Oracle® Solaris Cluster Data Service for MySQL Cluster Guide

Part No: E63546
September 2015

Oracle Solaris Cluster Data Service for MySQL Cluster Guide

Part No: E63546

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info` or visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs` if you are hearing impaired.

Référence: E63546

Accès aux services de support Oracle

Les clients Oracle qui ont souscrit un contrat de support ont accès au support électronique via My Oracle Support. Pour plus d'informations, visitez le site `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info` ou le site `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs` si vous êtes malentendant.

Contents

Tables

Examples

Using This Documentation

- **Overview** – Explains how to install and configure the Oracle Solaris Cluster HA for MySQL Cluster data service
- **Audience** – Technicians, system administrators, and authorized service providers
- **Required knowledge** – Advanced experience troubleshooting and replacing hardware

Product Documentation Library

Documentation and resources for this product and related products are available at `http://www.oracle.com/pls/topic/lookup?ctx=E56676-01`.

Feedback

Provide feedback about this documentation at `http://www.oracle.com/goto/docfeedback`.

♦ ♦ ♦ CHAPTER 1

Installing and Configuring HA for MySQL Cluster

This chapter explains how to install and configure the Oracle Solaris Cluster HA for MySQL Cluster (HA for MySQL Cluster) data service.

This chapter contains the following sections:

HA for MySQL Cluster Overview

HA for MySQL Cluster enables the Oracle Solaris Cluster software to manage MySQL Cluster by providing components to perform the orderly startup, shutdown, and fault monitoring of MySQL Cluster processes. When a MySQL Cluster database cluster is managed by the HA for MySQL Cluster data service, the MySQL Cluster instances become scalable or multiple-master resources across the Oracle Solaris Cluster nodes.

A MySQL Cluster instance on top of Oracle Solaris Cluster software consists of the `ndbd` server, the `ndb` management server, the `ndb` shutdown controller, and MySQL Cluster Servers. You can configure HA for MySQL Cluster components as a scalable or multiple-master service. You cannot configure HA for MySQL Cluster components as a failover service. The only exception is the MySQL Cluster Server component.

For conceptual information about failover data services, multiple-masters data services, and scalable data services, see *Oracle Solaris Cluster 4.3 Concepts Guide* .

Overview of Installing and Configuring HA for MySQL Cluster

The following table lists the tasks for installing and configuring HA for MySQL Cluster and provides cross-references to detailed instructions for performing these tasks. Perform these tasks in the order that they are listed in the table.

TABLE 1 Task Map: Installing and Configuring HA for MySQL Cluster

Task	For Instructions, Go To
1. Plan the installation.	"Planning the HA for MySQL Cluster Installation and Configuration" on page 16
2. Install and configure MySQL Cluster software.	"Installing and Configuring MySQL Cluster" on page 20
3. Verify installation and configuration.	"Verifying the Installation and Configuration of MySQL Cluster" on page 24
4. Install HA for MySQL Cluster packages.	"Installing the HA for MySQL Cluster Package" on page 25
5. Register and configure HA for MySQL Cluster.	"Registering and Configuring HA for MySQL Cluster" on page 27
6. Verify HA for MySQL Cluster installation and configuration.	"Verifying the HA for MySQL Cluster Installation and Configuration" on page 36
7. Tune the HA for MySQL Cluster fault monitor.	"Tuning the HA for MySQL Cluster Fault Monitor" on page 38
8. Debug HA for MySQL Cluster.	"Debugging HA for MySQL Cluster" on page 42

Planning the HA for MySQL Cluster Installation and Configuration

This section contains the following information that you need to plan your HA for MySQL Cluster installation and configuration.

- "MySQL Cluster and Oracle Solaris Containers" on page 17
- "Oracle Solaris Cluster HA for MySQL Cluster Components" on page 17
- "Configuration Restrictions" on page 17
- "Configuration Requirements" on page 18

- "Dependencies Between HA for MySQL Cluster Components" on page 18
- "Configuration Guidelines" on page 19

MySQL Cluster and Oracle Solaris Containers

Oracle Solaris Cluster HA for MySQL Cluster is supported in the following forms.

- Global zones.
- Zone clusters – A zone cluster is almost a complete virtual cluster. It offers complete isolation between different zone clusters, so a user in zone cluster 1 cannot see anything in zone cluster 2. However, the administrator of the global cluster has access to both zone clusters.

Oracle Solaris Cluster HA for MySQL Cluster Components

HA for MySQL Cluster is a combination of the following components.

