are no longer replicated. These records can accumulate when a group is removed or an object specifier is deselected from a group when a backup node is not available. This command also provides an option to clear all records from the sync check database files.

Input Parameters

PURGE

Specifies whether to remove only obsolete records or all records from the sync check database files.

The possible values are:

- *ALL—Purge all records from the sync check results database files.
- *OBsolete—Purge only obsolete records from the sync check results database files. By default, this value is used.

Example

```
PRGHASC PURGE(*ALL)
```

Removes all records from the sync check results database files on this node.

Restrictions

This command must be issued on the node where the sync check results database files are located. This is always the backup node of a group.

Minimum Authority Level

Any user with access to the iCluster product library.

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **98**

DMSYNCACT (Sync Check and Activate command)

Use on any node to perform a *CHECKSUM sync check on physical files in a specified replication group, and repair mismatched records.

The *CHECKSUM sync check compares the contents of physical database files by checking database *FILE objects in the specified replication group and comparing attributes (excluding record counts) and results of MD5 checksum calculations on the object contents. The command repairs mismatched records and activates the files.

```
DMSYNCACT GROUP( ) SELECT( ) BACKUP( )
```

Input Parameters

GROUP

Specifies the name of an active replication group.

SELECT

Specifies the subset of files in the replication group.

The possible values are:

- *RLE—Selects the files that are suspended with the record level error (RLE) reason code, and repairs the mismatched records. By default, this value is used.
- *ALL—Selects all files with an object type of *FILE and object attribute of PFDTA, and repairs the files that are suspended with any reason code.

BACKUP

Specifies the name of the backup node for which the objects will be activated.

If the group is not an *HADR group, you can use the special value *ONLY or you can specify the name of the group's backup node. If the group is an *HADR group, you must name one of the group's backup nodes.

The possible values are:

- <name>—Specifies the name of the group's backup node.
- *ONLY—Specifies that the group's only backup node is the backup node for the object activation. This value is not valid for *HADR groups.

Example

```
DMSYNCACT GROUP(GRP1)
```

Performs a *CHECKSUM sync check on the group **GRP1** and activates the suspended files that have record level errors.

Specifies that the group's only backup node is the backup node for the object activation.

Restrictions

None. You can issue this command on any node in the cluster.

Minimum Authority Level

*USER

Menu Access

iCluster Commands menu - **F22 (Shift + F10)** - Option **99**

STRHASC (Start Group Sync Check command)

Starts a sync check for a group.

```
STRHASC TARGET( ) GROUP( ) SCTYPE( ) DATE( ) TIME( ) OUTPUT( ) EMAILRPT( ) ACTOOS( ) RFSHTYPE( )
MAXACTSIZ( ) MAXACTNUM( )
LOCK( ) SBMJOB( ) DLTOBSOBJ( ) SENDTO( ) RUNCHKSUM( ) RUNCHKOBSL( ) REPAIR( )
CHKSUS( )
```

A sync check verifies that objects on the backup node are the same as the objects on the primary node within a replication group. All objects for the group are checked.

How often you perform a sync check depends on how often you want to check that the primary and backup node objects are the same. To select the attributes to include in a sync check, use the DMSCATTR (Select Sync Check Attributes for a Group) or the SELSCATTR (Select Sync Check Attributes) command. Including more attributes in the sync check extends the sync check.

Input Parameters

TARGET

The name of the backup node for the group on the **GROUP** parameter.

GROUP

The name of the group for which sync check is to be performed or the special value *ALLSEQ.

*ALLSEQ—Indicates that a sync check is run sequentially for all of the groups whose primary node is the node where the DMSTRSC command is invoked. The sync checks for the individual groups are run one after another.

A group's sync check is started when the previous group's sync check is complete. The delay specified by the 'Group job start delay' parameter of the iCluster system values is used before starting the next group's sync check. This option cannot be used when another sync check is currently active.

SCTYPE (Sync check type)

The type of sync check to perform.

- *FULL—Performs an *OBJATTR sync check on all objects that are replicated by the group and a *FILEATTR sync check on database file objects. If the selected group is not active, only an *OBJATTR sync check is run and a message is issued. A file contents sync check and obsolete object sync check can also be run optionally as part of the *FULL sync check if the group is active. By default, this value is used.
- *FILEATTR—Checks whether the *FILE objects on the primary node exist on the backup node and checks that the attributes of the primary node *FILE objects match the attributes of the backup node *FILE objects.
*FILEATTR sync check requires that the files being checked are journaled, and that all of the group required database journal scrape and apply processes are active. Source physical files can be sync checked with *FILEATTR sync check if either of the following two requirements are met:
 - If the file is journaled and there is a scrape and apply process for the file journal active for the group that is being sync checked.
 - If the file is not journaled but there is a scrape and apply process active for the default database journal for the group that is being sync checked.

! Notes:

- *FILEATTR sync check verifies objects of attribute type PF-SRC only if the group that the object belongs to replicates at least one object that is journaled. For example, at least one PF-DTA object in the group ensures *FILEATTR sync checks include the PF-SRC objects.
- *FILEATTR sync checks do not report object content mismatches if you do not refresh or mirror the object contents with MIRRCNTS(*NO) in the DMSELOBJ (Select Objects to Group) command.

- *OBJATTR—Checks whether objects exist on the backup node and verifies their authorities and journaling attributes (for journaled objects), and checks attributes that are specific to certain object types. For example, data area contents. BSF and native objects are checked.
- *DTAARA—Performs an *OBJATTR sync check.
- *BSF—Performs an *OBJATTR sync check.
- *CHKJRN—Performs an *OBJATTR sync check.
- *LIST—Performs an *OBJATTR sync check.
- *CHECKSUM—Checks whether the contents of database *FILE objects, IFS *STMF, and Document Library Object (DLO) documents matching the group's object specifiers and replicated by the group are identical on primary and backup, based on MD5 checksum calculations. The objects on the backup node are first checked to ensure they have the same basic attributes as the objects on the primary node. For database *FILE objects, the basic attributes do not include record counts. For IFS *STMF and DLO, the basic attributes include type and CCSID. For each database *FILE object, IFS *STMF, or DLO document that exists on the backup node and has the same basic attributes as the object on the primary node, iCluster then verifies the contents.

For database *FILE objects, *CHECKSUM sync check only checks files that are selected into a group with MIRRCNTS(*YES) and PFUPDMTD(*RRN) in the DMSELOBJ (Select Objects to Group) command. Requires that the files being checked are journaled, and that all of the group's required database journal scrape and apply processes are active.

For journaled IFS *STMF objects, *CHECKSUM sync check requires that the objects being checked are journaled, and that all of the group's required database journal scrape and apply processes are active. For non-journaled IFS *STMF and DLO documents, *CHECKSUM sync check requires that the replication group is active and the apply job is active.

- *CHKOBSELETE—Checks for objects that exist on the backup node but do not exist on the primary node.

DATE

The date to start the sync check.

The possible values are:

- <date>—A date in local system date format.
- *CURRENT—Perform the sync check when you issue this command. By default, this value is used.

TIME

The time to start the sync check.

The possible values are:

- <time>—A time value expressed in HHMMSS format.
- *CURRENT—Performs the sync check when you issue this command. By default, this value is used.

OUTPUT

The type of sync check report to generate.

The possible values are:

- *MISMATCH—Lists the out-of-sync objects, but does not list the details of the attributes that are mismatched. To view sync check results, use the DSPHASC (Display Sync Check Results) command.
- *ALL—Lists the out-of-sync objects, and the details of the attributes that are mismatched. To view sync check results, use the DSPHASC (Display Sync Check Results) command.
- *NONE—Does not generate a spooled file report for the sync check. Use the DMSCRPT (Sync Check Report) command to view the sync check results. By default, this value is used.

EMAILRPT (Email sync check summary)

Specifies whether to send an email containing a summary sync check report.

The possible values are:

- *YES—Send a summary report of the sync check's results in an email note to the email recipients listed in the DMWRKALR panel. The summary report displays the following results:
 - date
 - time
 - duration
 - group name
 - sync check type
 - specifier
 - number of objects checked
 - number of objects failed
 - number of objects not available (without listing individual objects or reason codes)
- *NO—Do not send an email note with the summary report. By default, this value is used.

ACTOOS (Activate all out-of-sync objects)

Specifies whether to activate the out-of-sync (OOS) objects for the group with the specified refresh type after a sync check, unless the objects are suspended with a reason code that prevents auto-reactivation. For example, the **DMACTOOS** (activate all OOS objects) command is called after a sync check, if this parameter is set to *YES.

- ! **Note:** This parameter is only used if the **GROUP** parameter is set to a specific group name and not set as *ALLSEQ.

The possible values are:

- *YES Runs the **DMACTOOS** command when the sync check processing completes.
- *NO: Does not run the **DMACTOOS** command at the end of sync check processing. By default, this value is used.

RFSHTYPE (Refresh type)

Indicates the type of refresh used for the group's OOS objects.

- ! **Note:** This parameter is only used if the **ACTOOS** parameter is set to *YES.

The possible values are:

- *FULL: Fully activates and refreshes the group's OOS objects to the backup node and removes them from the list of out of sync objects for the group. Physical database files are refreshed using the refresh method used for the objects as part of a group refresh. In other words, the physical file refresh method as determined from the object specifier, group, and product system values.
- *FIXREC: Refreshes only physical file records that are OOS with a *CHECKSUM sync check. The files are activated and removed from the list of OOS objects for the group, provided they are not listed as OOS for another reason.
- *AUTH: Refreshes only the private authorities and ownership of objects that are listed as OOS with the authority differences (AUT), number of authorized users (NAU), object owner (OWN), or primary group (PG) reason code.

The objects are removed from the list of OOS objects for the group, provided the objects are not listed as being out of OOS for another reason besides the authority difference. Objects that are currently suspended are not activated, although their authorities are refreshed.

- *JRN: Refreshes only the journal information of objects that are OOS with the journaling on target (JRN) reason code. The objects are removed from the list of OOS objects for the group, provided they are not listed as being OOS for another reason.

MAXACTSIZ (Maximum activation size)

Specifies the largest object size to include in the activation. The size of an object can affect network performance and introduce mirroring latency.

- ! **Note:** This parameter is only used if the **ACTOOS** parameter (activate all OOS objects) is set to *YES.

The possible values are:

- *max-size*: Specify the maximum size (in KB) of an object. The largest value that you can enter for this field is 52,428,800.
- *NOMAX: Includes objects of all sizes in the activation. This value can have serious performance issues on

the primary node if there are very large objects to process.

MAXACTNUM (Maximum number of activation objects)

Specifies the maximum number of objects to include in the activation. The number of objects can affect network performance and introduce mirroring latency.

-  **Note:** This parameter is only used if the **ACTOOS** parameter is set to *YES.

The possible values are:

- *maximum-number-of-objects*: Specify the maximum number of objects. The largest value that you can enter for this field is 10,000.
- *NOMAX: Includes all objects in the activation. This value can have serious performance issues on the primary node, if there are a very large number of objects to process.

LOCK (Lock files)

Indicates whether to lock the files that are involved in the sync check during a file attribute or file contents sync check. Locking ensures that an accurate synchronization takes place between the primary and backup nodes, but increases the time to complete the operation.

-  **Note:** If objects are locked during a sync check, it is recommended not to initiate a sync check when the objects in question are being heavily used.

To ensure that the backup node files are also not changed by users, the recommended setting for the DMSETSVAL (Set Cluster System Values) parameter **LOCKTGT** is *YES.

The possible values are:

- *YES—Locks files during the sync check. If a file cannot be locked, then the file is not involved in the sync check, it will be listed in the sync check report under the Out of Sync or In Use Object list with reason code LCK. Because each object is locked while the attributes of the object are retrieved, this option produces a slower sync check as each object is individually locked and then unlocked. However, each object is locked for a short period of time, which means that access to the object is possible while other objects are being checked. Locking files increases the time it takes to complete the sync check.
- *NO—Does not lock files during the sync check. You must ensure that the files are not modified during the sync check. By default, this value is used.

SBMJOB (Submit sync check job)

Specifies to submit a job to the **XDMCLUSTER** subsystem to run the sync check.

The possible values are:

- *YES—Runs the sync check by a submitted job in the **XDMCLUSTER** subsystem. By default, this value is used.
- *NO—Runs the sync check in the job where the STRHASC (Start group sync check) command is invoked.

DLTOBSOBJ (Delete obsolete objects)

Specifies whether to automatically delete obsolete objects during *CHKOBSELETE sync checks. Obsolete objects exist on the backup node but do not exist on the primary node. This parameter applies only when the value of the **Enable continuous sync check** (CONT) parameter is *YES. The *YES option on the DLTOBSOBJ parameter is only available for the *CHKOBSELETE sync check type, and the *FULL sync check type when the RUNCHKOBST parameter is set to *YES.

The possible values are:

- *NO—Does not delete obsolete objects on the backup node. By default, this value is used.
- *YES—Deletes obsolete objects on the backup node as part of *CHKOBSELE sync check, or *FULL sync check with the RUNCHKOBSL parameter is set to *YES.

SENDTO (Send output to)

Specifies where to send the spooled file results of the sync check. This parameter is ignored if the OUTPUT parameter is *NONE.

- ⓘ **Note:** To send a report to print on a node, that node's LPD port must be available. For example, to send the spooled file results of a sync check from the backup node to the primary node, the LPD port on the primary node must be open. See "Adding the dmcluster entry to the TCP/IP service table" in the *Rocket iCluster Installation Guide*, or check with your network administrator.

The possible values are:

- *ALL—Send the spooled file results of the sync check from the backup node to all active nodes in the cluster. By default, this value is used.
- *PRIMARY—Send the spooled file results of the sync check from the backup node to the primary node, as long as the primary node is active.
- *BACKUP—Do not send the spooled file results of the sync check to any other node. The spooled file results exist on the backup node only.

RUNCHKSUM (Check file contents with)

Specifies whether to check the contents of physical database files that are replicated by the group as part of the full (*FULL) sync check. This parameter applies only to *FULL sync checks.

The possible values are:

- *CLUSTER—Uses the cluster system value assigned for this option in the DMSETVAL command. By default, this value is used.
- *NO—Does not check for obsolete objects.
- *YES—Compares the contents of the physical database files, IFS, and DLO objects using the same method that is used for *CHECKSUM sync checks.

RUNCHKOBSL (Check obsolete objects with)

Specifies whether to check for obsolete objects as part of the *FULL sync check. This parameter applies only to full (*FULL) sync checks.

The possible values are:

- *CLUSTER—Uses the cluster system value assigned for this option in the DMSETVAL command. By default, this value is used.
- *NO—Does not check for obsolete objects.
- *YES—Checks for obsolete objects during a *FULL sync check.

REPAIR (Auto-repair)

Specifies whether to automatically repair mismatched objects. Automatic repair occurs only when the replication group is active during sync check.

The possible values are:

- *YES—Repair mismatched *FILE objects, IFS, and DLO objects after a *CHECKSUM sync check.
- *NO—Does not repair mismatched objects. By default, this value is used.
- *AUTH—Refresh private authorities when authorities (including ownership and primary group) are found

to be mismatched by a *FULL or *OBJATTR sync check.

- *JRN—Specifies to automatically correct journal objects whose journaling is found to be mismatched by a *FULL, *FILEATTR, or *OBJATTR sync check. The repaired object on the backup node is journaled to the same journal as the object on the primary node.
- *CBU—Indicates that database *FILE objects whose "Created by user" attribute is found to be mismatched by a *FULL or *OBJATTR sync check will be refreshed as part of the sync check processing.
- *CRD—Refreshes library (native) objects when their "Creation date and time" attribute is found to be mismatched by a *FULL or *OBJATTR sync check.

CHKSUS (Perform *CHECKSUM sync check)

Specifies whether to perform a *CHECKSUM sync check on suspended objects.

The possible values are:

- *YES—Checks suspended *FILE objects, IFS, and DLO objects during a *CHECKSUM sync check.
- *NO—Does not check suspended *FILE objects, IFS, and DLO objects. By default, this value is used.

Example

```
STRHASC TARGET(TGT1) GROUP(GRP1) SCTYPE(*OBJATTR) DATE(*CURRENT)
TIME(*CURRENT) OUTPUT(*MISMATCH)
```

An object attribute sync check begins at the current date and time for the backup node **TGT1** and group **GRP1** for the selected objects. The report lists the objects that are not synchronized, but does not list details about the attributes that are out of sync.

Restrictions

You must issue this command on the primary node of the replication group.

Minimum Authority Level

Any user with access to the iCluster product library.

Menu Access

This command is no available from the iCluster menus.

STRHASCUSR (Start User-Specified Sync Check command)

Use on the primary node to check if a set of user-specified objects are synchronized between the primary and backup nodes in a group.

```
STRHASCUSR TARGET( ) GROUP( ) SCTYPE( ) OBJSPEC( ) PATH( ) DATE( )
TIME( ) OUTPUT( ) EMAILRPT( ) LOCK( ) SBMJOB( ) SENDTO( ) REPAIR( )
CHKSUS( ) SELREC( ) STRRRN( ) ENDRRN( ) PERCENT( )
```

This command performs a synchronization check on a single object or a subset of objects that are replicated within the replication group. When a group is replicating an object specifier with the *ALL setting, use this user-specified

object selection to ensure synchronization for a single object that is referenced by an object specifier. By comparison, the STRHASC (Start group sync check) and DMSTRSC (Start group sync check) commands perform a sync check on all objects that are selected to the group.

Specify the attributes to include in a sync check with the SELSCATTR (Select Sync Check Attributes) command. The sync check takes longer when more attributes are specified.

Since this command is run only on the primary node, you can schedule the sync check.

Input Parameters

TARGET

The name of the backup node for the replication group.

GROUP

The name of a replication group.

SCTYPE (Sync check type)

The type of sync check to perform.

The possible values are:

- *FULL—Performs an *OBJATTR sync check on all objects that match the object filter that are replicated by the group and a *FILEATTR sync check on database file objects. If the selected group is not active, only an *OBJATTR sync check is run and a message is issued. By default, this value is used.
- *FILEATTR—Checks whether the *FILE objects on the primary node exist on the backup node. It also checks that the attributes of the primary node *FILE objects match the attributes of the backup node *FILE objects.

*FILEATTR sync check requires that the files that are being checked are journaled, and that all of the group's required database journal scrape and apply processes are active. Source physical files can be sync checked with *FILEATTR sync check, if one of the following requirements are met:

- If the file is journaled and there is a scrape and apply process for the file's journal active for the group that is being sync checked.
- If the file is not journaled but there is a scrape and apply process active for the default database journal for the group that is being sync checked.

The *FILEATTR sync check verifies objects of attribute type PF-SRC only if the group the object belongs to replicates at least one object that is being journaled. For example, having at least one PF-DTA object in the group ensures *FILEATTR sync checks include the PF-SRC objects.

- *OBJATTR—Checks whether objects exist on the backup node. Also checks their authorities and journaling attributes (for journaled objects). BSF and native objects are checked.
- *BSF—Performs an *OBJATTR sync check.
- *DTAARA—Performs an *OBJATTR sync check.
- *CHKJRN—Performs an *OBJATTR sync check.
- *LIST—Performs an *OBJATTR sync check.
- *CHECKSUM—Checks whether the contents of database *FILE objects, IFS *STMF, and Document Library Object (DLO) documents matching the OBJSPEC or PATH parameter and replicated by the group, are identical on primary and backup, based on MD5 checksum calculations. The objects on the backup node are first checked to ensure they have the same basic attributes as the objects on the primary node. For database *FILE objects, the basic attributes do not include record counts. For IFS *STMF and DLO, the basic attributes include type and CCSID. For each database *FILE object, IFS *STMF, or DLO document that exists on the backup node and has the same basic attributes as the object on the primary node, iCluster then verifies the contents.

For database *FILE objects, *CHECKSUM sync check only checks files that are selected into a group with MIRRCNTS(*YES) and PFUPDMTD(*RRN). *CHECKSUM sync check requires that the files being checked are journaled, and that all of the group's required database journal scrape and apply processes are active.

For journaled IFS *STMF objects, *CHECKSUM sync check requires that the objects being checked are journaled, and that all of the group's required database journal scrape and apply processes are active. For non-journaled IFS *STMF and DLO documents, *CHECKSUM sync check requires that the replication group is active and the apply job is active.

OBJSPEC (Filter native IBM i objects)

The set of parameters that filters the native IBM i objects to check.

Object

The name of the objects to include in the sync check.

The possible values are:

- *ALL—Includes all object names in the sync check. By default, this value is used.
- <name>—The name of the object.
- <generic*>—A generic name that identifies one or more objects.

Library

<name>—The name of the library where the objects are located.

Object type

<type>—The type of objects to include in the sync check.

The possible values are:

- <name>—Specifies the object type name.
- *ALL—Includes all object types in the sync check. By default, this value is used.

Object attribute

The attribute of the objects to include in the sync check. Applies only when *FILE or *DEVD is specified for the **Object type** parameter.

The possible values are:

- <attribute>—The object attribute name. Enter any value that conforms to the IBM i naming conventions.

The following values have special meaning to iCluster: PFSRC (source physical file), PFDTA (data physical file), and PF (any physical file). If PF is used, the object specifier will match PFSRC or PFDTA files. Other values listed are standard IBM i file subtypes.

The object specifier that is formed from the name, library, type, and extended attribute components must exactly match an object specifier that is selected to the group, or must be a subset of an object specifier that is selected to the group.

- *ALL—Includes all object attributes in the sync check. By default, this value is used.

PATH (BSF objects location)

Identifies the location of the BSF objects to include in the sync check for BSF objects. Use the **OBJSPEC** parameter to select native IBM i objects.

Enclose the path in single quotation marks ('') and start with a forward slash (/), for example '/Dir3/Dir4/

file'. The path can be 2 - 5000 characters.

Generic path names of the form '/**mydir***' are supported, where the generic indicator '*' is the final character of the path name. When using generic path names, all subdirectories are included recursively.

Generic file name extensions after the asterisk (*) are supported, where the file name extension follows the period (.) and is 1 - 8 alphabetic characters.

For example, '/**mydir**/*.**txt**' and '/**mydir**/**prod***.**txt**'. File name extension support does not apply to /QDLS pathnames.

DATE

The date to start the sync check.

The possible values are:

- <date>—Specifies a particular date in *MMDDYY* format, where *MM* is the month, *DD* is the day, and *YY* is the year.
- *CURRENT—Performs the sync check on the date when you issue this command. By default, this value is used.

TIME

The time to start the sync check.

The possible values are:

- <time>—A time value expressed in *HHMMSS* format.
- *CURRENT—Invokes the sync check at the time when you issue this command. By default, this value is used.

OUTPUT

The type of sync check report to generate.

The possible values are:

- *MISMATCH—Lists the out-of-sync objects, but does not list the details of the attributes that are mismatched. To view sync check results, use the DSPHASC (Display Sync Check Results) command.
- *ALL—Lists the out-of-sync objects, and the details of the attributes that are mismatched. To view sync check results, use the DSPHASC (Display Sync Check Results) command.
- *NONE—Does not generate a spooled file report for the sync check. The results of the sync check are recorded in a set of database files on the backup node. Use the DMSCRPT (Sync Check Report) command to view the sync check results in the database files. By default, this value is used.

EMAILRPT (Email sync check summary)

Specifies whether to send an email containing a summary sync check report.

The possible values are:

- *YES—Send a summary report of the sync check's results in an email note to the email recipients listed in the DMWRKALR panel. The summary report displays the following results:
 - date
 - time
 - duration
 - group name
 - sync check type
 - specifier

- number of objects checked
- number of objects failed
- number of objects not available (without listing individual objects or reason codes)
- *NO—Do not send an email note with the summary report. By default, this value is used.

LOCK (Lock files)

Indicates whether to lock the files that are involved in file attribute and file contents sync checks. Locking ensures that an accurate sync check takes place between the primary and backup nodes, but it increases the time to complete the operation.

- ! **Note:** If objects are locked during a sync check, it is recommended that you do not initiate a sync check when the objects in question are being heavily used.

It is recommended that the DMSETSVAL (Set Cluster System Values) parameter **LOCKTGT** is set to *YES to ensure that the backup node files are also not changed by users.

The possible values are:

- *YES—Locks files during the sync check. If a file cannot be locked, it will be listed in the sync check report under the Out of Sync or In Use Object list with reason code LCK.. Because each object is locked while the attributes of the object are retrieved, this option produces a slower sync check as each object is individually locked and then unlocked. However, each object is locked for a short period of time, which means that access to the object is possible while other objects are being checked. Locking files increases the time it takes to complete the sync check.
- *NO—Does not lock files during the sync check. You must ensure that the files are not modified during the sync check. By default, this value is used.

SBMJOB (Submit job)

Specifies to submit a job to the XDMCLUSTER subsystem to run the sync check.

The possible values are:

- *YES—Runs the sync check by a submitted job in the XDMCLUSTER subsystem. By default, this value is used.
- *NO—Runs the sync check in the job where the STRHASCUSR (Start user-specified sync check) command is invoked.

SENDDTO

Specifies where to send the spooled file results of the sync check. This parameter is ignored if the OUTPUT parameter is *NONE.

- ! **Note:** To send a report to print on a node, that node's LPD port must be available. For example, to send the spooled file results of a sync check from the backup node to the primary node, the LPD port on the primary node must be open. See "Adding the dmcluster entry to the TCP/IP service table" in the *Rocket iCluster Installation Guide*, or check with your network administrator.

The possible values are:

- *ALL—Send the spooled file results of the sync check from the backup node to all active nodes in the cluster. By default, this value is used.
- *PRIMARY—Send the spooled file results of the sync check from the backup node to the primary node, as long as the primary node is active.
- *BACKUP—Do not send the spooled file results of the sync check to any other node. The spooled file

results exist on the backup node only.

REPAIR (Auto-repair)

Specifies to automatically repair mismatched objects. Automatic repair occurs only when the replication group is active during sync check.

The possible values are:

- *YES—Repair mismatched *FILE objects, IFS, and DLO objects after a *CHECKSUM sync check.
- *NO—Does not repair mismatched objects. By default, this value is used.
- *AUTH—Refresh private authorities when authorities (including ownership and primary group) are found to be mismatched by a *FULL or *OBJATTR sync check.
- *JRN—Specifies to automatically correct journal objects whose journaling is found to be mismatched by a *FULL, *FILEATTR, or *OBJATTR sync check. The repaired object on the backup node is journaled to the same journal as the object on the primary node.
- *CBU—Indicates that database *FILE objects whose "Created by user" attribute is found to be mismatched by a *FULL or *OBJATTR sync check will be refreshed as part of the sync check processing.
- *CRD—Refreshes library (native) objects when their "Creation date and time" attribute is found to be mismatched by a *FULL or *OBJATTR sync check.

CHKSUS (*CHECKSUM sync check suspended objects)

Specifies to perform a *CHECKSUM sync check on suspended objects.

The possible values are:

- *YES—Checks suspended *FILE objects, IFS, and DLO objects during a *CHECKSUM sync check.
- *NO—Does not check suspended *FILE objects, IFS, and DLO objects. By default, this value is used.

SELREC (Select records)

Specifies to check only a subset of records or a percentage of records of a single database file during a *CHECKSUM sync check.

The possible values are:

- *RRNRNG—Checks a range of records by relative record number. Specify the range of records with the **STRRRN** and the **ENDRRN** parameters. By default, this value is used.
- *PERCENT—Checks a percentage of the records of the file. Specify the percentage with the **PERCENT** parameter.

STRRRN (Starting relative record number)

Specifies the starting relative record number for a *CHECKSUM sync check when the **SELREC** parameter value is *RRNRNG.

The possible values are:

- *FIRST—Indicates the first record of the file (relative record number 1). By default, this value is used.
- <number>—Specifies the relative record number of the file.

ENDRRN (Ending relative record number)

Specifies the ending relative record number for a *CHECKSUM sync check when the **SELREC** parameter value is *RRNRNG.

The possible values are:

- *LAST—Indicates the last record of the file. By default, this value is used.
- <number>—Specifies the relative record number of the file.

PERCENT

Specifies a starting point for the sync check and a percentage of a file to check during a *CHECKSUM sync check when the **SELREC** parameter value is *PERCENT.

The possible values are:

- *FIRST—Indicates the first record of the file (relative record number 1). By default, this value is used.
- *LAST—Indicates the last record of the file.
- <number>—A number between 1 and 100 that specifies the percentage of the records of a file. The default value is 100.

Example

```
STRHASCUSR TARGET(TGT1) GROUP(GRP1) SCTYPE(*FILEATTR)
OBJSPEC(*ALL LIB1 *FILE PF) DATE(*CURRENT) TIME(*CURRENT)
OUTPUT(*MISMATCH) LOCK(*NO)
```

Indicates that the sync check begins at the current date and time for the backup node **TGT1** and the group **GRP1** and checks the selected attributes for all *FILE objects in the **LIB1** library. The sync check report lists only the objects that are not synchronized.

Restrictions

You must issue this command on the primary node of the replication group.

Minimum Authority Level

*USER

Menu Access

None.

Registered iCluster user commands

Use the registered iCluster user commands to register and manage users.

Related information

- [**DMWRKUSR \(Work with Users command\)**](#)
- [**DMADDUSR \(Add User command\)**](#)
- [**DMCHGUSR \(Change User command\)**](#)
- [**DMRMVUSR \(Remove User command\)**](#)

DMWRKUSR (Work with Users command)

Displays a screen that allows you to work with the user profiles with certain privileges to work with nodes, groups, object specifiers, and other iCluster cluster components on this system.

```
DMWRKUSR
```

From the screen that is displayed, you can add or remove profiles, and change security attributes for a user.

Input Parameters

None.

 **Note:** iCluster will never replicate the RBTUSER, RBTADMIN, or RBTNETPT user profiles.

Example

```
DMWRKUSR
```

Displays a screen that allows you to work with user IDs that have been granted privileges to view or modify iCluster components.

Restrictions

You must issue this command on the node where the operation will be performed.

Minimum Authority Level

QSECOFR or a user profile that has *SECADM authority in the following situations:

- As part of their basic user profile.
- As part of their primary group profile.
- As part of any of their supplemental profiles.

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **100**

DMADDUSR (Add User command)

Registers an IBM i user profile or group profile in iCluster.

```
DMADDUSR USER( ) AUTH( ) DESC( ) PASSWORD( )
```

You must register an IBM i user profile or group profile and identify the authority level (Administrator, Operator, or User) for each profile to define the cluster operations that can be performed.

The user is registered only on the node where this command is issued.

Input Parameters

USER

An IBM i user profile or group profile that is defined on the node where this command is issued.

AUTH

The authority level to assign to the user profile.

The possible values are:

- *USER—Grants user privileges to the user. By default, this value is used.
- *ADMIN—Grants administrative privileges to the user.
- *OPERATOR—Grants operator privileges to the user.

DESC

A short description that allows you to identify this user among all others that have been defined in iCluster. Typically, this description identifies the full name of the user. You can use up to 50 characters.

PASSWORD

The iCluster password to associate with the user profile.

This password is an iCluster password only. For security reasons, it is recommended that the password for the IBM i operating system be different than the password that is used for iCluster.

The possible values are:

- <password>—Specifies a password with a maximum of 10 characters and containing only alphanumeric characters. Passwords are case-sensitive.
- *—Indicates that no password has to be specified with the user name.

A user who does not have a password can still use iCluster commands in the native IBM i environment. This is the default value.

Example 1

```
DMADDUSR USER(MJONES) AUTH(*OPERATOR) DESC('Mary Jones') PASSWORD(LEMONADE)
```

Registers the IBM i user profile MJONES (Mary Jones) in iCluster with operator privileges.

Example 2

```
DMADDUSR USER(BSMITH) AUTH(*ADMIN) DESC('Bruce Smith') PASSWORD(APPLEJUICE)
```

Registers the IBM i user profile BSMITH (Bruce Smith) in iCluster with administrative privileges.

Example 3

```
DMADDUSR USER(GRPPROF) AUTH(*ADMIN) DESC('All members of GRPPROF') PASSWORD(SECRET)
```

Registers the IBM i group profile GRPPROF (all members of this group profile) in iCluster with administrative privileges.

Restrictions

This command cannot be issued by the DMCLUSTER user profile.

Minimum Authority Level

QSECOFR or a user profile that has *SECADM authority in the following situations:

- As part of their basic user profile.
- As part of their primary group profile.
- As part of any of their supplemental profiles.

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **101**

DMCHGUSR (Change User command)

Changes the authority level, password, or description of a user profile or group profile registered to iCluster.

```
DMCHGUSR USER( ) AUTH( ) DESC( ) PASSWORD( )
```

The user profile must have already been added through the DMADDUSR (Add User) command.

The user profile or group profile must exist on the node where this command is issued.

Input Parameters

USER

The iCluster user. This user profile, or group profile, must already be registered with iCluster on the node where this command is issued.

AUTH

The authority level for the user profile.

The possible values are:

- *USER—Changes the authority level to user.
- *ADMIN—Changes the authority level to administrator.
- *OPERATOR—Changes the authority level to operator.
- *SAME—Keeps the current setting for this parameter.

DESC

A short description that typically identifies the full name of the user.

The possible values are:

- <description>—Specifies the description text. You can use up to 50 characters.
- *SAME—Keeps the current setting for this parameter.

PASSWORD

The iCluster password that is associated with the user. You can change the password or keep it the same.

This password is not the same password that is used to log on to the IBM i under the same name. It is an iCluster password, and so it can be different from the current password used to log on to the IBM i system. For security reasons, it is recommended that a different password is used for iCluster.

The possible values are:

- <password>—Specifies a password with a maximum of 10 characters. The first character must be alphabetic, and the others must be alphabetic or numeric. Passwords are case-sensitive.

- *NONE—Specifies no password. A user with no password can still use iCluster commands in the native IBM i environment.
- *—Keeps the current password. By default, this value is used.

Example 1

```
DMCHGUSR USER(MJONES) AUTH(*OPERATOR) DESC('Mary Jones') PASSWORD(ROOTBEER)
```

Changes the authority level of user MJONES (Mary Jones) to operator. Changes the password to ROOTBEER.

Example 2

```
DMCHGUSR USER(BSMITH) AUTH(*ADMIN) DESC('Bruce Smith') PASSWORD(*)
```

Changes the authority level of user BSMITH (Bruce Smith) to administrator. The password is not changed.

Restrictions

This command cannot be issued by the DMCLUSTER user profile.

Minimum Authority Level

QSECOFR or a user profile that has *SECADM authority in the following situations:

- As part of their basic user profile.
- As part of their primary group profile.
- As part of any of their supplemental profiles.

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **102**

DMRMVUSR (Remove User command)

Removes an iCluster user and revokes the user authorization to use iCluster.

```
DMRMVUSR USER( )
```

This command does not delete the IBM i user profile from the node on which it is invoked.

Input Parameters

USER

The user profile to remove. The user profile must already be registered to iCluster on the node where this command is issued.

Example

```
DMRMVUSR USER(MJONES)
```

Removes the user profile MJONES from iCluster.

Restrictions

This command cannot be issued by the DMCLUSTER user profile.

Minimum Authority Level

QSECOFR or a user profile that has *SECADM authority in the following situations:

- As part of their basic user profile.
- As part of their primary group profile.
- As part of any of their supplemental profiles.

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **103**

Utility commands

Use the utility commands to automatically configure iCluster, start the replication monitoring, manage alerts, and retrieve configuration information.

Related information

[DMAUTOCFG \(Auto-configure iCluster command\)](#)
[DMMONHA \(Monitor iCluster Replication command\)](#)
[ENDMONHA \(End iCluster Replication Monitoring command\)](#)
[DMWRKALR \(iCluster Alert Destinations command\)](#)
[DMADDALR \(Add iCluster Alert Destination command\)](#)
[DMCHGALR \(Change iCluster Alert Destination command\)](#)
[DMRMVALR \(Remove iCluster Alert Destination command\)](#)
[RTVHACFGS \(Retrieve HA Configuration Source command\)](#)
[DMMRKSYNC \(Mark Journal Positions and Synchronize Objects command\)](#)
[DMCHKALRD \(Check alert destinations command\)](#)

DMAUTOCFG (Auto-configure iCluster command)

Automatically create nodes, groups, object specifiers for each library, default journals, and sets the product system values. After you run this command, modify the configuration to fit your business requirements.

```
DMAUTOCFG PRMNODE( ) PRMIPADR( ) BCKNODE( ) BCKIPADR( ) PORT( ) PRMGRP( )  
SECGRP( ) SELOBJO( ) DSPLIBSIZE( )
```

By using this command, you do not have to issue separate commands to create nodes, groups, object specifiers for

each library, default journals, or manually set the iCluster system values.

This command automatically completes the following configuration activities:

- Creates the primary and secondary nodes
- Creates the default journals **HADJRN** and **HABSFJRN** and journal receivers **HAD0000001** and **BSF0000001** in the **AAAJRNLIB** library
- Sets the iCluster product system values to use the journals in the **AAAJRNLIB** library as the default journals
- Creates two replication groups with the names that are specified with the **PRMGRP** and **SECGRP** parameters. Both replication groups use the default journals in the **AAAJRNLIB** library.
- Creates the **SYSTEM** group to replicate commonly used system objects
- Creates the following object specifiers and selects them to the **SYSTEM** replication group:
 - All user profiles, except DMCLUSTER
 - All authorization lists (excluding lists with names that begin with the letter Q)
 - All output queues (excluding queues with name that begin with the letter Q)
 - The **QGPL** library (excluding objects with names that begin with the letter Q)

The user libraries are presented to enable you to choose object specifiers for the primary and secondary groups by library. Or you can press F13 to place everything into the primary group.