TABLE 2 HA for MySQL Cluster Components

Component Name	Description
`ndb` management server	MySQL Cluster requires a daemon called the `ndb` management server to start, stop, and configure a MySQL Cluster cluster. The presence of the management server is required for probing the `ndbd` daemon as well.
`ndb` daemon	The `ndb` daemon implements the MySQL Cluster storage engine called `ndbengine`.
`ndbd` shutdown controller	The `ndbd` shutdown controller brings the MySQL Cluster to a state that enables the `ndbd` daemons to be shut down in any order.
MySQL Cluster Server	A normal MySQL Cluster server which provides the SQL interface for the MySQL Cluster Cluster tables.

Configuration Restrictions

This section describes configuration restrictions that apply only to HA for MySQL Cluster.

Caution - Your data service configuration might not be supported if you do not observe these restrictions.

- **Location for the data directories** – Each instance of the management server or the `ndb` daemon must have its own data directory. The `ndb` daemon instances of one MySQL Cluster Cluster located on the same node can share the same data directory with the management server. The data directory cannot be a global file system shared by all management server or `ndb` daemon instances of the MySQL Cluster across the nodes.
- **Communication between the `ndbd` daemons** – The MySQL Cluster must be configured so that the `ndbd` daemons communicate over the `clprivnet` interfaces of Oracle Solaris Cluster software. Provide IP aliases for the `clprivnet` addresses in the `/etc/inet/hosts` file and configure the `ndb` nodes with these aliases in the MySQL Clusters configuration file `config.ini`. In a non-global zone configuration, you must create the `clprivnet` addresses for the non-global zones.
- **MySQL Cluster arbitration** – MySQL Cluster arbitration must be disabled when MySQL Cluster is configured on Oracle Solaris Cluster nodes. Set the following parameters in the MySQL Cluster `config.ini` file:

  ```
  Arbitration=WaitExternal
  ArbitrationTimeout=2-times-heartbeat-timeout
  ```

  The `heartbeat-timeout` parameter will be displayed when executing the following command:

  ```
  # cluster show
  ```
- **MySQL Cluster version** – The minimum MySQL Cluster version is 7.0.7. Older versions do not support the disabling of MySQL Cluster arbitration.

Configuration Requirements

- **Resource group topology** – If you create more than one `ndb` daemon resource for the same cluster, you must place all `ndb` daemon resources in the same resource group, and the `ndb` shutdown controller must depend on all of them.
- **Non-global zones** – In the underlying non-global zones of zone clusters, you must provide addresses on the private interconnect. Your address range for the private interconnect must have ample spare addresses.

Dependencies Between HA for MySQL Cluster Components

The dependencies between the HA for MySQL Cluster components are described in the following table.

TABLE 3 Dependencies Between HA for MySQL Cluster Cluster Components

Component	Dependency
MySQL Cluster management server resource in a global cluster or zone cluster	SUNW.SharedAddress is required only if the MySQL Cluster Cluster management server should be load balanced in a scalable configuration.
MySQL Cluster ndbd daemon resource in a global cluster or zone cluster	MySQL Cluster management server resource is required.
MySQL Cluster shutdown controller resource in a global cluster or zone cluster	MySQL Cluster ndbd daemon resource is required.
MySQL Cluster server resource in a global cluster or zone cluster	1. MySQL Cluster shutdown controller resource is required. 2. SUNW.SharedAddress is required only if the MySQL Cluster server should be load balanced in a scalable configuration. For any other possible dependency in a MySQL Cluster Server resource, such as SUNW.HAStoragePlus, a failover container resource, or SUNW.LogicalHostname, see, the MySQL Cluster documentation for more details.

You set these dependencies when you register and configure HA for MySQL Cluster. For more information, see "Registering and Configuring HA for MySQL Cluster" on page 27.

If more elaborate dependencies are required, see the r_properties(5) and rg_properties(5) man pages for further dependencies and affinities settings.