Add object specifiers to replication groups to turn on object auditing for the selected objects. The last updated date and time is modified on these objects. The next time that the Save Changed Object (SAVCHGOBJ) IBM i command is run on your system, it might save more objects than expected. Do a full system save before the next **SAVCHGOBJ**. For details on using this command to configure a new installation, see the *Rocket iCluster Installation Guide*.

Input Parameters

PRMNODE

Specify the name of the primary node for iCluster. If you are using Cluster Resource Services as the failover mechanism, and the cluster uses multiple versions of the operating system, you must run this command on the node that runs the lowest version of the operating system and specify that node as the primary node. This parameter is required.

PRMIPADR

Specify the IP address or the fully qualified host name of the primary node.

The IP address of the node can be specified in dotted quad notation (142.4.15.133) or in domain name form (prod400.mycorp.com). This parameter is required.

BCKNODE

Specify the name of the backup node for iCluster. This parameter is required.

BCKIPADR

Specify the IP address or the fully qualified host name of the backup node.

The IP address of the node can be specified in dotted quad notation (142.4.15.133) or in domain name form (prod400.mycorp.com). This parameter is required.

PORt

The TCP/IP port number on the node that is reserved for iCluster communications.

This port number was specified when the dmcluster service was defined to the TCP/IP service table.

The default value is 4545.

PRMGRP

Specify the name of the primary iCluster group. The command will display a list of all user libraries (*ALLUSR) on the system. You can use F13 to select all objects in all of these libraries by default to this group, or you will have the option of choosing which libraries contain the objects that should be selected to this group by placing a P (for primary) next to the library name.

SECGRP

Specify the name of the secondary iCluster group.

SELOBJO

Specify whether you want to select object specifiers when running this command.

The possible values are:

- *YES—Runs this command, but only object specifiers are selected. The groups, nodes, and journals must have already been created by a prior run of this command. The command assumes all nodes, groups and journals have been created and will go directly to the display of all user libraries and will allow the user to select which libraries contain the objects that should be selected to which group.
- *NO—Runs the command in its entirety, creating nodes, groups, journals, and sets iCluster system values. By default, this value is used.

DSPLIBSIZE

Specify whether you want to display the size of object specifier libraries and the number of objects in each library when running this command.

-  **Note:** Enabling this option will greatly increase the amount of time it takes to execute the command.

The possible values are:

- *YES—Displays the size of each object specifier library and the number of objects in each library.
- *NO—Does not display the size and number of objects. By default, this value is used.

Example 1

```
DMAUTOCFG PRMNODE(PROD) PRMIPADR('prod.myshop.com') BCKNODE(DRBOX) BCKIPADR('drbox.myshop.com')
PRMGRP(Production) SECGRP(Develop) SELOBJO(*NO) DSPLIBSIZE(*NO)
```

Creates a primary node named **PROD**, a backup node named **DRBOX**, a primary group with the name **Production**, and a secondary group with the name **Develop**.

The command also creates journals and sets default iCluster system values.

Example 2

```
DMAUTOCFG SELOBJO(*YES)
```

Displays a list of libraries to easily select object specifiers. Bypasses adding nodes, configuring groups, and creating journals.

Restrictions

The iCluster product subsystem (XDMCLUSTER) and the DMCHATL job must be active on both nodes.

Minimum Authority Level

*ADMIN

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **110**

DMMONHA (Monitor iCluster Replication command)

Provides a tool to monitor group replication status, event log messages, and suspended objects. You can also specify how out-of-sync objects are monitored.

```
DMMONHA EVTLOG( ) SUSOBJ( ) GRPSTS( ) CLNEVTLOG( ) CHKITV( ) SUSTIM( )
GRPTIM( ) SELMSGID( ) OMITMSGID( ) DESTYPE( ) OOSOBJ( ) OOSTIM( )
```

Use this command to start the replication monitoring, which can send alert messages when the DMMONHA replication monitoring job on the node that owns the iCluster metadata detects a situation that requires operator attention. Valid email alert destinations must be configured for SMTP and registered with the DMWRKALR (Work with iCluster Alert Destinations) or DMADDALR (Add iCluster Alert Destination) commands. You can also specify message queues as destinations for alert messages.

Input Parameters

EVTLOG

Specifies the lowest level of severity of event log messages, or specific event log message IDs, that trigger an alert.

The possible values are:

- *SEV30—Check event log messages with a severity of 30 or higher. By default, this value is used.
- *SEV40—Check event log messages with a severity of 40 or higher.
- *NONE—Check no messages.
- *SELECT—Check event log messages with a severity of 40 or higher, and also messages with the message identifiers (MSGID) that are specified with the **SELMSGID** and **OMITMSGID** parameters.

SUSOBJ

Specifies to monitor suspended objects.

The possible values are:

- *YES—Check suspended objects and send an alert message when a suspended object remains suspended for the time that is specified in the **SUSTIM** parameter. By default, this value is used.
- *NO—Do not check suspended objects.

GRPSTS

Specifies to check group replication status.

The possible values are:

- *YES—Check status of group replication and send an alert message when group status changes from *ACTIVE to *INACTIVE or *IN_DOUBT and stays inactive for the time that is specified in the **GRPTIM** parameter.
- *NO—Do not check group replication status. By default, this value is used.

CLNEVTLOG

Specifies to clear expired messages from the event logs for the active nodes in the cluster. The lifetime of event log messages is set with the **Logged Message Lifetime** system value.

- ! **Note:** The "Maximum event log records" value (one of the Event log parameters in DMSETSVAL (Set Cluster System Values) takes precedence over the **CLNEVTLOG** value. If the size of the event log on a node exceeds the maximum specified in the "Maximum event log records" system value, the event log will be cleared regardless of the **CLNEVTLOG** value.

The possible values are:

- *YES—Remove expired messages from the event logs when the DMMONHA job runs.
- *NO—Do not remove expired messages from the event logs. By default, this value is used.

CHKITV

Specifies a number of minutes to define the interval between the checks that are done by the DMMONHA job.

SUSTIM

Specifies a number of minutes to define the minimum duration for an object to be suspended before the suspended object status is reported in an alert message.

GRPTIM

Specifies a number of minutes to define the minimum duration for a group to be inactive before the inactive group status is reported in an alert message.

SELMSGID

Specifies a list of message identifiers in the event log for monitoring by the DMMONHA job.

The possible values are:

- *ALL—Check all messages of severity 30 or higher. By default, this value is used.
- *SUSPND—Check all messages that indicate an object is suspended.
- *LATNCY—Check all messages that indicate that a replication latency threshold is exceeded.
- <list>—Check messages of any severity that are specified in a list of up to 20 message identifiers.

Messages of severity 40 are always checked when event log checking is enabled, regardless of the contents of the list of message IDs to monitor.

OMITMSGID

Specifies a list of message identifiers in the event log to omit from monitoring by the DMMONHA job.

The possible values are:

- *NONE—Omit no messages. By default, this value is used.
- <list>—Omit messages that are specified in a list of up to 20 message identifiers.

Messages of severity 40 are always checked when event log checking is enabled, regardless of the contents of the list of message IDs to omit from monitoring.

DESTTYPE

Specifies which destinations are used for the alert messages that are generated by the DMMONHA job.

The possible values are:

- *ALL—Sends messages to all of the email address and message queue destinations that are defined with the **DMADDALR** command and are listed in the **DMWRKALR** panel. By default, this value is used.
- *EMAIL—Sends messages only to the email addresses that are listed in the **DMWRKALR** panel.

- *MSGQ—Sends messages only to the message queue destinations that are listed in the **DMWRKALR** panel.

OOSOBJ

Specifies whether out-of-sync objects are monitored.

Possible values include:

- *YES—Checks the out-of-sync objects and sends an alert message when an out-of-sync object remains out of sync for the time specified in the OOSTIM parameter. By default, this value is used.
- *NO—Does not check the out-of-sync objects.

OOSTIM

Specifies the number of minutes to define the maximum duration for an object to be out of sync before the out-of-sync object status is reported in an alert message.

Restrictions

The monitoring job runs only on the node that owns the iCluster metadata. The node that owns the iCluster metadata, and any other node that might potentially become the node that owns the iCluster metadata, must be configured for SMTP to enable email alerts. If the DMMONHA cluster system value is set to enable monitoring, the DMMONHA job starts when the node is started.

Minimum Authority Level

*OPERATOR

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **111**

Example

```
DMMONHA GRPSTS(*YES) GRPTIM(10)
```

Submits a DMMONHA job to the node that owns the iCluster metadata to check group replication status at the default time interval of five minutes. By default *SEV30 and higher messages are checked, suspended objects that have been suspended for more than 15 minutes are checked and the event log is not cleaned.

An alert message is sent if the status of an active group changes to inactive and remains inactive for 10 minutes or longer. The job ends when the node that owns the iCluster metadata is ended, or when the job is ended by the ENDMONHA (End iCluster Replication Monitoring) command.

Monitor iCluster Replication (DMMONHA) log example

After you enter the parameter values and press Enter, the Monitor Event Log might return results such as the following:

```
Monitor event log . . . . . *SEV30      *SEV30, *SEV40, *NONE...
Monitor suspended objects . . . *YES       *YES, *NO
Monitor out of sync objects . . *YES       *YES, *NO
```

Monitor group status	*NO	*YES, *NO
Clean up event logs	*NO	*YES, *NO
Check interval (min)	5	Number of minutes
Objects suspended for (min) . .	15	Number of minutes
Objects out of sync for (min) .	15	Number of minutes
Group inactive or in doubt for	15	Number of minutes
Destination type	*ALL	*EMAIL, *MSGQ, *ALL, *SMS

Related information

[Configuring email for replication monitoring alerts](#)

[DMSETSVAL \(Set Cluster System Values\) command](#)

[DMWRKALR \(iCluster Alert Destinations command\)](#)

ENDMONHA (End iCluster Replication Monitoring command)

Ends replication monitoring, which runs on the node that owns the iCluster metadata.

```
ENDMONHA
```

Input Parameters

None.

Example

```
ENDMONHA
```

Ends the DMMONHA job immediately.

Restrictions

You must issue this command only on the node where the replication monitoring job DMMONHA is running.

Minimum Authority Level

*OPERATOR

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **112**

DMWRKALR (iCluster Alert Destinations command)

Displays a list of alert destinations and their attributes.

The alert destination list has options to add, change, display, and remove alert destinations. Alert email messages are sent when the DMMONHA replication monitoring job detects a situation that requires operator attention.

DMWRKALR

Start the DMMONHA replication monitoring job with the DMMONHA (Monitor iCluster Replication) command or by setting the automatic replication monitoring parameters with the DMSETSVAL (Set Cluster System Values) command, and then ending and restarting the nodes in the cluster.

Input Parameters

None.

Output

Alert destinations and their attributes.

Example

DMWRKALR

Displays a list of alert destinations and their attributes.

Restrictions

The monitoring job runs only on the node that owns the iCluster metadata. This node, and any other node that might become the owner of the iCluster metadata, must be configured for SMTP to enable email alerts. The **DMCLUSTE** directory entry for the DMCLUSTER user profile must have an SMTP user ID and a valid SMTP domain.

Minimum Authority Level

*OPERATOR

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **113**

DMADDALR (Add iCluster Alert Destination command)

Adds an email or message queue alert destination.

DMADDALR ADDRTYPE() ADDRESS() MSGQ() CHARSET() ENCODING()

Input Parameters

ADDRTYPE

Specifies the type of address to receive the alert messages.

The possible values are:

- *EMAIL—Send alert messages by email. By default, this value is used.
- *MSGQ—Send alert messages by message queue.
- *SMS—Send alert messages to email addresses or cell phone numbers as email addresses that will receive a brief summary message (up to 160 bytes in length) instead of the full email message that is normally sent to the email addresses in the iCluster alert destinations list (DMWRKALR).

ADDRESS

A valid destination email address in SMTP format. SMTP must be enabled on the node that owns the iCluster metadata. The **DMCLUSTE** directory entry for the DMCLUSTER user profile must have an SMTP user ID and a valid SMTP domain. This parameter applies only when the address type is *EMAIL.

MSGQ

The message queue for messages from the DMMONHA job. You must also specify the library where the message queue resides. This parameter applies only when the address type is *MSGQ.

CHARSET

Specifies the character set designator for the multipurpose Internet mail extensions (MIME).

The possible values are:

- US-ASCII. By default, this value is used.
- UTF-8
- <value>—Specifies a valid entry for the charset parameter in MIME-format email messages.

ENCODING

Specifies the encoding designator for the Content-Transfer-Encoding field in the MIME-format email messages.

The possible values are:

- 7BIT By default, this value is used.
- 8BIT
- BASE64
- BINARY
- quoted-printable
- ietf-token
- x-token

Example

```
DMADDALR ADDRTYPE(*EMAIL) ADDRESS('username@example.com') CHARSET( ) ENCODING( )
```

Specifies to send alert messages by email. Adds the email address username@example.com to the alert destinations. The alert messages for this destination use the default US-ASCII character set and the default 7-bit encoding.

Restrictions

To enable email alerts, the node that owns the iCluster metadata must be configured for SMTP. The DMCLUSTE directory entry for the DMCLUSTER user profile must have an SMTP user ID and a valid SMTP domain.

For message queue alerts, the content of a message queue entry is limited to 3000 characters in the second-level text. If the content of the message exceeds the capacity of a message queue entry, the list of items in a message queue alert message is truncated. The HAMSGQ message queue in library ICLUSTER cannot be used for the message queue.

Minimum Authority Level

*ADMIN

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **114**

DMCHGALR (Change iCluster Alert Destination command)

Changes the attributes of an email or message queue alert destination.

```
DMCHGALR ADDRESS( ) CHARSET( ) ENCODING( )
```

Input Parameters

ADDRTYPE

Specifies the type of address to receive the alert messages.

The possible values are:

- *EMAIL—Send alert messages by email. By default, this value is used.
- *MSGQ—Send alert messages by message queue.
- *SMS—Send alert messages to email addresses or cell phone numbers as email addresses that will receive a brief summary message (up to 160 bytes in length) instead of the full email message that is normally sent to the email addresses in the iCluster alert destinations list (DMWRKALR).

ADDRESS

Specifies the destination email address.

CHARSET

Specifies the character set designator for the multipurpose Internet mail extensions (MIME).

The possible values are:

- *SAME—Keeps the current setting. By default, this value is used.
- US-ASCII
- UTF-8
- <value>—Specifies a valid entry for the charset parameter in MIME-format email messages.

ENCODING

Specifies the encoding designator for the Content-Transfer-Encoding field in the MIME-format email messages.

The possible values are:

- *SAME—Keeps the current setting. By default, this value is used.
- 7BIT
- 8BIT
- BASE64
- BINARY
- quoted-printable
- ietf-token
- x-token

Example

```
DMCHGALR ADDRESS('username@example.com') CHARSET( ) ENCODING(8BIT)
```

Changes the alert destination for email username@example.com to use 8-bit encoding.

Restrictions

The node that owns the iCluster metadata must be configured for SMTP to enable email alerts. The **DMCLUSTE** directory entry for the **DMCLUSTER** user profile must have an SMTP user ID and a valid SMTP domain.

Minimum Authority Level

*ADMIN

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **115**

DMRMVALR (Remove iCluster Alert Destination command)

Removes an alert destination.

```
DMRMVALR ADDRTYPE( ) ADDRESS( ) MSGQ( )
```

Input Parameters

ADDRTYPE

Specifies the type of address to remove.

The possible values are:

- *EMAIL—Remove an email alert destination. By default, this value is used.
- *MSGQ—Remove a message queue alert destination.

ADDRESS

The email address to remove. This parameter applies only when the address type is *EMAIL.

MSGQ

The message queue to remove. This parameter applies only when the address type is *MSGQ.

Example

```
DMRMLVALR ADDRTYPE(*EMAIL) ADDRESS('username@example.com')
```

Removes the alert destination for an email (username@example.com).

Restrictions

None.

Minimum Authority Level

*ADMIN

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **116**

RTVHACFGS (Retrieve HA Configuration Source command)

Retrieves the CL source statements that describe the existing high availability (HA) configuration. The CL source can be used to recreate the iCluster configuration.

```
RTVHACFGS GROUP( ) PRIMNODE( ) SRCFILE( )
```

This command creates a new source file with members: SETHASVAL, CRTHANODES, CTHAGRPS, RTHAOBJS, CRTHABSFS, CRTJRNMG, CRTSCATTR, CRTUSRINF and CRTALRINF. Each member can be compiled into a CL program to recreate your high availability configuration for system values, nodes, cluster resource groups, native and BSF object specifiers, journal management, sync check attributes, user management, and alert management.

Input Parameters

GROUP

Specify the cluster resource group or groups for which to retrieve configuration information.

The possible values are:

- *ALL—Specifies to retrieve configuration information for all groups on all nodes.
- *PRIMNODE—Specifies to retrieve configuration information for all groups for the node that is specified with the **PRIMNODE** parameter.
- <generic*>—Specifies a generic group name to identify multiple groups.
- <name>—Specifies the name of the group.

PRIMNODE

When the **GROUP** parameter is *PRIMNODE, specify the name of the primary node for the groups whose configuration you want to retrieve.

- <name>—Specifies the name of the primary node.

SRCFILE

Specify the qualified name of the CL source file to generate.

- <name>—Specifies the library and name of the CL source file. Prefix the file name with the name of the library where the file is generated. For example, LIB1/QICLSRC.

Example

```
RTVHACFGS GROUP(SALES*) SRCFILE(MYBCKUP/QICLSRC)
```

Retrieves the current high availability configuration for all groups that start with SALES and creates the CL source file QICLSRC in library MYBCKUP.

Restrictions

None.

Minimum Authority Level

Users with access to the iCluster product library.

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **117**

DMMRKSYNC (Mark Journal Positions and Synchronize Objects command)

Marks journal positions in the iCluster metadata for a specified replication group, synchronizes the objects replicated by the group to the backup node, and starts the group using marked journal positions.

```
DMMRKSYNC GROUP( ) SAVFLIB( ) LIBS( ) SYNCBSF( ) SAVPVTAUT( ) RMTUSRPRF( ) RMTPWD( )
```

```
RTPRMTPWD( ) DLTSAVF( ) CHGOBJAUD( ) STRGRP( )
```

For object synchronization, the objects in the replication scope of the group are saved into savefiles in the specified savefile library. The native objects that can be saved with the SAVOBJ command are saved into save files with the same names of the libraries they are in, the non-DLO BSF objects are saved into a savefile with the name BSFSAVF, and the DLO BSF objects are saved into a savefile with the name DLOSAVF.

The savefiles are sent to the backup node by FTP or DMSNDOBJ (Send Object Immediately), depending on whether a user profile and its password on the backup node for FTP are specified for the command.

On the backup, the objects in replication scope are restored from the savefiles. Before the native objects in a library are restored from the savefile for the library, the library is created on the backup if it does not exist, or cleared if it exists.