Configuration Guidelines

- **Communication path for all MySQL Cluster resources** – Use the IP aliases for the clprivnet addresses as host names for the ndb management server and the MySQL Cluster server together with the ndbd daemon. This practice ensures that complete communication between the MySQL Cluster processes is restricted to the private interconnect.
- **Resource group topology** – Create separate resource groups for the management server resource, the ndb daemon including the ndbd shutdown controller, and the MySQL Cluster server. This setup greatly decouples administrative restart actions of the management server, the ndb daemons, and the MySQL Cluster server. You can take the ndbd resource group offline if you want to shut down your ndb storage engine.
- **Shutdown and restart procedures** – The ndb daemons are grouped in node groups whose members replicate data among each other. All the configured node groups must have at least one member. The data of a MySQL Cluster cluster with an empty node group is incomplete and can become inconsistent. To avoid such data inconsistency, all the data nodes (ndb daemons) panic if a node group becomes empty. To prevent this behavior, restart the data

nodes without loading data by using the shutdown controller's stop algorithm. After this restart, you can perform an unordered shutdown of the `ndb` daemons. Note the following statements:

- You cannot perform a normal shutdown of the `ndb` daemons one by one. Therefore, restart the `ndb` daemons without loading data before you perform a shutdown one by one.
- Upon a stop of the shutdown controller, the data of the MySQL Cluster Cluster is unavailable unless the stop action of the shutdown controller is suspended.
- If the shutdown controller and the `ndb` daemons are in one resource group, the easiest way to shutdown is to take this resource group offline. Disabling all the data nodes on their own without disabling of the shutdown controller leads to an abnormal shutdown of half of the nodes.
- A rolling restart of the data nodes is possible by either disabling and re-enabling the data nodes one by one, or just shutting down a data node with MySQL Cluster methods. In this case, Oracle Solaris Cluster software detects the absence of this process tree and restarts it. You then have to tolerate the error messages of the vanished process tree.

Installing and Configuring MySQL Cluster

This section explains the special requirements for installing MySQL Cluster Cluster for use with HA for MySQL Cluster. For complete information about installing and configuring MySQL Cluster, see `http://www.mysql.com`.

The MySQL version delivered with the Oracle Solaris OS is a pure MySQL server version. It does not include MySQL Cluster. You must obtain MySQL Cluster software from `http://www.mysql.com`.

Determine whether you have to configure HA for MySQL Cluster to run in a global zone or a zone cluster configuration.

To install and configure MySQL Cluster in a global zone configuration or in a zone-cluster configuration, complete the following tasks:

- “How to Enable a MySQL Cluster Database” on page 21
- “How to Install and Configure MySQL Cluster” on page 22

▼ How to Enable a MySQL Cluster Database

For a complete example of deploying in a global zone, see Appendix B, "Deployment Example: Installing HA for MySQL Cluster in a Global Cluster".

1. **(Optional) Adjust the `heartbeat_quantum` and the `heartbeat_timeout` parameter if appropriate.**

   ```
   # cluster set -p heartbeat_quantum=50%-of-heartbeat-timeout \
   -p heartbeat-timeout=value-in-milliseconds
   ```

2. **(Optional) If you want to configure MySQL Cluster in a zone cluster, log in to the target zone.**

   ```
   # zlogin zone-name
   ```

3. **As the `root` role, register the `SUNW.gds` resource types.**

   ```
   # clresourcetype register SUNW.gds
   ```

4. **(Optional) Create a failover resource group to contain the shared-address resource.**

   ```
   # clresourcegroup create shared-address-resource-group
   ```

5. **(Optional) Create the shared-address resource.**

   ```
   # clressharedaddress create -g shared-address-resource-group shared-address-resource
   ```

6. **(Optional) Enable the shared-address resource group.**

   ```
   # clresourcegroup online -eM shared-address-resource-group
   ```

7. **Create the management server resource group.**

   ```
   # clresourcegroup create -p Maximum_primaries=2 -p Desired_primaries=2 \
   mgmd-resource-group
   ```

8. **Create the `ndbd` daemon resource group.**

   ```
   # clresourcegroup create -p Maximum_primaries=2 -p Desired_primaries=2 \
   ndbd-resource-group
   ```

9. **Create the MySQL Cluster server resource group.**

   ```
   # clresourcegroup create -p Maximum_primaries=2 -p Desired_primaries=2 \
   mysql-resource-group
   ```

10. **(Optional) Set the affinities for the MySQL Cluster server resource group.**

```
# clresourcegroup set -p RG_affinities=++ndbd-resource-group mysql-resource-group
```

Next Steps Go to "How to Install and Configure MySQL Cluster" on page 22.

▼ How to Install and Configure MySQL Cluster

Before You Begin Ensure that the MySQL Cluster software is downloaded. If not, go to "Installing and Configuring MySQL Cluster" on page 20 to install it on your cluster. Then return here to continue.

Note - For complete information about installing MySQL Cluster, go to `http://www.mysql.com/`.

For a complete example of deployment in a global zone, see Appendix B, "Deployment Example: Installing HA for MySQL Cluster in a Global Cluster".

The sample commands in this task assume the following values:

- The user and the group that owns the MySQL Cluster are named `mysql`.
- The MySQL Cluster data is installed in the root (`/`) file system. This system can be any file system, including ZFS.

1. **Install the MySQL Cluster binaries on all nodes.**

```
# cd dir-path
# /usr/sfw/bin/gtar xzvf downloaded-MySQL-Cluster-version
# ln -s ./created-dir ./mysql
```

2. **Create the MySQL Cluster user and group on all nodes.**

```
# groupadd -g 1000 mysql
# useradd -u 1000 -g 1000 -d /local/mysql -s /bin/sh mysql
```

3. **Change the owner of the MySQL Cluster on all nodes.**

```
# chown -RL mysql:mysql ./mysql
```

4. **Create the data directory for the management server on all nodes to host the management server.**

```
# mkdir MySQL-Cluster-management-server-data-directory
```

5. **Create the MySQL Cluster management server `config.ini` file according to your requirements.**

   The HA for MySQL Cluster data service provides a sample `config.ini` file for the MySQL Cluster management server.

   The contents of `/opt/SUNWscmys/ndb_mgmd/etc/config.ini_sample` provides a sample MySQL Cluster configuration file that you can use to create your MySQL Cluster instance MySQL Cluster Management Server `Datadirectory/config.ini`. You must still edit that file to reflect your configuration values.

   ```
   # cp /opt/SUNWscmys/ndb_mgmd/etc/config.ini_sample \
   MySQL-Cluster-management-server-data-directory/config.ini
   ```

6. **Create the MySQL Cluster `ndb` daemon `my.cnf` file.**

   The HA for MySQL Cluster data service provides a sample `my.cnf` file for the MySQL Cluster `ndb` daemon.

   The content of `/opt/SUNWscmys/ndbd/etc/my.cnf_sample` provides a sample MySQL Cluster configuration file for the `ndb` daemons, that you can use to create your MySQL Cluster instance MySQL Cluster `ndb` Daemon `Datadirectory/my.cnf`. You must still edit that file to reflect your configuration values.

   ```
   # cp /opt/SUNWscmys/ndbd/etc/my.cnf_sample \
   MySQL-Cluster-ndb-daemon-data-directory/my.cnf
   ```

7. **Install a MySQL Cluster server on all nodes that you want to host a MySQL Cluster server.**

   Refer to *Oracle Solaris Cluster Data Service for MySQL Guide* . As an alternative, you can refer to Appendix B, "Deployment Example: Installing HA for MySQL Cluster in a Global Cluster"

   Note - Make sure to stop the installation process after bootstrapping the MySQL Cluster database and changing the file permissions. If you want to use a sample configuration file for your MySQL Cluster server configuration, use `/opt/SUNWscmys/etc/my.cnf_sample_mysqld_cluster`. You must still edit that file to reflect your configuration values.

Verifying the Installation and Configuration of MySQL Cluster

Before you install the HA for MySQL Cluster packages, verify that each MySQL Cluster instance that you created is correctly configured to run in a cluster. The instance consists of the MySQL Cluster processes together with the associated MySQL Cluster servers.

▼ How to Verify the Installation and Configuration of MySQL Cluster

Perform this procedure for each MySQL Cluster instance that you created in "Installing and Configuring MySQL Cluster" on page 20.

This procedure does not verify that your application is highly available because you have not yet installed your data service.

1. **(Optional) For a zone cluster, log in to your target zone.**

```
# zlogin mysql-zone
```

2. **Start the management server on all nodes that host the MySQL Cluster management server.**

```
# cd MySQL-Cluster-management-server-data-directory
# MySQL-Cluster-installation-directory/bin/ndb_mgmd --configdir=
 \
MySQL-Cluster-management-server-data-directory \
-f MySQL-Cluster-management-server-data-directory/config.ini \
--ndb_nodeid=actual-nodeid
```

Note - The actual node ID is mentioned in the `config.ini` file.