The command can only be issued on the group's primary node when the group is *INACTIVE. The group cannot have any target library redirection for its native objects. The group cannot use any IASP redirection for its BSF objects; for example, the primary IASP and backup IASP of the group have to be the same. Native objects in system-owned libraries and BSF objects in system-owned directories cannot be saved and cannot be synchronized with this command.

If the combined length of the names of the DLO included folders that are selected for the group is greater than 5000 characters, they cannot be saved by the DMMRKSYNC command. Also, if the combined length of the names of the excluded DLO folders selected for the group is greater than 5000 characters, the included DLO folders cannot be saved.

There is no check for objects that are excluded from the replication scope of the group but exist on the backup node. If there are such objects in a library on the backup node, they will be deleted when the library is cleared before the saved objects from the primary node are restored into the library.

The processing on the backup node will wait for a savefile to be ready to be restored on the backup node until a timeout period has elapsed. If the timeout is exceeded without the savefile being available, the processing of DMMRKSYNC will be terminated, and the group will not be started. The timeout value is the greater of 2 hours and the iCluster system value "Communications timeout". See the DMSETVAL command for more information on the iCluster system values.

The save processing uses the Object compression product system value that is configured with the DMSETVAL command.

Input Parameters

GROUP

Specifies the name of the group for which the starting journal positions will be marked in the iCluster metadata, and the objects in the replication scope will be synchronized to the backup node. The group must be of type *REPL. This is a required parameter.

SAVFILIB

Specifies the name of the library of the savefiles on the primary node to which you want to save the objects in the replication scope. This is a required parameter.

LIBS

Specifies libraries to synchronize.

The possible values are:

- *ALL—Specifies that all libraries in the replication scope of the group are selected for synchronization. By default, this value is used.
- *NONE—Specifies that no libraries are selected, so no native objects will be synchronized.
- library-list—Up to 5 libraries can be selected for native object synchronization.

Note: If *ALL or *NONE is specified for the parameter, it must be the first and only item in the library list.

SYNCBSF

Specifies whether or not the BSF objects (the objects in the Integrated File System (IFS) and the Document Library Object (DLO) system) that are in the replication scope of the group will be synchronized.

The possible values are:

- *YES—Synchronizes the BSF objects. By default, this value is used.
- *NO—Does not synchronize the BSF objects.

Note: If LIBS(*NONE) is specified, the SYNCBSF parameter must be *YES.

SAVPVTAUT

Specifies whether to save private authorities when saving the native and non-DLO BSF objects in the group's

replication scope. Saving private authorities will increase the amount of time it takes to save the objects, but it can simplify the synchronization of the objects by restoring their private authorities on the backup node when the objects themselves are restored.

The possible values are:

- *NO—No private authorities are saved. By default, this value is used.
- *YES— Private authorities are saved for each native or non-DLO BSF object that is saved. Note that Document Library objects (DLO) can only be saved without their private authorities, even when *YES is specified for the SAVPVTAUT parameter.

You must have save system (*SAVSYS) or all object (*ALLOBJ) special authority to specify this value.

If the OS release of the backup node is lower than that of the primary node, all objects in the group's replication scope will be saved without private authorities, even if you specify SAVPVTAUT(*YES).

If you specify *NO for the SAVPVTAUT parameter, you can run a sync check for the group with SCTYPE(*FULL) REPAIR(*AUTH) to refresh the private authorities of the objects on the backup node, after the group is started from the marked journal positions.

RMTUSRPRF

Specifies the user profile on the backup node as the FTP user profile for sending the savefiles to the backup node via FTP.

The possible values are:

- *NONE—Specifies that no user profile is provided as the FTP user profile on the backup node. The savefiles will be sent to the backup node using the DMSNDOBJ command. By default, this value is used.
- user-profile-name— Specifies the name of the user profile on the backup node as the FTP user profile.

RMTPWD

Specifies the password of the Remote user profile (specified by parameter RMTUSRPRF) on the backup node. The Remote user profile and Remote user password are used in combination when the savefiles are sent to the backup system via FTP.

The possible values are:

- remote-user-password—Specifies a string up to 10 characters as the password of the Remote user profile on the backup node.

This parameter is ignored if RMTUSRPRF(*NONE) is specified.

RTPRMTPWD

Type the Remote user password in the "Remote user password" (RMTPWD) parameter again to make sure that you have entered it correctly. If the remote user password you enter here is different from the password in the RMTPWD parameter, the command is rejected.

The possible values are:

- retype-remote-user-password—Specifies a string up to 10 characters as the password of the Remote user profile on the backup node.

This parameter is ignored if RMTUSRPRF(*NONE) is specified.

DLTSAVF

Specifies whether the savefiles of the replicated native objects and BSF objects will be deleted on the primary node and the backup node after the objects have been synchronized to the backup.

The possible values are:

- *NO—Specifies that the savefiles are kept on both the primary node and backup node. By default, this value is used.
- *BOTH— Specifies that the savefiles are deleted on both the primary node and backup node.
- *PRIM— Specifies that the savefiles are deleted on the primary node, but kept on the backup node.
- *BACK— Specifies that the savefiles are deleted on the backup node, but kept on the primary node.

CHGOBJAUD

Specifies whether to change the object auditing value to *CHANGE for all native and BSF objects in the replication scope of the group. iCluster requires the changes of all the replicated objects to be audited on the primary node. However, if you are certain that the objects already have object auditing value *CHANGE or *ALL, you can specify CHGOBJAUD(*NO) to speed up the DMMRKSYNC processing.

The possible values are:

- *YES— Changes the object auditing value of all objects in the replication scope to *CHANGE. By default, this value is used.
- *NO—Does not change the object auditing value of the objects in the replication scope.

STRGRP

Specifies whether to start the group with USEMARKED(*YES) after the journal positions have been marked in the iCluster metadata, and the objects have been synchronized to the backup node.

The possible values are:

- *NO—Does not start the group. By default, this value is used.
- *YES— Starts the group with USEMARKED(*YES).

If STRGRP(*NO) is specified, DMMRKSYNC will complete without starting the group. You can start the group by running the DMSTRGRP command with USEMARKED(*YES).

Example

```
DMMRKSYNC GROUP(GRP1) SAVFLIB(SAVFLIB1) STRGRP(*YES)
```

Marks the current positions for journals that are used by group **GRP1** and specifies SAVFLIB1 as the library of the savefiles on the primary node where objects in the replication scope will be saved. Starts the group with USEMARKED(*YES).

Restrictions

You must issue this command on an active node in the cluster when the group is inactive. This command can only be issued on the group's primary node.

Minimum Authority Level

*OPERATOR

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **118**

DMCHKALRD (Check alert destinations command)

Tool that checks the accessibility and availability of alert destinations.

```
DMCHKALRD DESTTYPE ()
```

The DMCHKALRD (Check Alert Destinations) command sends a test alert message to a predefined list of email addresses, SMS addresses, and message queues (alert destinations). Use the DMWRKALR (Work with iCluster Alert Destinations) command to set up the list of alert destinations.

The check job runs only on the node that owns the iCluster metadata. This node, as well as any other node that can potentially take ownership of the metadata, must be set up with SMTP to forward email alerts to destinations.

Input Parameters

DESTTYPE

Specifies which destinations to use for alert messages sent by the **DMCHKALRD** job.

The possible values are:

- *ALL—Sends text messages to all destinations that are defined with the DMADDALR (Add iCluster Alert Destination) command and listed in the **DMWRKALR** panel. Select this value to include email addresses, SMS addresses, and message queues. By default, this value is used.
- *EMAIL—Sends the messages only to addresses with the *EMAIL type in the **DMWRKALR** panel.
- *MSGQ—Sends the messages to only addresses with the *MSGQ type in the **DMWRKALR** panel.
- *SMS—Sends the messages to only addresses with the *SMS type in the **DMWRKALR** panel.

Example

```
DMMRKSNC DESTTYPE (*ALL)
```

Restrictions

You must issue this command on an active node in the cluster. This command runs only on the node that owns the iCluster metadata, even though you can execute the command from any node.

Minimum Authority Level

*OPERATOR

Related information

[DMADDALR \(Add iCluster Alert Destination command\)](#)

[DMWRKALR \(iCluster Alert Destinations command\)](#)

Switchable device commands

Use the switchable device commands to manage switchable devices.

Related information

- [**DMADDSWDEV \(Add Switchable Device Entry to Group command\)**](#)
- [**DMCHGSWDEV \(Change Switchable Device Entry to Group command\)**](#)
- [**DMRMSWDEV \(Remove Switchable Device Entry to Group command\)**](#)

DMADDSWDEV (Add Switchable Device Entry to Group command)

Adds a switchable disk storage device entry to a Switchable Device replication group.

```
DMADDSWDEV GROUP( ) SWDEV( ) ONLINE( )
```

A replication group of type *SWDEV can have a switchable disk storage device associated with it. A switchable disk storage device controls a disk unit that is assigned to the primary node of a group but can be reassigned to the new primary node after a switchover or failover of the group.

Input Parameters

GROUP

Specifies the name of the switchable resource group (type *SWDEV) to which you want to add a switchable disk storage device entry.

SWDEV

Specifies the name of the switchable disk storage device which you want to add to the group. You can associate a maximum of 256 switchable disk storage devices with a single group.

ONLINE

Indicates whether the switchable disk storage device being added to the switchable resource group is varied on or varied off when the group is switched over from one node to another or when it is failed over to another node.

The possible values are:

- *YES—Specifies that the switchable disk storage device is varied on when the group is switched over from one node to another, or when it is failed over to another node. By default, this value is used.
- *NO—Specifies that the switchable disk storage device will not be varied on when the group is switched over from one node to another or when it is failed over to another node.

Example

```
DMADDSWDEV GROUP(ASP33) SWDEV(IASP1) ONLINE(*YES)
```

Indicates that the switchable disk storage device **IASP1** is added to group **ASP33**, and that **IASP1** is varied on when the group's primary node is switched over to the current backup node.

Restrictions

You must issue this command on an active node in the cluster that uses Cluster Resource Services as the failover mechanism.

Minimum Authority Level

*ADMIN

Menu Access

Work with Groups screen - Option 25

DMCHGSWDEV (Change Switchable Device Entry to Group command)

Changes the parameters of a switchable disk storage device entry for a Switchable Device replication group.

```
DMCHGSWDEV GROUP( ) SWDEV( ) ONLINE( )
```

A replication group of type *SWDEV can have a switchable disk storage device associated with it. A switchable disk storage device controls a disk unit that is assigned to the primary node of a group but can be reassigned to the new primary node after a switchover or failover of the group.

Input Parameters

GROUP

Specifies the name of the switchable resource group (type *SWDEV) for which you want to change a switchable disk storage device entry.

SWDEV

Specifies the name of a switchable disk storage device which you want to change for the group.

ONLINE

Indicates whether the switchable disk storage device being changed for the switchable resource group is varied on or varied off when the group is switched over from one node to another or when it is failed over to another node.

The possible values are:

- *YES—Specifies that the switchable disk storage device is varied on when the group is switched over from one node to another or when it is failed over to another node.
- *NO—Specifies that the switchable disk storage device will not be varied on when the group is switched over from one node to another or when it is failed over to another node.

Example

```
DMCHGSWDEV GROUP(ASP33) SWDEV(IASP1) ONLINE(*NO)
```

Indicates that the switchable disk storage device IASP1 for group ASP33 is varied off at switchover.

Restrictions

You must issue this command on an active node in the cluster that uses Cluster Resource Services as the failover mechanism. The nodes in the group must be enabled for switchable resources.

Minimum Authority Level

*ADMIN

Menu Access

Work with Groups screen - Option 26

DMRMVSWDEV (Remove Switchable Device Entry to Group command)

Removes a switchable disk storage device entry from a Switchable Device replication group.

```
DMRMVSWDEV GROUP( ) SWDEV( )
```

Input Parameters

GROUP

Specifies the name of the switchable resource group (type *SWDEV) from which you want to remove a device entry.

SWDEV

Specifies the name of a switchable disk storage device which you want to remove from the group.

Example

```
DMRMVSWDEV GROUP(ASP33) SWDEV(IASP1)
```

Indicates that the switchable disk storage device IASP1 is removed from group ASP33.

Restrictions

You must issue this command on an active node in the cluster that uses Cluster Resource Services as the failover mechanism.

Minimum Authority Level

*ADMIN

Menu Access

Work with Groups screen - Option 27

Other commands

Use these commands to restore the iCluster communications registry, view settings for nodes, and set and specify commands to run at sync points.

Related information

[**SETHAREG \(Restore iCluster Communications Registry command\)**](#)

[**DMSETSYNC \(Set Sync Point command\)**](#)

[**DMSYNCNCCMD \(Run Command at Sync Point command\)**](#)

[**DMCHGUPRF \(Change user profile command\)**](#)

SETHAREG (Restore iCluster Communications Registry command)

Restores the iCluster communications registry that might have been corrupted through node failure, interruption, or other circumstances.

After the SETHAREG (Restore iCluster Communications Registry) command completes, all replication at the node must be ended, the node itself must be ended, and the XDMCLUSTER subsystem must be ended, and then restarted for the restored communications registry to take effect.

SETHAREG

Input Parameters

None.

Example

SETHAREG

Restores the iCluster communications registry on the local node.

Restrictions

You must issue this command on the node where the operation will be performed.

The iCluster product library must be the current library.

Minimum Authority Level

Any user with access to the iCluster product library.

Menu Access

This command is not available from the iCluster menus.

DMSETSYNC (Set Sync Point command)

Places checkpoint entries in journals.

DMSETSYNC GROUP() USREXIT() SCRAPERECEIVE() APPLY() USRDATA()
BACKUP()

Checkpoint entries define synchronization points when journal entries are scraped, received, and applied on the primary and backup nodes. After you issue this command, you cannot delete or change the checkpoint journal entry that you defined. Exercise caution before you issue this command.

After the checkpoint condition is met, a user exit program can be called by one of the replication jobs on the primary

or backup node. For scrape and apply jobs, the checkpoint condition implies that all jobs have reached the checkpoint and are waiting for the user exit program to be completed. Depending on the return code of the user exit program, these jobs will continue replication beyond the checkpoint or complete. For the receive jobs, the checkpoint condition is met when all the jobs except the last one have passed the checkpoint. In this case, it is the last job that triggers the user exit program.

A replication job is any iCluster job that is associated with replication activities. The following replication jobs are related to this command:

- Journal scraper jobs
- Jobs that receive journal entry information from the primary node and put it into the staging store (called receive jobs)
- Apply processes or jobs that remove journal entry information from the staging store and apply it to the backup node

You can use checkpoint journal entries when it is necessary to synchronize operations on primary and backup nodes. After defining a checkpoint journal entry on the primary node, synchronization is achieved when the checkpoint is reached on the backup node. A user exit program can then be called to perform some operation with the user-defined data that is passed to the program. For example, you may want to backup a consistent database or perform summary calculations after all entries have been replicated from the primary node and applied on the backup node.

A replication apply job on a backup node processes journal entries until it reaches the checkpoint journal entry that is generated by this command. At this point, the job waits for other specified replication jobs to reach this checkpoint journal entry. Synchronization is achieved when all active replication jobs reach the checkpoint journal entry. This means that one or more jobs will wait at the checkpoint journal entry until all replication jobs reach the sync point

The following operational considerations apply to using the DMSETSYNC (Set Sync Point) command:

- After you issue the DMSETSYNC (Set Sync Point) command, you cannot delete or change the checkpoint journal entry that you defined.
- Normal mirroring operations are not active when the user exit program is running. Unless a special value is returned by one of the user exit program arguments, mirroring is resumed after the user exit program completes. It is recommended to minimize the execution time of the user exit programs so that mirroring can resume as soon as possible.

Input Parameters

GROUP

The name of the replication group that will synchronize at a checkpoint journal entry. Replication jobs that are associated with the group will synchronize at the checkpoint journal entry. The group must be active.

USREXIT

The name of the user exit program to invoke when all active replication jobs reach the checkpoint journal entry.

The same user exit program can be invoked at different synchronization points (journal entry scrape, receive, and apply). User exit programs that are invoked when journal entries are being scraped for a local journaling group must reside on the primary node. If the user exit program is also invoked at journal receive or apply times, it must also reside in the same location on the backup node. To specify a different user exit program at multiple sync points, issue the DMSETSYNC (Set Sync Point) command (once for each sync point) so that a different user exit program can be specified for each invocation of the command.

Note: User exit programs must be compiled with the **Use adopted authority** option set to *NO. Otherwise, the user exit programs cannot be run and mirroring continues.

The possible values are:

- <name>—Specifies the user exit program name.
- *NONE—Does not call a user exit program.

The library where the user exit program resides must be identified when *NONE is not specified. Prefix the name of the user exit program name with the name of the library where the program is located (for example, LIB1/USREXIT1).

Or choose the following value:

- *PRODLIB—Specifies the iCluster installation library.

If iCluster cannot find the user exit program, an appropriate error message is generated and mirroring continues.

SCRAPE

Indicates whether you want to synchronize group replication jobs when journal entries are scraped on the primary node. This parameter is supported only for local journaling groups.

At this sync point, the user exit program that is specified through the **USREXIT** parameter is invoked when all active scrape jobs reach the checkpoint on the primary node by scraping the journal entry that defines the synchronization point.

You can use a user exit to end mirroring by setting a return value. Setting the value for the scrape affects both the scrape and receive processes. The apply process is not affected.

The possible values are:

- *NO—Does not synchronize group journal scraper jobs at the checkpoint journal entry. The user exit program is not invoked. By default, this value is used.
- *YES—Synchronizes group journal scraper jobs at the checkpoint journal entry and then invokes the user exit program if one is specified.

 **Note:** If the group uses a remote journal, then you must set this value to *NO.

RECEIVE

Indicates whether to call the user exit program after all the receive jobs pass through the sync point.

 **Note:** Unlike the scrape and apply jobs, the receive jobs are not synchronized at sync point journal entries. The receive jobs pass through the sync point at their own pace. The user exit program that is specified through the **USREXIT** parameter is called asynchronously by the receive job that is the last one to receive a sync point journal entry.

Unlike the scrape and apply jobs, you cannot end the receive jobs by using a specific return code from the user exit program. However, the receive jobs will stop automatically if the scrape jobs are stopped by the checkpoint user exit program.

The possible values are:

- *NO—Does not invoke the user exit program after all the receive jobs pass the sync point (received sync point journal entry). By default, this value is used.
- *YES—Invokes the sync point user exit program if one is specified.

APPLY

Indicates whether to synchronize replication jobs when journal entries are applied on the backup node.

At this sync point, the user exit program that is specified through the **USREXIT** parameter is invoked when all active replication jobs for the backup node apply processes synchronize at the checkpoint journal entry.

You can use a user exit to end the apply jobs by setting a return value. Setting this value for the apply process affects only the apply process.

The possible values are:

- *YES—Synchronizes group apply processes at the checkpoint journal entry and then invokes the user exit program if one is specified. By default, this value is used.
- *NO—Does not synchronize group apply processes at the checkpoint journal entry and the user exit program is not invoked.

USRDATA

Identifies the user-defined data to pass to the user exit program that is specified through the **USREXIT** parameter.

You can pass a maximum of 400 bytes of data to the user exit program. If the data contains spaces or non-alphanumeric characters (commas, periods, and so on), enclose the data in single quotes.

 **Note:** This parameter is ignored if no user exit program is specified.

BACKUP

The name or type of the backup node of the group's replication processes for which synchronization points will be set. The values *HABACKUP and *DRBACKUP are only valid if a group of type *HADR is named on the GROUP parameter.

The possible values are:

- *ALL—Specifies that synchronization points will be set for all backup nodes for the group specified. This is the only valid option for groups that are not *HADR groups.

 **Note:** If this option is selected for an *HADR group, two sets of synchronization points will be set for the group: one set of synchronization points will be used for the processes that replicate between the primary node and the HA backup node of the group, and the other set will be used for the processes that replicate between the primary node and the DR backup node.

- *HABACKUP—Specifies that synchronization points will be set for the group but they will apply only to the processes that replicate to the HA backup node of the group. This value is valid only when an *HADR group is specified in the GROUP parameter.
- *DRBACKUP—Specifies that synchronization points will be set for the group but they will apply only to the processes that replicate to the DR backup node of the group. This option is valid only when an *HADR group is specified in the GROUP parameter.
- <name>—Specifies the single backup node for which synchronization points will be set for the specified group's replication processes. This option is valid only when an *HADR group is specified in the GROUP parameter.

Example 1

```
DMSETSYNC GROUP(GRP1) USREXIT(LIB1/CPUPGM) SCRAPER(*NO) APPLY(*YES)  
USRDATA(QTY400STS1PRC800)
```

Replication jobs that are associated with group **GRP1** are synchronized at the checkpoint journal entries.

The checkpoint user exit program **CPUPGM** is located in library **LIB1**. This program must reside on the backup node.

Replication jobs will only be synchronized when journal entries are applied on the backup node. No sync points are established for journal scraping or receiving.

User-defined data consisting of a sequence of characters are passed to the user exit program **CPUPGM**.

Example 2

```
DMSETSYNC GROUP(GRP2) USREXIT(*NONE) SCRAPER(*YES) APPLY(*YES)
```

Replication jobs that are associated with group **GRP2** are synchronized at the checkpoint journal entries.

A user exit program has not been specified in this command.

Replication jobs are synchronized when journal entries are scraped on the primary node as well as applied on the backup node.

Example 3

```
DMSETSYNC GROUP(GRP3) USREXIT(*PRODLIB/CPUPGM) SCRAPER(*YES) APPLY(*YES)  
USRDATA('DJONES 750098 ASMITH 912457')
```

Replication jobs that are associated with group **GRP3** are synchronized at the checkpoint journal entries.

The checkpoint user exit program **CPUPGM** is located in the iCluster installation library. This program must reside in the same library on both the primary and backup nodes.

Replication jobs are synchronized when journal entries are scraped on the primary node and applied on the backup node.

User-defined data consisting of a sequence of characters are passed to the user exit program **CPUPGM**. Note that single quotes are required to enclose data that includes spaces and other non-alphanumeric characters.

Restrictions

You must issue this command on an active node in the cluster when the group is active.

Minimum Authority Level

*OPERATOR

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **72**

Related information

[Path object specifiers](#)

[Passing arguments to sync point user exit programs](#)

DMSYNCCMD (Run Command at Sync Point command)

Runs a specified command at synchronization check points for a group or a set of groups.

```
DMSYNCCMD CMD( ) PRIMNODE( ) GROUP( ) BACKUP( )
SCRAPE( ) APPLY( ) ENDMIR( ) SBMJOB( ) JOBD( )
JOBNAME( ) USER( ) INLLIBL( )
```

This command creates the sync check points and runs the specified command. This command is similar in function to **DMSETSYNC**, but runs a command and does not require a program. This command can be run for more than a single group.

Invoke this command on an active node in the cluster when the specified group or groups are active.

Input Parameters

CMD

Specifies a command string up to 256 bytes. This command string runs when the synchronization point for the specified group or groups is reached. Verify the command in an external environment. The command is not validated when the **DMSYNCCMD** command is run. Validate the command externally or press F4 to prompt the command.

PRIMNODE

Specifies the primary node of the group or groups to create a synchronization check point and submit the command.

The possible values are:

- *CURRENT—Specifies that the primary node for a group or groups is the current node. By default, this value is used.
- <name>—Specifies the node name of the primary node for a group or groups.

GROUP

Specifies the name of the replication group or groups to define synchronization points for.

The possible values are:

- *PRIMNODE—Specifies to define synchronization points for the active replication groups whose primary node is the node specified in the PRIMNODE parameter.
- <name>—Specifies the replication group name. The group must be active.

BACKUP

Specifies the backup node of the group or groups to create a synchronization check point and submit the command for.

The possible values are:

- *FIRST—Specifies that the backup node of the groups for which synchronization points are set is the backup node of the first group in the list of groups. This option is available only when the **GROUP** parameter value is *PRIMNODE.

-
- *BOTH—Specifies that synchronization points will be defined for the group's replication processes to both backup nodes of the group. This value is valid only when a single *HADR group is specified in the **GROUP** parameter.
 - *HABACKUP—Specifies that synchronization points will be defined for the group, but they will apply only to the replication processes for the HA backup node of the group. This value is valid only when a single *HADR group is specified in the **GROUP** parameter.
 - *DRBACKUP—Specifies that synchronization points will be defined for the group, but they will apply only to the replication processes for the DR backup node of the group. This option is valid only when a single *HADR group is specified in the **GROUP** parameter.
 - <name>—Specifies the node name of the backup node for a group or groups. Synchronization points are defined for the active replication groups whose backup node is the specified node.

SCRAPE

Specifies whether to run the command at the synchronization point that is reached by the journal scrape processes on the primary node.

This parameter is ignored when at least one group that satisfies the **PRIMNODE**, **GROUP**, or **BACKUP** parameters is a remote journal group.

The possible values are:

- *NO—Specifies not to run the command at the synchronization point that is reached by the journal scrape processes. By default, this value is used.
- *YES—Specifies to run the command when the journal scrape jobs of the specified group or groups are synchronized at the checkpoint journal entry.

APPLY

Specifies whether to run the command at the synchronization point that is reached by the apply processes on the specified backup node or nodes.

The possible values are:

- *YES—Specifies to run the command at the synchronization point that is reached by the apply process. By default, this value is used.
- *NO—Specifies not to run the command at the synchronization point that is reached by the apply process.

ENDMIR

Specifies whether to end the replication process after the specified command is run.

The possible values are:

- *NO—Specifies not to end the replication process after the specified command is run. By default, this value is used.
- *YES—Specifies to end the replication process after the specified command is run. The replication process is ended for all groups or a single group, depending on how the command is run:
 - If the command is run by a journal scrape process, then all replication processes for the groups that are involved in the synchronization point are ended.
 - If the command is run by an apply process, then only the apply processes of the group that is involved in the synchronization point are ended.

SBMJOB

Specifies whether to run the command as a submitted job or run directly in the replication job that processes

the synchronization point.

The possible values are:

- *NO—Run the command directly in the replication job that processes the synchronization point. By default, this value is used.
- *YES—Submit the command to run as a separate submitted job.

JOBD

Specifies the job description for the submitted job, and the library of the job description. The default value is the **QDFTJOB**D job description in the **QGPL** library.

JOBNAME

Specifies the job name of the submitted job. The default value is **DMSYNCCMD**.

USER

Specifies the user profile that the submitted job runs under.

The possible values are:

- *CURRENT—Specifies that the job that runs the command is submitted under the current user profile. By default, this value is used.
- <username>—Specifies a user profile on the system where the job runs.

INLLIBL

Specifies the library list to use for the submitted job.

The possible values are:

- *SBMJOB—Specifies that the initial library list of the job that runs the command is the same library list that is specified for the submitting job. By default, this value is used.
- *JOB—Specifies that the initial library list of the job that runs the command is the same library list that is specified for the job description.

Example

```
DMSYNCCMD CMD(SAVLIB LIB(FMS*) DEV(TAP01))
BACKUP(SECONDARY) GROUP(*PRIMNODE) PRIMNODE(PRIMARY) APPLY(*YES)
ENDMIR(*YES) SBMJOB(*YES) JOB(QGPL/QDFTJOB) USER(*CURRENT)
INLLIBL(*JOB)
```

Submits a job to save the libraries that start with **FMS** on the system that is defined by the backup node **SECONDARY**.

The groups that are involved in the synchronization point that issues the **SAVLIB** command are the groups that are actively replicating from the primary node **PRIMARY** to the backup node **SECONDARY**.

The **SAVLIB** command is issued at the apply process checkpoint that is set up by the product in the same way that synchronization points are set up by the **DMSETSYNC** command.

All replication processes are ended after the **SAVLIB** command is run on the backup node.

The **SAVLIB** command is submitted to a batch job with the job description **QGPL/QDFTJOB**D and the user profile from which the **DMSYNCCMD** was originally issued.

Restrictions

You must invoke this command on an active node in the cluster when the specified group or groups are active.

Minimum Authority Level

*OPERATOR

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **77**

DMCHGUPRF (Change user profile command)

Changes the status of user profiles within specific scope.

```
DMCHGUPRF USRPRF() GROUP() USRCLS() EXCQPRF() STATUS() RSTPWD() FILE() BCKPRF() BCKFILE()
```

This command allows you to generically disable or enable user profiles. This is useful at switchover time or when user profile statuses need to be altered. Additionally, the password of the user profile can be set to the same value as the user profile if RSTPWD(*YES) is specified.

The following user profiles are not changed by this command:

- QSECOFR
- QSYSOPR
- QSYS
- DMCLUSTER

Input Parameters

User profile (USRPRF)

Specifies a user profile name or a generic name of the user profiles that must be changed.

The possible values are:

- *ALL—Specifies that all user profiles must be included in the processing.
- generic*-user-profile—Specifies the generic name of the user profile that must be enabled or disabled.
- *FILE—Specifies that the input file specified on the **FILE** parameter must be used for the user profiles and the status.
- user-profile—Specifies the name of the user profile that must be enabled or disabled.

This is a required parameter.

Group name (GROUP)

Specifies the group name of user profiles whose status needs to be changed.

The possible values are:

- *CURNODE—All user profiles on the current node need to be changed.

- Name—All user profiles that belong to a specific group need to be changed.

User class (USRCLS)

Specifies the user class that the profile must belong to. Only user profiles belonging to the specified user class will be changed.

The possible values are:

- *ALL—User profiles of any class will be included for processing.
- *USER—User profiles of the User class will be included for processing.
- *SYSOPR—User profiles of the System operator class will be included for processing.
- *PGMR—User profiles of the Programmer class will be included for processing.
- *SECADM—User profiles of the Security administrator class will be included for processing.
- *SECOFR—User profiles of the Security officer class will be included for processing.

Exclude Q* user profiles (EXCQPRF)

Specifies that user profiles starting with the letter Q must be skipped for the enable or disable action.

The possible values are:

- *YES—All user profiles that start with the letter Q will be skipped for processing.
- *NO—User profiles which start with the letter Q will be processed, except for the ones explicitly excluded by the command. See the command-level help for more information.

New user profile status (STATUS)

Specifies the status to which the user profile must be changed. If USRPRF is equal to *FILE, this value will be ignored.

The possible values are:

- *ENABLED—The selected user profiles will be enabled and can sign on and invoke jobs.
- *DISABLED—The selected user profiles will not be able to sign on and invoke jobs.

Reset password (RSTPWD)

If enabled, allows passwords to be reset to the same value as the name of the user profile.

Note: To prevent causing a security issue, passwords will only be reset for profiles that do not start with the letter Q.

The possible values are:

- *YES—The password of the user profile will be set to *USRPRF. In addition, the password will be set to **expired** so that it must be changed the next time the user signs on to the system.
- *NO—The password will not be reset.

Input file for profiles (FILE)

Sets the status of the user profiles according to the status in the specified file. The file must be of the DSPUSRPRF *OUTFILE format.

! **Note:** If RSTPWD(*YES) has been specified, the passwords will still be reset to the user profile name.

Backup profiles before change (BCKPRF)

Specify whether to backup user profile statuses before running the DMCHGUPRF command.

The possible values are:

- *YES—Backup a copy of the current snapshot of user profiles' statuses to the file specified in the **BCKFILE** parameter.
- *NO—Do not backup user profiles' statuses to the file specified in the **BCKFILE** parameter.

Backup file for profiles (BCKFILE)

Specifies the specific file for the current snapshot of user profiles. The file is of the DSPUSRPRF *OUTFILE format.

Example

Change user profile using command parameters:

```
DMCHGUPRF USRPRF(*ALL) GROUP(*CURNODE) USRCLS(*ALL) EXCQPRF(*YES) STATUS(*ENABLED) RSTPWD(*YES)  
BCKPRF(*YES) BCKFILE(CHGUPRFLIB/BKCHGUPRF)
```

Change user profile using input file:

```
DMCHGUPRF USRPRF(*FILE) GROUP(*CURNODE) USRCLS(*ALL) EXCQPRF(*YES) RSTPWD(*YES) FILE(FILELIB/FIL  
ENAME) BCKPRF(*YES) BCKFILE(CHGUPRFLIB/BKCHGUPRF)
```

Restrictions

You must invoke this command on an active node in the cluster.

Minimum Authority Level

*OPERATOR

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **400**

Journal management commands

Use journal management commands to configure and manage journals and journal receivers.

Related information

[DMWRKJRN \(Work with Journals command\)](#)

[CHGHAJRN \(Change Journal Receiver command\)](#)

[DLTHAJRCV \(Delete Journal Receiver command\)](#)

-
- [**ENDHADJRCV \(End Journal Management command\)**](#)
 - [**DMRMVJNMNG \(Remove Journal Management Record command\)**](#)
 - [**DMSTRJNMNG \(Restart Journal Management Jobs command\)**](#)
 - [**STRHARTJRN \(Start Remote Journals command\)**](#)
 - [**ENDHARTJRN \(End Remote Journals command\)**](#)

DMWRKJRN (Work with Journals command)

Displays a list of journals and their attributes.

```
DMWRKJRN JRN( ) FILTER( ) JRNEXIT( )
```

The journal list has options to start a journal management job, end the journal management job, change journal receivers, and work with journal attributes.

Input Parameters

JRN

The name of the journal and the name of the library where the journal resides. This parameter is required.

Identify the library where the journal resides. Prefix the journal with the name of the library where the journal is located. For example:

LIB2/JRN1

The possible values for the journal are:

- <name>—Specifies a journal name.
- <generic*>—Specifies a generic journal name, to identify multiple journals in the specified library or libraries.
- *ALL—Specifies all journals in the specified library or libraries.
- *ALLDTA—Specifies all database journals in the specified library or libraries. Excludes the QSYS/QAUDJRN audit journal.
- *ALLRMT—Specifies all journals with remote journals in the specified library or libraries. The audit journal QSYS/QAUDJRN is excluded.

The possible values for the library are:

- <name>—Specifies a library name.
- *LIBL—Specifies the set of libraries in your library list. The libraries are searched in order for the first occurrence of the specified journal. By default, this value is used.
- *CURLIB—Specifies the current library. If this parameter is not provided, the QGPL library is searched.
- *USRLIBL—Specifies the current library and the libraries in the user portion of the library list.
- *ALLUSR—Specifies all user libraries.
- *ALL—Specifies all libraries, including QSYS.

FILTER

Specifies the filter that includes or excludes journals for journal selection. The filter is applied only to journals that are specified with the **JRN** parameter.

The possible values of this parameter are: (*YES *YES), (*YES *NO), (*NO *YES), and (*NO *NO), where the first *YES or *NO value applies to Check iCluster and the second *YES or *NO value applies to Check user exit.

- Check iCluster
 - *YES—Check journals that are currently used by iCluster for replication. By default, this value is used.
 - *NO—Do not check whether the journals are used by iCluster for replication.
- Check user exit
 - *YES—Run a user exit program to select the journals to be displayed. You must specify a user exit program other than *NONE.
 - *NO—Do not run a user exit program to select the journals to be displayed. By default, this value is used.

JRNEXIT

Specifies the user exit program that determines if the journal should be displayed in the list.

The user exit program must exist when the **DMWRKJRN** command is run.

The possible values are:

- <name>—Specifies the name of the user exit program. Prefix the user exit program name with the name of the library where the user exit program is located (for example, **LIB1/PGM1**).

A sample user exit program named **JRNEXIT1** is provided in the **QACLSRC** file in the **ICLUSTER** library. You can modify this user exit program for your requirements.

- *NONE—Do not use a user exit program.

The user exit program that is invoked by the **DMWRKJRN** (Work with Journals) command must accept the following parameters:

- 10-character journal name. This input value is the name of journal that is specified with the **JRN** parameter.
- 10-character journal library. This input value is the name of the library of a journal in the list of journals that match the **JRN** parameter.
- 10-character buffer that is not used by the **DMWRKJRN** command allows use of the same user exit program for the **DMWRKJRN** command and the **DLTHAJRCV** command.
- 7-character return code. This output value determines one of the following actions:
 - <Blanks>—Display journal in the journal list.
 - *IGNORE—Do not display journal in the journal list.
 - Other values—No action is taken. Values other than blanks and *IGNORE are not valid and are ignored.

Example

```
DMWRKJRN JRN(*ALL/HADJRN) FILTER(*YES *NO) JRNEXIT(*NONE)
```

Displays all journals named HADJRN in all libraries that are used by iCluster for replication. No user exit program is run.

Restrictions

None

Minimum Authority Level

*USER

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **120**

CHGHAJRN (Change Journal Receiver command)

Changes journal receivers for the system audit journal or a database journal on the primary node. You can also use this command to delete processed receivers for a remote journal on the backup node.

```
CHGHAJRN JOURNAL( ) DLTRCV( ) DAYS( )
```

This command can also delete fully processed receivers. A processed receiver is one where all of the journal entries in the receiver are completely applied to the backup node. In addition, you can specify the minimum number of days that fully processed receivers must be detached before they can be deleted.

- If the journal is a remote journal, this command deletes the fully processed journal receivers when invoked on the backup node.
- If you are using this command for journals with remote journals attached, you should issue this command on the source system of the journal. In this situation, the remote journal must be *ACTIVE when you issue this command.

This journal cleanup procedure can be invoked while mirroring is active.

Input Parameters

JOURNAL

The name of the journal on the primary node for which journal receivers are generated.

- ! **Note:** If the journal has a remote journal and the command is invoked on the primary node, you must specify the name and library of the source journal.

You also need to identify the library where the journal resides. Prefix the journal with the name of the library where the journal receiver is located. For example:

LIB1/JRN1

DLTRCV

Indicates whether you want to delete fully processed receivers that are associated with the named journal.