3. **Check the status of the management servers on all hosts running the MySQL Cluster management server.**

```
# MySQL-Cluster-installation-directory/bin/ndb_mgm -e show
```

4. **Start the `ndbd` daemon on all nodes to host the MySQL Cluster `ndb` daemon.**

```
# cd MySQL-Cluster-ndb-daemon-data-directory
# MySQL-Cluster-installation-directory/bin/ndbd \
--defaults-file=MySQL-Cluster-ndb-daemon-data-directory/my.cnf
--ndb-connectstring=node1:1186 -ndb-nodeid=actual-nodeid
```

Note - The actual node ID is mentioned in the `config.ini` file.

---

5. **Check the status of the `ndb` daemons on one host running the MySQL Cluster cluster management server.**

   ```
   # MySQL-Cluster-installation-directory/bin/ndb_mgm -e show
   ```

6. **Start the MySQL Cluster server on all nodes to host the SQL server.**

   Perform to the verification procedure in *Oracle Solaris Cluster Data Service for MySQL Guide* , except do not kill the MySQL Cluster server at the end of the procedure.

7. **Check the status of the management servers on one host running the MySQL Cluster management server.**

   ```
   # MySQL-Cluster-installation-directory/bin/ndb_mgm -e show
   ```

8. **Stop the MySQL Cluster server on all nodes that host the MySQL Cluster server.**

   ```
   # kill -TERM `cat MySQL-Cluster-database-directory/mysqld.pid`
   ```

9. **On one node, stop the MySQL Cluster components.**

   ```
   # MySQL-Cluster-installation-directory/bin/ndb_mgm -e shutdown
   ```

10. **Check the status of the MySQL Cluster components on all hosts which ran the management server.**

    Ensure that all components are shut down.

    ```
    # ySQL-Cluster-installation-directory/bin/ndb_mgm -e show
    ```

Installing the HA for MySQL Cluster Package

If you did not install the HA for MySQL Cluster package during your initial Oracle Solaris Cluster installation, perform this procedure to install the package.

▼ How to Install the HA for MySQL Cluster Package

Perform this procedure on each cluster node where you want the HA for MySQL Cluster software to run.

1. **On the cluster node where you are installing the data service package, assume the `root` role.**

2. **Ensure that the data service package is available from the configured publisher and that the `solaris` and `ha-cluster` publishers are valid.**

   ```
   # pkg list -a ha-cluster/data-service/mysql
   # pkg publisher
   PUBLISHER                           TYPE     STATUS   P  LOCATION
   solaris                             origin   online   F  solaris-repository
   ha-cluster                          origin   online   F  ha-cluster-repository
   ```

   For information about setting the `solaris` publisher, see "Adding, Modifying, or Removing Package Publishers" in *Adding and Updating Software in Oracle Solaris 11.2*.

   Tip - Use the `-nv` options whenever you install or update to see what changes will be made, such as which versions of which packages will be installed or updated and whether a new BE will be created.

   If you do not get any error messages when you use the `-nv` options, run the command again without the `-n` option to actually perform the installation or update. If you do get error messages, run the command again with more `-v` options (for example, `-nvv`) or more of the package FMRI pattern to get more information to help you diagnose and fix the problem. For troubleshooting information, see Appendix A, "Troubleshooting Package Installation and Update," in *Adding and Updating Software in Oracle Solaris 11.2*.

3. **Install the HA for MySQL Cluster software package.**

   ```
   # pkg install ha-cluster/data-service/mysql
   ```

4. **Verify that the package installed successfully.**

   ```
   $ pkg info ha-cluster/data-service/mysql
   ```

   Installation is successful if output shows that `State` is `Installed`.

5. **Perform any necessary updates to the Oracle Solaris Cluster software.**

   For instructions on updating your software, see Chapter 11, "Updating Your Software," in *Oracle Solaris Cluster 4.3 System Administration Guide* .

Registering and Configuring HA for MySQL Cluster

Before you perform the procedures in this section, ensure that the HA for MySQL Cluster data service packages are installed.

Use the configuration and registration files in the `/opt/SUNWscmys/*/util` directories to register the HA for MySQL Cluster resources. The configuration files define the dependencies that are required between the HA for MySQL Cluster component and other resources. For information about these dependencies, see "Dependencies Between HA for MySQL Cluster Components" on page 18.