The possible values are:

- *NO—Does not delete fully processed journal receivers associated with the named journal. These will remain until you delete or archive them. By default, this value is used.
- *YES—Deletes fully processed receivers and remote receivers associated with the named journal.

DAYS

Indicates the minimum number of days that fully processed journal receivers must be detached before they can be deleted. You must specify a positive number.

- Note:** The age of the fully processed journal receivers is based on the detach date and time attribute of the journal receiver.

The possible values are:

- <number>—Specifies the number of days.
- *NONE—Specifies that the age of the receivers is not considered when deleting fully processed journals. By default, this value is used.

Example 1

```
CHGHAJRN JOURNAL(LIB1/HADJRN) DLTRCV(*YES) DAYS(4)
```

Generates new receivers for the default journal **HADJRN** located in the library **LIB1** and deletes fully processed receivers that have been detached for more than 4 days.

Example 2

```
CHGHAJRN JOURNAL(LIB1/JRN1) DLTRCV(*NO) DAYS(*NONE)
```

Generates new receivers for the journal **JRN1** located in the library **LIB1**. Fully processed receivers will not be deleted.

Restrictions

Schedule journal management procedures to run at least once a day and if you are dealing with high volumes, you should schedule them to run several times daily. This scheduling assures that journal receivers can be purged off the system if they become too large or grow too quickly.

You must issue this command on the local node. The node does not have to be active in the cluster.

Minimum Authority Level

Any user with access to the iCluster product library

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **121**

DMWRKJRN menu - Option **2**

DLTHAJRCV (Delete Journal Receiver command)

Deletes journal receivers. You can delete journal receivers when they are no longer required by iCluster or another application.

```
DLTHAJRCV JRN( ) CHKITV( ) DLTRCV( ) DLTICRMT( ) DLTRCVAGE( ) DLTUNSRCV( )
POSEXIT( ) ATTNRCV( ) SEQOPT( ) IGNRMTRCV( ) RMTRCVAGE( ) RMTUNSRCV( ) RMTPOSEXIT( ) RESETTHRSD(
) SEQTHRSD( )
```

Input Parameters

JRN

The name of the journal and the name of the library where the journal resides. This parameter is required.

Identify the library where the journal resides. Prefix the journal with the name of the library where the journal is located. For example:

LIB2/JRN1

The possible values for the journal are:

- <name>—Specifies a journal name.
- <generic*>—Specifies a generic journal name to identify multiple journals in the specified library or libraries.
- *ALL—Specifies all journals in the specified library or libraries.
- *ALLDTA—Specifies all database journals in the specified library or libraries. Excludes the QSYS/QAUDJRN audit journal.

The possible values for the library are:

- <name>—Specifies a library name.
- *LIBL—Specifies the set of libraries in your library list. The libraries are searched in order for the first occurrence of the specified journal.
- *PRD—Specifies all libraries with journals that are currently in use for replication. You must specify *ALL or *ALLDTA for the **JRN** parameter. This is the default value and it only works on primary node.
- *CURLIB—Specifies the current library. If this parameter is not provided, the QGPL library is searched.
- *USRLIBL—Specifies the current library and the libraries in the user portion of the library list. If there is no current library, then specifies only the libraries in the user portion of the library.
- *ALLUSR—Specifies all user libraries.
- *ALL—Specifies all libraries, including QSYS.

CHKITV

Specifies the time interval to check and delete journal receivers that are no longer needed.

The possible values are:

- *ONCE—Check once within the current job. No journal management job is submitted. This is required when you specify a generic name for the **JRN** parameter, both the journal name and the journal library. By default, this value is used.
- *WEEKLY—Check once a week. A week is defined as 168 hours.
- <hours>—Use an interval of the specified number of hours between checks. You must specify a positive number between 1 and 277.

DLTRCV

Specifies to delete processed journal receivers.

The possible values are:

- *YES—Delete the journal receivers that have been processed. By default, this value is used.
- *NO—Do not delete processed journal receivers. You can manually delete the journal receivers later. An OMI0413 warning message stating that the journal receiver is eligible for deletion is written to the job log.

DLTICRMT

Specifies to process and delete the journal receivers for a remote journal on the backup node.

The possible values are:

- *YES—Delete the journal receivers for a remote journal on the backup node. Remote journals that are not scraped by iCluster are not processed. If a user exit program is specified, then a user exit program with the same name must exist in the same library on the backup node.
- *NO—Do not delete the journal receivers for a remote journal. You can manually delete the journal receivers later. The OMI0413 warning message **Journal receiver could be deleted** is written to the job log. By default, this value is used.

DLTRCVAGE

Specifies the minimum age of journal receivers to be deleted. The journal receiver age is calculated from the time that the journal receiver is detached to the current time.

The possible values are:

- *NONE—No minimum age is required. By default, this value is used.
- <days>—A number between 1 and 9999 that specifies the number of days that journal receivers must be kept in a detached state before they can be deleted.

DLTUNSRCV

Specifies to delete unsaved journal receivers.

The possible values are:

- *NO—Unsaved journal receivers are not deleted. By default, this value is used.
- *YES—Unsaved journal receivers are deleted.

POSEXIT

Specifies the user exit program that is invoked to determine the receiver that is currently being processed by an application other than iCluster. Processing of receivers is done up to, but not including, the receiver that is returned by the user exit program.

The user exit program must exist when the **DLTHAJRCV** command is run.

The possible values are:

- <name>—Specifies the name of the user exit program. Prefix the user exit program name with the name of the library where the user exit program is located (for example, **LIB1/PGM1**).

A sample user exit program named **JRNEXIT2** is provided in the **QACLSRC** file in the **ICLUSTER** library. You can modify this user exit program for your requirements.

- *NONE—Do not use a user exit program. By default, this value is used.

The user exit program that is invoked by the **DLTHAJRCV** (Delete Journal Receivers) command must accept the following parameters:

- 10-character journal name. This input value is the name of journal that is specified with the **JRN** parameter.
- 10-character journal library. This input value is the name of the library of a journal in the list of journals that match the **JRN** parameter.
- 10-character receiver name. This output value is the name of the journal receiver for the journal that is specified with the **JRN** parameter.
- 10-character receiver library. This output value is the name of the journal receiver library that is specified with the **JRN** parameter.
- 7-character return code. This output value determines one of the following actions:

- <Blanks>—The journal receiver is returned to the user exit program. Journal receivers that are older than this receiver can be deleted.
- *IGNORE—The specified journal is not used by any application other than iCluster. The receivers for this journal are deleted according to the conditions of this command.
- Other values—No action is taken. Values other than blanks and *IGNORE are not valid and are ignored.

ATTNRCV

Specifies to attach a new journal receiver each time the journal receivers for a local journal are checked and deleted after the specified interval.

The possible values are:

- *NO—A new journal receiver is not attached to the journal. By default, this value is used.
- *YES—A new journal receiver is attached to the local journal.

If you specify *YES for this parameter, you can also specify the SEQOPT parameter. The SEQOPT parameter specifies whether the journal sequence number is continued from the currently attached journal receiver, or the journal sequence number is reset to 1 in the newly attached journal receiver. This parameter is only used for local journals. The possible values are:

*CONT—The journal sequence number of the next journal entry created is one greater than the sequence number of the last journal entry in the currently attached journal receiver.

*RESET—The journal sequence number of the first journal entry in the newly attached journal receiver is reset to 1.

- *PRD—A new journal receiver is attached to the local journal only if the journal is currently in use by iCluster.

SEQOPT

Specifies whether the journal sequence number is continued from the currently attached journal receiver or the journal sequence number is reset to 1 in the newly attached journal receiver. This parameter is only used for local journals, and only applies when the ATTNRCV parameter is set to *YES.

The possible values are:

- *CONT—The journal sequence number of the next journal entry created is one greater than the sequence number of the last journal entry in the currently attached journal receiver. By default, this value is used.
- *RESET—The journal sequence number of the first journal entry in the newly attached journal receiver is reset to 1.

IGNRMRTRCV

Specifies whether to delete the journal receivers of a local journal when the journal receivers have not been replicated to the remote journal receivers. This parameter is used only for local journals that have remote journals.

The possible values are:

- *NO—Specifies to delete receivers of the local journals that have remote journals after they are fully replicated to remote receivers. By default, this value is used.
- *YES—Specifies to delete the receivers of the local journals that have remote journals regardless of whether they are fully replicated to remote receivers.

RMTRCVAGE

Specifies the minimum age of a remote journal receiver in days before it is considered for deletion on backup systems.

The age is calculated from the time the remote journal receiver was detached up to the current time.

The parameter applies to receivers of remote journals that are used by iCluster replication groups and whose local journals reside on the current system. The parameter only applies when the DLTICRMT parameter is set to *YES.

The possible values are:

- *DLTRCVAGE—Specifies that the value of the parameter DLTRCVGE for receivers on the current system is also used for remote receivers on backup systems. By default, this value is used.
- *NONE— Specifies that a remote receiver can be deleted on the backup system if other conditions are satisfied, regardless of its age.
- number-of-days—Specifies the number of days a remote receiver has to be kept on the backup system before it can be considered for deletion.

RMTUNSRCV

Delete unsaved remote receivers. Specifies whether unsaved remote journal receivers on backup systems can be deleted.

The parameter applies to receivers of remote journals that are used by iCluster replication groups and whose local journals reside on the current system. The parameter only applies when the DLTICRMT parameter is set to *YES.

The possible values are:

- *DLTUNSRCV—Specifies that the value of the parameter DLTUNSRCV for receivers on the current system is also used for remote receivers on backup systems. By default, this value is used.
- *NO—Specifies that remote journal receivers that have not been saved should always be kept on backup systems.
- *YES— Specifies that unsaved remote journal receivers on backup systems are deleted if all other conditions are satisfied.

RMTPOSEXIT

Specifies the user exit program that is invoked to determine the remote journal receiver currently being used by an application other than iCluster on the backup system. Processing of remote journal receivers on the backup system is done up to, but not including, the remote receiver returned by the user exit program, that is, the receiver currently needed by the application.

The parameter applies to receivers of remote journals that are used by iCluster replication groups and whose local journals reside on the current system. The parameter only applies when the DLTICRMT parameter is set to *YES.

See the parameter **POSEXIT** for the specification of the user exit program.

The possible values are:

- *POSEXIT—Specifies that the value of the parameter POSEXIT for receivers on the current system is also used for remote receivers on backup systems. By default, this value is used.
- *NONE—No exit program is invoked to retrieve the remote journal receiver that is currently being used on the backup system.
- exit-program—Specifies the name and library of an existing exit program that is invoked to retrieve the remote journal receiver currently being used by an application on the backup system.

RESETTHRSD

Specifies whether to attach a new receiver to the journal with the sequence number in the new receiver reset to 1, when the current journal sequence number reaches a threshold as a percentage of the maximum journal sequence number allowed for the journal. The threshold percentage is specified by the parameter SEQTHRSD.

When the "Receiver size options" of the journal contains *MAXOPT1 or *MAXOPT2, the maximum journal sequence number of the journal is 9,999,999,999. When it contains *MAXOPT3, the maximum journal sequence number of the journal is 18,446,744,073,709,551,600. The parameter is ignored if the "Receiver size options" of the journal does not contain any value of *MAXOPT1, *MAXOPT2, or *MAXOPT3.

This parameter is only used for local journals.

The possible values are:

- *NO—Specifies that journal sequence numbers are not checked for reaching a threshold value. By default, this value is used.
- *YES—Specifies that a new receiver is attached to the journal with the sequence number in the new receiver reset to 1, when the current journal sequence number reaches the threshold (percentage of the maximum sequence number) specified by the parameter SEQTHRSD.

SEQTHRSD

Specifies the percentage of the maximum journal sequence number of the journal as the threshold for resetting the journal sequence number to 1. The parameter is ignored if the value of parameter RESETTHRSD is *NO.

This parameter is only used for local journals.

The possible values are:

- <threshold_percentage>—Specifies the percentage of the maximum sequence number of the journal as the threshold for resetting the journal sequence number to 1. The range is 1-99. The default value is 90.

Example

```
DLTHAJRCV JRN(*PRD/*ALL)
```

Runs once and deletes all journal receivers in all libraries that are no longer required by iCluster for replication. Receivers of remote journals are not processed, age is not used to determine eligibility for deletion, unsaved receivers are not deleted. New receivers will not be attached. No user exit program is run.

Restrictions

Journal management ends when the XDMCLUSTER subsystem ends, for example because of an IPL. To prevent journal management from ending, modify the IBM i QSTRUP default startup program to add this command after the XDMCLUSTER subsystem is started. QSTRUP is specified in the IBM system value QSTRUUPGM, and is typically in the QSYS library.

Minimum Authority Level

*USER

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **122**

DMWRKJRN menu - Option **1**

ENDHADJRCV (End Journal Management command)

Ends the active journal management job that regularly checks and deletes the journal receivers for the specified

journal.

```
ENDHADJRCV JRN( ) OPTION( ) DELAY( )
```

Input Parameters

JRN

The name of the journal and the name of the library where the journal resides. This parameter is required.

The possible values for the journal are:

- <name>—Specifies a journal name.
- <generic*>—Specifies a generic journal name to identify multiple journals in the specified library or libraries.
- *ALL—Specifies all journals in the specified library or libraries.
- *ALLDTA—Specifies all database journals in the specified library or libraries. Excludes the QSYS/QAUDJRN audit journal.

The possible values for the library are:

- <name>—Specifies a library name.
- *LIBL—Specifies the set of libraries in your library list. The libraries are searched in order for the first occurrence of the specified journal.
- *PRD—Specifies all libraries with journals that are currently in use for replication. You must specify *ALL or *ALLDTA for the JRN parameter. This is the default value, and this value only works on the primary node.
- *CURLIB—Specifies the current library. If this parameter is not provided, the QGPL library is searched.
- *USRLIBL—Specifies the current library and the libraries in the user portion of the library list. If there is no current library, then specify only the libraries in the user portion of the library.
- *ALLUSR—Specifies all user libraries.
- *ALL—Specifies all libraries, including QSYS.

OPTION

Specifies how the journal management job is stopped.

The possible values are:

- *CNTRLRD—Specifies a controlled stop. A controlled stop is the recommended end option to enable the journal management job to end gracefully by allowing the necessary tasks to complete. By default, this value is used.
- *IMMED—Specifies an immediate stop.

DELAY

Specifies the maximum time (time-out) for the journal management job to end in a controlled manner. When this time-out is exceeded, the journal management job stops immediately. This parameter applies only when the **OPTION** parameter is set to *CNTRLRD.

The value is:

- <time>—Specifies the delay time in seconds. The default value is 300 seconds.

Example

```
ENDHADJRCV JRN(ICLUSTER/HADJRN) OPTION(*CNTRLRD) DELAY(120)
```

Ends the active journal management job for the HADJRN journal in the ICLUSTER library. The journal management job is ended in a controlled manner. If the journal management job does not end after 120 seconds, the job is ended immediately.

Restrictions

None

Minimum Authority Level

*USER

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **123**

DMWRKJRN menu - Option **4**

DMRMVJNMNG (Remove Journal Management Record command)

Deletes the journal management entry that records the DLTHAJRCV (Delete Journal Receivers) command parameters and job information of the journal management job that has been ended.

```
DMRMVJNMNG JRN( )
```

Input Parameters

JRN

The name of the journal and the name of the library where the journal resides. This parameter is required.

Identify the library where the journal resides. Prefix the journal with the name of the library where the journal is located. For example:

LIB2/JRN1

Example

```
DMRMVJNMNG JRN(LIB1/JRN1)
```

Deletes the journal management entry for the journal **JRN1** in library **LIB1**.

Restrictions

None.

Minimum Authority Level

*USER

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **125**

DMWRKJRN menu - Option **6**

DMSTRJNMNG (Restart Journal Management Jobs command)

Restarts the journal management jobs after a system IPL or after ending the iCluster XDMCLUSTER subsystem.

DMSTRJNMNG

The command restarts journal management jobs for all journals that have journal management entries but are not currently managed by an active journal management job. This command will not end and restart active jobs.

Input Parameters

None.

Example

DMSTRJNMNG

Restarts the journal management jobs immediately.

Restrictions

None

Minimum Authority Level

*USER

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **126**

DMWRKJRN menu - PF7 key

STRHARTJRN (Start Remote Journals command)

Starts all remote journals for one or more local journals.

STRHARTJRN JRN()

You can specify either a generic journal name or a non-generic one for the **JRN** parameter.

Input Parameters

JRN

The name of the journal and the name of the library where the local journal resides. This is a required parameter.

The possible values for journal are:

- *ALL—Specifies that all journals in the specified library or libraries must be processed for starting remote journals.
- generic-name—Specifies the generic name of the local journals to be processed for starting remote journals. A generic name is a character string that contains one or more characters followed by an asterisk (*). If a generic name is specified, all journals that have names with the same prefix as the generic name are processed.
- journal-name—Specifies the name of the journal to be processed for starting remote journals.

The possible values for library are:

- *PRD—Specifies that all journals currently used by iCluster for replication are checked. You can only specify *PRD on primary node, and if you specify *PRD as the library, you can only specify *ALL for the journal name.
- *LIBL—Specifies that all libraries in the library list for the current thread are searched.
- *CURLIB—Specifies that the current library for the thread is searched. If no current library is specified for the thread, the QGPL library is searched.
- *USRLIBL—Specifies that the current library and the libraries in the user portion of the library list are searched, if a current library entry exists in the library list for the current thread. If there is no current library entry, only the libraries in the user portion of the library list are searched.
- *ALLUSR—Specifies that all user libraries in the system are searched.
- *ALL—Specifies that all libraries in the system, including QSYS, are searched.
- library-name—Specifies the library name of the journal.

Example

```
STRHARTJRN JRN(*PRD/*ALL)
```

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **128**

DMWRKJRN menu - Option 13

ENDHARTJRN (End Remote Journals command)

Ends all remote journals for one or more local journals that are not used by iCluster replication processes on this node.

```
ENDHARTJRN JRN()
```

You can specify either a generic journal name or a non-generic one for the **JRN** parameter.

Input Parameters

JRN

The name of the journal and the name of the library where the local journal resides. This is a required parameter.

- ! **Note:** If the local journals are being used by iCluster replication processes on this node, their remote journals will not be ended.

The possible values for the journal are:

- *ALL—Specifies that all journals in the specified library or libraries must be processed for ending remote journals.
- generic-name—Specifies the generic name of the local journals to be processed for ending remote journals. A generic name is a character string that contains one or more characters followed by an asterisk (*). If a generic name is specified, all journals that have names with the same prefix as the generic name are processed.
- Specifies the name of the journal to be processed for ending remote journals.

The possible values for the library are:

- *PRD—Specifies that all journals currently used by iCluster for replication are checked. You can only specify *PRD on the primary node, and if you specify *PRD as the library, you can only specify *ALL for the journal name.
- *LIBL—Specifies that all libraries in the library list for the current thread are searched.
- *CURLIB—Specifies that the current library for the thread is searched. If no current library is specified for the thread, the QGPL library is searched.
- *USRLIBL—Specifies that the current library and the libraries in the user portion of the library list are searched, if a current library entry exists in the library list for the current thread. If there is no current library entry, only the libraries in the user portion of the library list are searched.
- *ALLUSR—Specifies that all user libraries in the system are searched.
- *ALL—Specifies that all libraries in the system, including QSYS, are searched.
- library-name—Specifies the library name of the journal.

Example

```
ENDHARTJRN JRN(*PRD/*ALL)
```

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **129**

DMWRKJRN menu - Option 14

Access path maintenance commands

Delaying access path maintenance for access path logical files can improve the apply performance on the backup node, especially when there are a large number of logical files and record-level changes.

An access path list is a file whose records contain the names of logical files (indexes) and their libraries. An access path list is used to determine the logical files whose maintenance is controlled by the access path list maintenance

parameters that are specified with the access path management commands. You can specify that access paths for logical files should be rebuilt immediately, or delayed for a specified list of files.

Related information

- [**DMWRKAPLST \(Work with Access Path Lists command\)**](#)
- [**DMCRTAPLST \(Create Access Path List command\)**](#)
- [**DMDLTAPLST \(Delete Access Path List command\)**](#)
- [**DMDSPAPLST \(Display Access Path List command\)**](#)
- [**DMADDAPL \(Add Access Path to List command\)**](#)
- [**DMRMVAPL \(Remove Access Path from List command\)**](#)
- [**DMADDPFAPL \(Add PF Access Paths to List command\)**](#)
- [**DMRMVPFAPL \(Remove PF Access Paths from List command\)**](#)
- [**DMSTRAPM \(Start Access Path Maintenance command\)**](#)
- [**DMENDAPM \(End Access Path Maintenance command\)**](#)
- [**DMCHGAPM \(Change Access Path Maintenance command\)**](#)
- [**DMDSPAPM \(Display Access Path Maintenance Details command\)**](#)
- [**DMREGAPLST \(Register Access Path List command\)**](#)

DMWRKAPLST (Work with Access Path Lists command)

Use on the backup node to display the access path lists that are registered to iCluster and the options for access path list management.

```
DMWRKAPLST
```

Input Parameters

None.

Example

```
DMWRKAPLST
```

Displays the access path lists that are registered to iCluster and the options for access path list management.

Restrictions

None.

Minimum Authority Level

Any user with access to the iCluster product library.

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **150**

DMCRTAPLST (Create Access Path List command)

Use on the backup node to create an access path list.

```
DMCRTAPLST APLST( ) DESC( )
```

An empty file is created with the specified name and library. Each record of the file has two fields: **INDEXNAME** and **INDEXLIB**. Each field is of type CHAR with a length of 10. The file has a primary key on the **INDEXNAME** and **INDEXLIB** fields. A control data area with the same name and library is also created. The new access path list is registered with iCluster and is listed on the **DMWRKAPLST** panel.

Input Parameters

APLST

Specifies the name and the library of the access path list. This parameter is required. The possible values for the access path list name are:

- DMACCPLST—Specifies to create the **DMACCPLST** access path list. By default, this value is used.
- <name>—Specifies the access path list name.

The possible values for the library are:

- ICLUSTER—Specifies the ICLUSTER library.
- <name>—Specifies the library name where the access path list is created.

DESC

A short description that identifies the access path list. The description of the file of the access path list will also be changed with the description of the access path list.

- <description>—Specifies the description text. You can use up to 50 characters.

Example

```
DMCRTAPLST APLST(APLISTLIB/APLIST1) DESC('MY ACC PATH LIST')
```

Creates the **APLIST1** access path list in the **APLISTLIB** library on the node with the description 'MY ACC PATH LIST'.

Restrictions

None.

Minimum Authority Level

Any user with access to the iCluster product library.

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **151**

DMWRKAPLST menu - PF7 key

DMDLTAPLST (Delete Access Path List command)

Deletes an access path list that was created with the DMCRTAPLST (Create Access Path List) command.

```
DMDLTAPLST APLST( )
```

The physical file that contains the access path list and the control data area with the same name and library is also deleted.

Input Parameters

APLST

Specifies the name and the library of the access path list to delete. This parameter is required. The possible values for the access path list name are:

- DMACCPLST—Specifies to delete the DMACCPLST access path list. By default, this value is used.
- <name>—Specifies the access path list name.

The possible values for the library are:

- ICLUSTER—Specifies the ICLUSTER library.
- <name>—Specifies the library name where the access path list is deleted.

Example

```
DMDLTAPLST APLST(APLISTLIB/APLIST1)
```

Deletes the **APLIST1** access path list in the **APLLISTLIB** library.

Restrictions

None.

Minimum Authority Level

Users with access to the iCluster product library.

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **152**

DMWRKAPLST menu - Option **6**

DMDSPAPLST (Display Access Path List command)

Displays the access paths in an access path list that was created with the DMCRTAPLST (Create Access Path List) command.

```
DMDSPAPLST APLST( )
```

The access path list is populated with access paths by using the DMADDAPL (Add Access Path To List) or the DMADDPFAPL (Add PF Access Paths To List) command.

Each record that is displayed has a length of 20.

- The first 10 characters represent the name of the access path logical file.
- The second 10 characters represent the library of the access path logical file.

Input Parameters

APLST

Specifies the name and the library of the access path list. This parameter is required. The possible values for the access path list name are:

- DMACCPLST—Specifies to create the DMACCPLST access path list. By default, this value is used.
- <name>—Specifies the access path list name.

The possible values for the library are:

- ICLUSTER—Specifies the ICLUSTER library.
- <name>—Specifies the library name where the access path list is created.

Example

```
DMDSPAPLST APLST(APLISTLIB/APLIST1)
```

Displays the access paths in the **APLIST1** access path list in the **APLISTLIB** library.

Restrictions

None.

Minimum Authority Level

Users with access to the iCluster product library.

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **153**

DMWRKAPLST menu - Option **7**

DMADDAPL (Add Access Path to List command)

Adds an access path logical file to an access path list that was created with the DMCRTAPLST (Create Access Path List) command.

```
DMADDAPL APLST( ) ACCPATH( )
```

Input Parameters

APLST

Specifies the name and the library of the access path list. This parameter is required. The possible values for the access path list name are:

- DMACCPLST—Specifies the **DMACCPLST** access path list. By default, this value is used.

- <name>—Specifies the access path list name.

The possible values for the library are:

- ICLUSTER—Specifies the ICLUSTER library.
- <name>—Specifies the library name.

ACCPATH

Specifies the name and the library of the access path. The access path must be a logical file that defines an access path of a physical file (for example a SQL index). This parameter is required. The possible value for the access path is:

- <name>—Specifies the access path name.

The possible value for the library is:

- <name>—Specifies the library name.

Example

```
DMADDAPL APLST(APLISTLIB/APLIST1) ACCPATH(DBLIB1/INDEX1)
```

Adds the INDEX1 access path in the DBLIB1 library to the APLIST1 access path list in the APLISTLIB library.

Restrictions

None.

Minimum Authority Level

Any user with access to the iCluster product library.

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **154**

DMWRKAPLST menu - Option **2**

DMRMVAPL (Remove Access Path from List command)

Removes an access path logical file from an access path list that was created with the DMCRTAPLST (Create Access Path List) command.

```
DMRMVAPL APLST( ) ACCPATH( )
```

Input Parameters

APLST

Specifies the name and the library of the access path list. This parameter is required. The possible values for the access path list name are:

- DMACCPLST—Specifies the DMACCPLST access path list. By default, this value is used.
- <name>—Specifies the access path list name.

The possible values for the library are:

- ICLUSTER—Specifies the ICLUSTER library.
- <name>—Specifies the library name.

ACCPATH

Specifies the name and the library of the access path to remove from the access path list. This parameter is required. The possible value for the access path is:

- <name>—Specifies the access path name.

The possible value for the library is:

- <name>—Specifies the library name.

Example

```
DMRMVAPL APLST(APLISTLIB/APLIST1) ACCPATH(DBLIB1/INDEX1)
```

Removes the INDEX1 access path in the DBLIB1 library from the APLIST1 access path list in the APLISTLIB library.

Restrictions

None.

Minimum Authority Level

Any user with access to the iCluster product library.

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **155**

DMWRKAPLST menu - Option **3**

DMADDPFAPL (Add PF Access Paths to List command)

Adds all the eligible access paths of a physical file to an access path list.

```
DMADDPFAPL APLST( ) PF( )
```

The access path list must have been created with the DMCRTAPLST (Create Access Path List) command.

The eligible logical files that are based on the specified physical file are added to the access path list.

Ineligible logical files are excluded for the following reasons:

- The logical files are not access paths. For example, SQL views.
- The access paths require unique key values.
- The access path is shared implicitly with an access path that requires unique key values.

Input Parameters

APLST

Specifies the name and the library of the access path list. This parameter is required. The possible values for the access path list name are:

- DMACCPLST—Specifies the DMACCPLST access path list. By default, this value is used.
- <name>—Specifies the access path list name.

The possible values for the library are:

- ICLUSTER—Specifies the ICLUSTER library.
- <name>—Specifies the library name.

PF

Specifies the name and the library of the physical file whose access paths you want to add to the access path list. This parameter is required. The possible value for the physical file is:

- <name>—Specifies the physical file name.

The possible value for the library is:

- <name>—Specifies the library name.

Example

```
DMADDPFAPL APLST(APLISTLIB/APLIST1) PF(DBLIB1/TABLE1)
```

Adds the eligible access paths of the TABLE1 physical file in the DBLIB1 library to the APLIST1 access path list in the APLISTLIB library.

Restrictions

None.

Minimum Authority Level

Any user with access to the iCluster product library.

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **156**

DMWRKAPLST menu - Option **8**

DMRMVPFAPL (Remove PF Access Paths from List command)

Removes the access paths of a physical file from an access path list.

```
DMRMVPFAPL APLST( ) PF( )
```

The access path list must have been created with the DMCRTAPLST (Create Access Path List) command.

Input Parameters

APLST

Specifies the name and the library of the access path list. This parameter is required. The possible values for the access path list name are:

- DMACCPLST—Specifies the DMACCPLST access path list. By default, this value is used.
- <name>—Specifies the access path list name.

The possible values for the library are:

- ICLUSTER—Specifies the ICLUSTER library.
- <name>—Specifies the library name.

PF

Specifies the name and the library of the physical file whose access paths you want to remove from the access path list. This parameter is required. The possible value for the physical file is:

- <name>—Specifies the physical file name.

The possible value for the library is:

- <name>—Specifies the library name.

Example

```
DMRMVPFAPL APLST(APLISTLIB/APLIST1) PF(DBLIB1/TABLE1)
```

Removes the access paths of the **TABLE1** physical file in the **DBLIB1** library from the **APLIST1** access path list in the **APLISTLIB** library.

Restrictions

None.

Minimum Authority Level

Any user with access to the iCluster product library.

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **157**

DMWRKAPLST menu - Option **9**

DMSTRAPM (Start Access Path Maintenance command)

Starts access path maintenance for the access paths in an access path list that was populated with the DMADDAPL (Add Access Path To List) command or DMADDPFAPL (Add PF Access Paths To List) command.

```
DMSTRAPM APLST( ) MAINT( ) SBMJOB( ) LOADMEM( ) CYCLITV( )
```

Replicating a physical file with access paths might take a long time when there is a large number of record-level changes on the primary system. Applying changes on the backup node is impacted by the immediate maintenance of the access paths.

To decrease the impact of access path maintenance, you can delay the access path maintenance until a significant number of updates have been done. You can do this by using an access path list and the DMSTRAPM (Start Access Path Maintenance) command to schedule access path maintenance or to cycle between immediate and delayed access path maintenance.

You can add access paths to an access path list by using the DMADDAPL (Add Access Path To List) command or the DMADDPFAPL (Add PF Access Paths To List) command. To start a process that alternates between *IMMED (immediate) and *DLY (delayed) maintenance on the access paths in an access path list, specify MAINT(*CYCLE) when you run the DMSTRAPM (Start Access Path Maintenance) command.

Input Parameters

APLST

Specifies the name and the library of the access path list that contains the access paths on which to start maintenance. This parameter is required. The possible values for the access path list name are:

- DMACCPLST—Specifies to use the DMACCPLST access path list. By default, this value is used.
- <name>—Specifies the access path list name.

The possible values for the library are:

- ICLUSTER—Specifies the ICLUSTER library.
- <name>—Specifies the library name where the access path list is located.

MAINT

Specifies the type of access path maintenance. The access path maintenance applies to all members of each access path logical file in the access path list.

The possible values are:

- *IMMED—Specifies immediate access path maintenance for a logical file member whenever changes are made to its associated data. The access paths are always current and available, whether the logical file is open or closed. The access path maintenance process ends after all access paths in the list are changed to immediate maintenance.
- *DLY—Specifies to delay maintenance until the logical file member is opened for use. Then each access path is updated with the record level changes that occurred since the logical file was last closed. The access path maintenance process ends after all access paths in the list are changed to delayed maintenance.
- *CYCLE—Specifies to cycle between immediate maintenance and delayed maintenance. Define the interval between cycles with the CYCLITV parameter. Requires the DMENDAPM (End Access Path Maintenance) command to end the process. By default, this value is used.

SBMJOB

Specifies to submit a job for access path maintenance for the access paths in the access path list.

The possible values are:

- *YES—Specifies to submit a job for access path maintenance for the access paths in the access path list. By default, this value is used.
- *NO—Specifies to process access path maintenance in the current job. Requires that the DMENDAPM (End Access Path Maintenance) command be issued from another job.

LOADMEM

Specifies to load the access paths in the access path list into main memory before changing them to immediate maintenance.

The possible values are:

- *NO—Specifies not to load the access paths in the access path list into main memory before changing them to immediate maintenance. By default, this value is used.
- *YES—Specifies to load the access paths in the access path list into main memory before changing them to immediate maintenance. Available only for the MAINT(*CYCLE) and MAINT(*IMMED) access path maintenance types.

CYCLITV

Specifies the time interval between cycles for the MAINT(*CYCLE) access path maintenance.

The possible values are:

- <time>—Specifies a number of seconds between 1–9999. The default value is 60 seconds.

Example

```
DMSTRAPM APLST(APLISTLIB/APLIST1) MAINT(*CYCLE) LOADMEM(*YES) CYCLITV(45)
```

Specifies to start access path maintenance for the **APLIST1** access path list in the **APLISTLIB** library.

The access paths in the access path list are loaded into main memory before they are changed to immediate maintenance.

The cycle between immediate maintenance and delayed maintenance is 45 seconds.

Restrictions

None.

Minimum Authority Level

Any user with access to the iCluster product library.

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **158**

DMWRKAPLST menu - Option **1**

DMENDAPM (End Access Path Maintenance command)

Ends access path maintenance for the access paths in an access path list if the maintenance was started with the **DMSTRAPM** (Start Access Path Maintenance) command with the **MAINT(*CYCLE)** parameter.

```
DMENDAPM APLST( )
```

Input Parameters

APLST

Specifies the name and the library of the access path list whose access path maintenance you want to end. This parameter is required. The possible values are:

- **DMACCPLST**—Specifies the access path list **DMACCPLST**. By default, this value is used.
- <name>—Specifies the access path list name.

The possible values for the library are:

- **ICLUSTER**—Specifies the **ICLUSTER** library.
- <name>—Specifies the library name of the access path list.

Example

```
DMENDAPM APLST(APLISTLIB/APLIST1)
```

Specifies to end access path maintenance for the **APLIST1** access path list in the **APLISTLIB** library.

Restrictions

None.

Minimum Authority Level

Any user with access to the iCluster product library.

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **159**

DMWRKAPLST menu - Option **4**

DMCHGAPM (Change Access Path Maintenance command)

Changes the access path maintenance for a logical file.

```
DMCHGAPM ACCPATH( ) MAINT( ) LOADMEM( )
```

The DMCHGAPM—Change Access Path Maintenance command is similar to the IBM i Change Logical File (CHGLF) command with additional capabilities:

- You can specify to load the access path into main memory before changing the access path to immediate maintenance.
- The access path maintenance is changed for all access path logical files that share an access path implicitly with the specified logical file.

Input Parameters

ACCPATH

Specifies the name and the library of the access path to change maintenance for. The specified access path should be a logical file that defines an access path of a physical file (for example, a SQL index). It cannot require unique key values or share an access path implicitly with another access path that requires unique key values, because such access paths can only have immediate maintenance. This parameter is required. The possible value for the access path is:

- <name>—Specifies the access path name.

The possible value for the library is:

- <name>—Specifies the library name.

MAINT

Specifies the type of access path maintenance for the access path logical file. The access path maintenance applies to all members of the logical file. This parameter is required.

The possible values are:

- *IMMED—Specifies immediate access path maintenance for a logical file member whenever changes are made to its associated data. The access path is always current and available, whether the logical file is open or closed. By default, this value is used.
- *DLY—Specifies to delay maintenance until the logical file member is opened for use. Then the access path is updated with the record level changes that occurred since the logical file was last closed.

LOADMEM

Specifies to load the access path into main memory before changing it to immediate maintenance.

The possible values are:

- *NO—Specifies not to load the access path into main memory before changing it to immediate maintenance. By default, this value is used.
- *YES—Specifies to load the access path into main memory before changing it to immediate maintenance. Available only for the **MAINT(*IMMED)** access path maintenance type.

Example

```
DMCHGAPM ACCPATH(DBLIB1/INDEX1) MAINT(*IMMED) LOADMEM(*YES)
```

Specifies to change the access path maintenance for the INDEX1 access path in the DBLIB1 library. The access path maintenance is changed to *IMMED.

The access path in the access path is loaded into main memory before it is changed to immediate maintenance.

Restrictions

None.

Minimum Authority Level

Any user with access to the iCluster product library.

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **160**

DMDSPAPM (Display Access Path Maintenance Details command)

Displays the access path maintenance parameters and information for the maintenance job that was started with the DMSTRAPM (Start Access Path Maintenance) command with MAINT(*CYCLE).

```
DMDSPAPM APLST( ) DESC( ) MANAGED( ) MNTJOB( ) MAINT( ) SBMJOB( ) LOADMEM( ) CYCLITV( )
```

Input Parameters

APLST

Specifies the name and the library of the access path list. This parameter is required. The possible value for the access path list name is:

- <name>—Specifies the access path list name.

The possible value for the library is:

- <name>—Specifies the library name where the access path list resides.

DESC

Specifies a short description that allows you to identify this access path list.

The description can be a maximum of 50 characters long.

The description of the access path list is the same as the text description of the file for the access path list.

MANAGED

Specifies whether the access path list is currently being managed by an active maintenance job that was started by the DMSTRAPM command with MAINT value *CYCLE.