This section covers the following main topics:

- "Specifying Configuration Parameters for the MySQL Cluster Management Server Resource" on page 27
- "How to Create and Enable Resources for MySQL Cluster Management Server" on page 30
- "Specifying Configuration Parameters for the MySQL Cluster `ndb` Daemon Resource" on page 30
- "How to Create and Enable Resources for the MySQL Cluster `ndb` Daemon" on page 33
- "Specifying Configuration Parameters for the MySQL Cluster `ndb` Shutdown Controller Resource" on page 33
- "How to Create and Enable Resources for MySQL Cluster `ndb` Shutdown Controller" on page 34
- "How to Suspend the `ndb` Shutdown Controller's Stop Action" on page 35
- "How to Create and Enable Resources for the MySQL Cluster Server" on page 35

Specifying Configuration Parameters for the MySQL Cluster Management Server Resource

HA for MySQL Cluster provides a script that automates the process of configuring the MySQL Cluster management server resource. This script obtains configuration parameters from the `mysql_ndb_mgmd_config` file. A template for this file is in the `/opt/SUNWscmys/ndb_mgmd/util` directory. To specify configuration parameters for the MySQL Cluster management server resource, copy the `mysql_ndb_mgmd_config` file to another directory and edit this `mysql_ndb_mgmd_config` file.

Note - This configuration file needs to be accessible on each global-cluster node or zone-cluster node where MySQL Cluster is installed.

Each configuration parameter in the mysql_ndb_mgmd_config file is defined as a keyword-value pair. The mysql_ndb_mgmd_config file already contains the required keywords and equals signs. For more information, see "mysql_ndb_mgmd_config File" on page 45. When you edit the /myplace/mysql_ndb_mgmd_config file, add the required value to each keyword listed in the following table.

TABLE 4 Keyword-Value Pairs in the mysql_ndb_mgmd_config File

Parameter	Description	Example
RS=*mgmd-resource*	Specifies the name that you are assigning to the MySQL Cluster management server resource. You must specify a value for this keyword.	The name of the MySQL Cluster management server resource is mgm-rs.
RG=*mgmd-resource-group*	Specifies the name of the resource group where the MySQL Cluster management server resource will reside. You must specify a value for this keyword.	The name of the MySQL Cluster management server resource group is mgm-rg.
PORT=*1186 LH=shared-address-resource*	Specifies the value of a dummy port. This variable is used only at registration time. If you will not specify a LH variable, omit this value.	The value of the port for the MySQL Cluster management server resource is 1186.
LH=*shared-address-resource*	Specifies the name of the SUNW.SharedAddress resource for the MySQL Cluster management server resource. This name must be the SUNW.SharedAddress resource name you assigned when you created the resource in "How to Enable a MySQL Cluster Database" on page 21. If you did not register a SUNW.SharedAddress resource, omit this value.	The name of the SUNW.SharedAddress resource for the MySQL Cluster management server resourcemysqlclu-sa.
SCALABLE=*scalable-trigger*	Specifies a scalable registration for the MySQL Cluster management server resource. Any value here will trigger a scalable registration. If you did not register a SUNW.SharedAddress resource, omit this value.	It is a scalable resource, so the value for the scalable trigger SCALABLE is y.
LB_POLICY=*loadbalancing-policy*	Specifies the load-balancing policy for a scalable resource. Leaving this parameter empty for a scalable resource results in the default load-balancing policy. If you did not register a SUNW.SharedAddress resource, or if you want to use the default load-balancing configuration, omit this value.	The default load-balancing policy is used, so the LB_POLICY variable is empty.
HAS_RS=*dependency-list*	Specifies a comma-separated list of resources that the MySQL Cluster management server depends on.	The MySQL Cluster management server resource does not depend on any other