The possible values are:

- *NO—The access path list is not being managed by an active maintenance job.
- *YES—The access path list is currently being managed by an active maintenance job that was started by the DMSTRAPM command with MAINT(*CYCLE).

MNTJOB

Specifies the job name, job user, and job number of the access path maintenance job. The values could be *NONE, if the job is ended and its spooled files are deleted from the system.

MAINT

Specifies the type of maintenance for the access paths in the access path list. It applies to all members of each access path logical file in the access path list. It applies to all members of each access path logical file in the access path list. DMDSPAPM only displays the access path maintenance details of jobs started by the DMSTRAPM command with MAINT(*CYCLE).

The possible values are:

- *CYCLE— iCluster periodically cycles all the access paths between immediate maintenance and delayed maintenance. During each cycle, iCluster will first change the access path maintenance of each access path in the list to immediate, then to delayed. The interval between cycles is specified by the Cycle interval in seconds (CYCLITV) parameter.

The process started with the DMSTRAPM command using MAINT(*CYCLE) must be ended by the DMENDAPM command on the same access path list. Before the process is ended, iCluster changes the maintenance for all the access path logical files in the list to immediate (*IMMED).

SBMJOB

Specifies whether the DMSTRAPM request will be processed by a submitted maintenance job, or processed in the job that issued the DMSTRAPM command.

The possible values are:

- *YES—A job is submitted to maintain the access paths in the access path list.
- *NO—Access paths are maintained in the same job that issues the DMSTRAPM command. Note that the DMENDAPM command for this process can only be issued from another job.

LOADMEM

Specifies whether the access paths in the access path list are loaded into main memory before changing them to immediate maintenance.

The possible values are:

- *NO—The access paths in the access path list are not loaded into main memory before changing them to immediate maintenance.
- *YES—The access paths in the access path list are loaded into main memory before changing them to immediate maintenance. After the access path maintenance is changed, the access paths will be purged from main memory to disk. This option may speed up the processing of DMSTRAPM command with MAINT(*CYCLE).

CYCLITV

Specifies the time interval in seconds between cycles for the MAINT(*CYCLE) option.

- Number of seconds—Specify a positive number between 1 and 9999. This is the interval, in seconds, at which all the access paths in the access path list are cycled between immediate maintenance and delayed maintenance.

Example

```
DMDSPAPM APLST(ICLUSTER/DMACCPLST)
```

Displays the access path maintenance parameters for the access path list **ICLUSTER/DMACCPLST**. The job information is also displayed if the access path is managed by a maintenance job that was started by the DMSTRAPM command with MAINT (*CYCLE).

Restrictions

None.

Minimum Authority Level

Any user with access to the iCluster product library.

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **161**

DMWRKAPLST menu - Option **5**

DMREGAPLST (Register Access Path List command)

Registers an existing access path list that was created with the DMCRTAPLST (Create Access Path List) command in iCluster 7.1.

For details on using this command after an upgrade from iCluster 7.1 see the *Rocket iCluster Installation Guide*. Only registered access path lists are displayed when the DMWRKAPLST (Work with Access Path Lists) command has run.

```
DMREGAPLST APLST( ) DESC( )
```

Input Parameters

APLST

Specifies the name and the library of the access path list. This parameter is required. The possible values for the access path list name are:

- DMACCPLST—Specifies to register the DMACCPLST access path list. By default, this value is used.
- <name>—Specifies the access path list name.

The possible values for the library are:

- ICLUSTER—Specifies the ICLUSTER library.
- <name>—Specifies the library name where the access path list is created.

DESC

Specifies the description text. You can use up to 50 characters.

Example

```
DMREGAPLST APLST(APLISTLIB/APLIST1) DESC('MY ACC PATH LIST')
```

Registers the access path list **APLISTLIB/APLIST1** with the description 'MY ACC PATH LIST'.

Restrictions

None. Use this command only to register access path lists that were created with the DMCRTAPLST (Create Access Path List) command in iCluster 7.1.

Minimum Authority Level

Users with access to the iCluster product library.

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **162**

DMWRKAPLST menu - PF7 key

Tools for replicating job schedule entries

Use the tools for replicating job schedule entries to retrieve, apply, and remove job schedule entries.

Related information

- [**DMRTVJSE \(Retrieve Job Schedule Entries command\)**](#)
- [**DMAPYJSE \(Apply Job Schedule Entries command\)**](#)
- [**DMRMVJSE \(Remove Job Schedule Entries command\)**](#)

DMRTVJSE (Retrieve Job Schedule Entries command)

Retrieves job schedule entries from the system job scheduler QUSRSPS/QDFTJOBSCD into a user space, and optionally send the user space to a backup node.

```
DMRTVJSE JOB( ) USRSPC( ) DESC( ) BACKUP( )
```

Input Parameters

JOB

Specifies the job name of the job schedule entries to retrieve from the following job scheduler:

QUSRSPS/QDFTJOBSCD

The possible values are:

- *ALL—Specifies all job schedule entries. This is the default value.
- <jobname>—Specifies the job name of the job schedule entries.
- <generic_jobname*>—Specifies a generic job name to identify multiple job schedule entries. Specify one or more characters followed by an asterisk (*).

USRSPC

Specifies the name of the user space where the retrieved job schedule entries are required. This is a required parameter.

You must identify the library where the user space resides. Prefix the user space with the name of the library where the user space is located (for example, LIB1/USRSPC1). The user space is created automatically. If the specified user space exists, the existing user space is replaced.

DESC

A short description that identifies the user space that contains the retrieved job schedule entries.

The possible values are:

- *DFT—Specifies for the command to generate the text description with the job name parameter and the current node name. This is the default value.
- <description>—Specifies a short description of the user space. You can use up to 50 characters.

BACKUP

Specifies the name of the backup node to receive the user space. The user space is sent immediately from the primary node to the same library on the backup node.

The possible values are:

- *NONE—Specifies not to send the user space to any node. This is the default value.
- <name>—Specifies the name of the backup node where the user space is sent.

Example

```
DMRTVJSE JOB(NIGHT*) USRSPC(HSYSOBJ/NIGHT) DESC(*DFT)
```

Retrieves all of the job entries in the job scheduler that start with NIGHT and put the entries into the user space HSYSOBJ/NIGHT.

Restrictions

Run this command on the primary node.

Minimum Authority Level

Any user with access to the iCluster product library.

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option 200

DMAPYJSE (Apply Job Schedule Entries command)

After the job schedule entries are retrieved with the DMRTVJSE (Retrieve Job Schedule Entries) command, use this command to apply the retrieved job schedule entries to the current system job scheduler of QUSRSYS/QDFTJOBSCD.

```
DMAPYJSE USRSPC( ) JSEID( ) HOLD( ) SKPEXP( )
```

When the value of the **SCDDATE** parameter is *NONE for the IBM i Add Job Schedule Entry (ADDJOBSCDE) command, the job schedule entries are applied without a specific schedule date. The first submission date is relative to the current date. If the schedule date and time are later than the current date and time, job schedule entries with a specific schedule date are applied with the schedule date. If the schedule date and time are earlier than the current date and time, then the next submission date is used.

Input Parameters

USRSPC

Specifies the name of the user space that contains the retrieved job schedule entries. This is a required parameter.

You must identify the library where the user space resides. Prefix the user space with the name of the library where the user space is located (for example, LIB1/USRSPC1). The user space is created automatically. If the specified user space exists, the existing user space is replaced.

JSEID

Specifies the identifier of the job schedule entries when they are applied to the job scheduler QUSRSPS/ QDFTJOBSCD. The original description text of the job schedule entries is concatenated to this identifier to generate the text description of the entries to apply. This description makes the entries that are added by this command recognizable when you run the IBM i Work with Job Schedule Entries (WRKJOBSCDE) command.

The possible values are:

- *USRSPC—Specifies to use the name of the user space that is specified on the **USRSPC** parameter to identify the job schedule entries. By default, this value is used.
- <jobscheduleID>—Specifies the identifier of the job schedule entries. You can use up to 50 characters.

HOLD

Specifies whether to immediately set the status of the job schedule entries to *HLD after the entries are applied to the current system job scheduler. Initially holding the job schedule entries prevents the jobs from being started in an uncontrolled manner.

The possible values are:

- *YES—Immediately sets the status of the job schedule entries to *HLD. By default, this value is used.
- *NO—Specifies to apply the job schedule entries with the current status. Job schedule entries might still be held, based on the **SKPEXP** parameter.

SKPEXP

Specifies whether to skip expired job schedule entries. An expired job schedule entry is an entry that is set for a one-time execution with FRQ(*ONCE) for the IBM i ADDJOBSCDE (Add Job Schedule Entry) command, and the next submission date and time are earlier than the current date and time.

The possible values are:

- *YES—Specifies to skip the expired job schedule entries. By default, this value is used.
- *NO—Specifies to apply the expired job schedule entries with the current date and time as the scheduled date and time. The expired job schedule entries are immediately placed in *HLD status.

Example

```
DMAPYJSE USRSPC(HSYSOBJ/NIGHT) JSEID(NIGHTJOBS)
```

Applies the job schedule entries that are recorded in the user space **HSYSOBJ/NIGHT** to the current system job scheduler.

The identifier **NIGHTJOBS** is part of the job schedule entry text description.

Restrictions

Run this command on the primary node.

Minimum Authority Level

Any user with access to the iCluster product library.

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **201**

DMRMVJSE (Remove Job Schedule Entries command)

Removes entries from the current system job scheduler **QUSRYS/QDFTJOBSCD**. Only the job schedule entries that match the specified identifier and job name are removed.

```
DMRMVJSE JOB( ) JSEID( )
```

Input Parameters

JOB

Specifies the job name of the job schedule entries to remove from the job scheduler **QUSRYS/QDFTJOBSCD**.

The possible values are:

- *ALL—Specifies to remove all job schedule entries. By default, this value is used.
- <generic_jobname*>—Specifies a generic job name to identify and remove multiple job schedule entries. Use one or more characters, followed by an asterisk (*).
- <jobname>—Specifies the job name of the job schedule entries to remove.

JSEID

Specifies the identifier of the job schedule entries to remove. This is a required parameter.

The possible values are:

- <jobscheduleID>—Specifies the identifier of the job schedule entries.

Example

```
DMRMVJSE JOB(NIGHT*) JSEID(NIGHTJOBS)
```

Removes the job schedule entries in the current system for job scheduler with job names starting with **NIGHT** and has the job schedule identifier **NIGHTJOBS** at the beginning of the text description.

Restrictions

Run this command on the primary node.

Minimum Authority Level

Any user with access to the iCluster product library.

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **202**

Tools for testing files on the backup node

Use the DMCHGTRGBK and DMCHGCSTBK commands to enable database file triggers and referential constraints on a backup node so that you can test physical files on that node.

By default, triggers and referential constraints are disabled on backup nodes. You can stop replication to the node, use the commands to enable the triggers and referential constraints, and then test the files. When testing is complete, you can use the commands to disable the triggers and referential constraints and restart replication.

Related information

[DMCHGTRGBK \(Change Trigger State On Backup command\)](#)

[DMCHGCSTBK \(Change Referential Constraint State on Backup command\)](#)

DMCHGTRGBK (Change Trigger State On Backup command)

Use on a backup node to enable or disable triggers for physical files, and to stop replication and enable triggers to perform tests that require triggers.

```
DMCHGTRGBK GROUP( ) STATE( ) KEEPDSBPR( )
```

After you complete the tests, you can disable the triggers and restart replication. You can also enable or disable triggers for files related to a single group that is replicating to the current node, or files related to all groups that are replicating to the current node.

Input Parameters

GROUP

Specifies the name of the group to have the trigger state changed for the replicated physical files on the backup node.

The possible values are:

- *BACKUP—Change trigger state for physical files that are replicated by all groups with this node as the backup node. By default, this value is used.
- <group-name>—Specifies the group name.

STATE

Specifies the state to which the triggers of the physical files are to be changed.

The possible values are:

- *ENABLED—Specifies that the triggers are to be enabled. By default, this value is used.
- *DISABLED—Specifies that the triggers are to be disabled.

KEEPDSBPR

Used when the command is run with STATE(*ENABLED). It specifies whether to keep the triggers disabled on the backup if they are currently disabled on the primary node of the group that is replicating the file. This parameter is ignored if the STATE parameter is not set to *ENABLED.

The possible values are:

- *NO—All triggers of the physical files replicated are enabled on the backup node, regardless of their current state on the primary node. By default, this value is used.
- *YES—The triggers are not enabled on the backup node if they are currently disabled on the primary node of the group that is replicating the file. Note that if the group has target library redirection and is not eligible for roleswitch, iCluster will enable all triggers of the replicated physical files on the backup node regardless and will ignore the setting of the KEEPDSBPR parameter.

Example

```
DMCHGTRGBK GROUP(*BACKUP) STATE(*ENABLED) KEEPDSBPR(*NO)
```

Changes the trigger state for physical files that are replicated by all groups with this node as the backup node. Enables triggers on the backup node regardless of their current state on the primary node.

Restrictions

Run this command on the backup node.

Minimum Authority Level

*OPERATOR

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **300**

DMCHGCSTBK (Change Referential Constraint State on Backup command)

Use on a backup node to enable or disable referential constraints for physical files, and to stop replication and enable referential constraints to perform tests that require referential constraints.

```
DMCHGCSTBK GROUP( ) STATE( ) KEEPDSBPR( )
```

After you complete the tests, you can disable the referential constraints and restart replication. You can also enable or disable referential constraints for files related to a single group that is replicating to the current node, or files related to all groups that are replicating to the current node.

Input Parameters

GROUP

Specifies the name of the group to have the referential constraint state changed for the replicated physical files on the backup node.

The possible values are:

- *BACKUP—Change referential constraint state for physical files that are replicated by all groups with this node as the backup node. By default, this value is used.
- <group-name>—Specifies the group name.

STATE

Specifies the state to which the referential constraints of the physical files are to be changed.

The possible values are:

- *ENABLED—Specifies that the referential constraints are to be enabled. By default, this value is used.
- *DISABLED—Specifies that the referential constraints are to be disabled.

KEEPDSBPR

Used when the command is run with STATE(*ENABLED). It specifies whether to keep the referential constraints disabled on the backup if they are currently disabled on the primary node of the group that is replicating the file. This parameter is ignored if the STATE parameter is not set to *ENABLED.

The possible values are:

- *NO—All referential constraints of the physical files replicated are enabled on the backup node, regardless of their current state on the primary node. By default, this value is used.
- *YES—The referential constraints are not enabled on the backup node if they are currently disabled on the primary node of the group that is replicating the file. Note that if the group has target library redirection and is not eligible for roleswitch, iCluster will enable all referential constraints of the replicated physical files on the backup node regardless and will ignore the setting of the KEEPDSBPR parameter.

Example

```
DMCHGCSTBK GROUP(*BACKUP) STATE(*ENABLED) KEEPDSBPR(*NO)
```

Changes the referential constraint state for physical files that are replicated by all groups with this node as the backup node. Enables the referential constraints on the backup node regardless of their current state on the primary node.

Restrictions

Run this command on the backup node.

Minimum Authority Level

*OPERATOR

Menu Access

iCluster Commands menu - F22 (Shift + F10) - Option **301**

Rocket® iCluster resources

iCluster resources include product documentation, videos, white papers, technical notes, online training, blog posts, and newsletters.

Rocket Documentation Library

To search for and print documentation for Rocket Software products, access the Rocket Product Documentation Library at docs.rocketsoftware.com.

A mobile version of this library is automatically displayed when you access the documentation library with any mobile device.

iCluster Desktop documentation

iCluster User Guide

Review information about using and configuring iCluster.

iCluster Installation Guide

Review information about installing and upgrading iCluster on-premises, and iCluster on Cloud.

iCluster External Storage Reference Guide

Review information about integrating iCluster with external hardware solutions that are separate from, or external to, the Power System hardware that runs IBM i.

iCluster Resilient Applications Reference Guide

Review information about using iCluster with resilient applications. This product requires IBM i Cluster Resource Services as the failover mechanism.

iCluster Release Notes

Review the latest enhancements and resolved issues for each release.

iCluster Web documentation

iCluster Web User Guide

Review reference and task information related to iCluster Web.

iCluster Web Administrator Guide

Review information on managing users and user roles in this guide. If you experience problems during an iCluster Web session, refer to the troubleshooting tips.

iCluster Web Installation Guide

Review the necessary system requirements, preconditions, and software installation steps for iCluster Web.

iCluster Web Release Notes

Review the latest enhancements and resolved issues for each release.

Online resources

Videos and white papers

Product demonstration videos and white papers are available on the Rocket Software website (www.rocketsoftware.com/products/rocket-icloud/resources).

Demonstration videos include iCluster monitoring, using the iCluster Role Switch Simulator, performing a switchover, running sync checks, retrieving configuration details, managing suspended objects, and managing

journals.

White papers include *Questions To Ask When Purchasing A High Availability and Disaster Recovery Software Solution and Protecting Your Development Environment*.

Technical notes

Troubleshooting technical notes are available on the Rocket iCluster **Downloads** page in the Rocket Community Forum (rocketsoftware.com/support). Registration is required.

Rocket Blog

Tech Tuesday weekly technical tips and techniques.

iCluster port requirements

Use the following tables to refer the TCP/IP ports required for iCluster and related functions.

Table: iCluster port information

Function name	Service name	Service provider	Port number	Explanation
iClusterService	dmcluster	Rocket iCluster	4545	4545 is the default port specified when the dmcluster service was defined to the TCP/IP service table.
iCluster Communications	-	Rocket iCluster	1027-65505. For example:10001-10100, or 17001-17100	The range of ports on source and target as defined by DMADDNODE (using MINPORT and MAXPORT parameters). For details, refer to the "DMADDNODE (Add Node)" section of the <i>Rocket iCluster User Guide</i> .
iCluster Replications and Status Monitor	drda	IBM	446-447	iCluster uses DDM connections to get the metadata from the target node. For DDM a service port on the target is required. By default, 446 is used.
Remote Journaling (IPV4, IPV6)	rmtjournal rmtjournal-ipv6	IBM	3777-3888	The remote journal TCP connections use the TCP local port of 3777 or 3888.
Printing and output	LPD (line printer daemon)	IBM	515	LPD provides the capability to distribute printer output to the system. If this port is not active, run the STRTCPSVR SERVER(*LPD) command to start it.

When you deploy iCluster on Cloud, you must enable the following services.

Table: iCluster on Cloud service provider information

Cloud provider	Service	Explanation
Skytap	ICMP	ICMP service is disabled by default. Enable ICMP for use by iCluster Monitor and IBM Cluster services.
IBM Cloud	DDM	When you deploy an IBM i instance using the IBM Power VS OS image, the DDM request access in the system's network attributes default value is *REJECT. Change the value to *OBJAUT.

If any of the above ports are restricted using a firewall or any other mechanism, iCluster or related functions might fail to operate. For assistance with configuring ports or working with a firewall beyond the above information, contact the firewall provider or obtain a consulting agreement.