		resource. So the `HAS_RS` variable is `empty`.
`RS_PROP=`*resource-property-string*	Specifies a string that is formatted as: `-p` *property-value*=*value*.	Additional standard and extension properties.
`PARFILE=`*parameter-file*	Specifies the name of the parameter file where the MySQL Cluster management server specific parameters of the MySQL Cluster management server resource are stored. This file will be created during the registration process. You must specify a value for this keyword.	The parameter file is `/pfile/mgmd-pfile`, so the `PARFILE` variable is set to `/pfile/mgmd-pfile`.
`BASEDIR=`*MySQL-Cluster-base-directory*	Specifies the directory where MySQL Cluster is installed. A valid `BASEDIR` variable specifies a directory which contains `ndb_mgmd` under `bin` or `libexec`.	MySQL Cluster is installed in `/usr/local/mysql`, so the value of `BASEDIR` is `/usr/local/mysql`.
`USER=`*mgmd-user*	Specifies the user under which the MySQL Cluster management server is started. If you do not specify any value, the MySQL Cluster management server is started as root.	The MySQL Cluster management server should be started under the `root` user, so the `USER` variable is `mgmd-user`.
`TRY_RECONNECT=`*1*	Specifies how often a connection to the MySQL Cluster management server should be retried before the attempt is abandoned. You must specify a value for this parameter.	If the first connection failure to a MySQL Cluster management server should lead to an abort, so the value of `TRY_RECONNECT` is `1`.
`CONNECT_STRING=`*mgm-connect-string*	Specifies a valid connect string for the management servers in the format `nodename_1[:port]`, `nodename_2[:port]`. The local management server must be the first `nodename` in the list.	The MySQL Cluster management servers are running on `priv_node1` and `priv_node2`, so `CONNECT STRING` is set to `priv_node1`, `priv_node2`.
`CONFIG_DIR=`*directory-for- config.ini*	Specifies the directory where the MySQL Cluster configuration file `config.ini` and the management server's cache file are stored. You must specify a value for this parameter.	The `config.ini` file is placed under `/mgmd-data`, so the `CONFIG_DIR` variable is set to `/mgmd-data`.
`ID=`*mgmd-server-id*	Specifies the unique server ID for this management server. The value must match the entry in the `config.ini` file. You must specify a value for this parameter.	The unique ID on `priv_node1` is 1 and the unique ID on `priv_node2` is 2, so ID is set to `1`. The unique ID on `priv_node1` is 1 and the unique ID on `priv_node2` is 2, so ID is set to `2`.

▼ How to Create and Enable Resources for MySQL Cluster Management Server

Before You Begin Ensure that you have edited the `mysql_ndb_mgmd_config` file to specify configuration parameters for the HA for MySQL Cluster management server service. For more information, see "Specifying Configuration Parameters for the MySQL Cluster Management Server Resource" on page 27.

1. **Create the parameter file directory on all nodes to run the MySQL Cluster management server.**

```
# mkdir /pfile
```

2. **Create the parameter file on all nodes to run the MySQL Cluster management server.**

```
# ksh /opt/SUNWscmys/ndb_mgmd/util/mysql_ndb_mgmd_register \
-f /myplace/mysql_ndb_mgmd_config -p
```

3. **Register the resource for the MySQL Cluster management server on one node.**

```
# ksh /opt/SUNWscmys/ndb_mgmd/util/mysql_ndb_mgmd_register \
-f /myplace/mysql_ndb_mgmd_config
```

4. **Enable the MySQL Cluster management server resource group.**

```
# clresourcegroup online -eM mgm-rg
```

Specifying Configuration Parameters for the MySQL Cluster `ndb` Daemon Resource

HA for MySQL Cluster provides a script that automates the process of configuring the MySQL Cluster `ndb` daemon resource. This script obtains configuration parameters from the `mysql_ndbd_config` file. A template for this file is in the `/opt/SUNWscmys/ndbd/util` directory. To specify configuration parameters for the MySQL Cluster `ndb` daemon resource, copy the `mysql_ndbd_config` file to another directory and edit this `mysql_ndbd_config` file.

Note - This configuration file needs to be accessible from the global or local zone on each node where the MySQL Cluster is installed.

Each configuration parameter in the `mysql_ndbd_config` file is defined as a keyword-value pair. The `mysql_ndbd_config` file already contains the required keywords and equals signs.

For more information, see “`mysql_ndbd_config` File” on page 47. When you edit the `/myplace/mysql_ndbd_config` file, add the required value to each keyword.

The keyword-value pairs in the `mysql_ndbd_config` file are listed in the following table.

TABLE 5 Keyword-Value Pairs in the `mysql_ndbd_config` File

Parameter	Description	Example
`RS=`*ndb-daemon-resource*	Specifies the name that you are assigning to the MySQL Cluster `ndb` daemon resource. You must specify a value for this keyword.	The name of the MySQL Cluster `ndb` daemon resource is `ndb-rs`.
`RG=`*ndb-daemon-resource-group*	Specifies the name of the resource group where the MySQL Cluster `ndb` daemon resource will reside. You must specify a value for this keyword.	The name of the MySQL Cluster `ndb` daemon resource group is `ndb-rg`.
`HAS_RS=`*mgmd-resource,dependency-list*	Specifies a comma-separated list of resources that the MySQL Cluster `ndb` daemon depends on. You must include the resource name of the MySQL Cluster management server resource here.	The MySQL Cluster `ndb` daemon resource depends on the management server resource `mgm-rs`. So the `HAS_RS` variable is set to `mgm-rs`.
`RS_PROP=`*resource-property-string*	Specifies a string that is formatted as `-p` *property-value=value*.	Additional standard and extension properties.
`PARFILE=`*parameter-file*	Specifies the name of the parameter file where the MySQL Cluster `ndb` daemon specific parameters of the MySQL Cluster `ndb` daemon resource are stored. This file is automatically created at registration time. You must specify a value for this keyword.	The parameter file is `/pfile/ndbd-pfile`, so the `PARFILE` variable is set to `/pfile/ndbd-pfile`.
`BASEDIR=`*mysql-base-directory*	Specifies the directory where MySQL Cluster is installed. A valid `BASEDIR` value specifies a directory which contains `ndbd` under `bin` or `libexec`.	MySQL Cluster is installed in `/usr/local/mysql`, so the value of `BASEDIR` is `/usr/local/mysql`.
`USER=`*mgmd-user*	Specifies the user under which the MySQL Cluster `ndb` daemon is started. If you do not specify any value, the MySQL Cluster management server is started as root.	The MySQL Cluster `ndb` daemon should be started under the root user, so the `USER` variable is `empty`.
`TRY_RECONNECT=`*1*	Specifies how often a connection to the MySQL Cluster	If the first connection failure to a MySQL Cluster management

	management server should be retried before the attempt is abandoned. You must specify a value for this parameter.	server should lead to an abort, so the value of `TRY_RECONNECT` is `1`.
`CONNECT_STRING=`*mgm-connect-string*	Specifies a valid connect string for the management servers in the format `priv_nodename_1 [:port]`, `priv_nodename_2[: port]`.	The MySQL Cluster management servers are running on `priv_node1` and `priv_node2`, so `CONNECT STRING` is set to `priv_node1,priv_node2`.
`ID=`*ndb-server-id*	Specifies the unique server ID for this `ndb` daemon. The value must match the entry in the management servers `config.ini` file. You must specify a value for this parameter.	The unique id on `priv_node1` is `3` and the unique id on `priv_node2` is `4`. The `clprivnet` address on node 1 translates to `priv_node1`, so ID is set to `3`. The unique id on `priv_node1` is `3` and the unique id on `priv_node2` is `4`. The `clprivnet` address on node 2 translates to `priv_node2`, so `ID` is set to `4`.
`MULTI_THREAD=`*multithreading-trigger*	Any entry here triggers the start of the multithreaded version of the `ndb` daemon. Leaving this value undefined results in the start of the single-threaded `ndb` daemon.	The Multithreading trigger `MULTI_THREAD` is set to `y`.
`DATA_DIR=`*ndb data directory*	Specifies the directory where the `my.cnf` file and the data for the `ndb` daemon is stored. You must specify a value for this parameter.	The data directory for the `ndb` daemon is `/ndbd-data`, so `DATA_DIR` is set to `/ndbd-data`.
`ERROR_ON_SHOW=`*error-code-for-failed-mgm-connections*	Specifies the return code value for failed connections of the probe command to the management servers. The value should be less than or equal to 100 and follows the semantics for a `SUNW.gds probe` command. You must specify a value for this parameter.	The error code for failed management server connections is `25`, so `ERROR_ON_SHOW` is set to `25.`

▼ How to Create and Enable Resources for the MySQL Cluster ndb Daemon

Before You Begin Ensure that you have edited the mysql_ndbd_config file to specify configuration parameters for the HA for MySQL Cluster ndb daemon service. For more information, see "Specifying Configuration Parameters for the MySQL Cluster ndb Daemon Resource" on page 30.

1. **Create the parameter file directory on all nodes to run the MySQL Cluster ndb daemon, if you have not done so already.**

```
# mkdir /pfile
```

2. **Create the parameter file on all nodes to run the MySQL Cluster ndb daemon.**

```
# ksh /opt/SUNWscmys/ndbd/util/mysql_ndbd_register \
-f /myplace/mysql_ndbd_config -p
```

3. **Register the resource for the MySQL Cluster ndb daemon on one node.**

```
# ksh /opt/SUNWscmys/ndbd/util/mysql_ndbd_register \
-f /myplace/mysql_ndbd_config
```

4. **Enable the MySQL Cluster management server resource group.**

```
# clresourcegroup online -eM ndb-rg
```

Specifying Configuration Parameters for the MySQL Cluster ndb Shutdown Controller Resource

HA for MySQL Cluster provides a script that automates the process of configuring the MySQL Cluster ndb shutdown controller resource. This script obtains configuration parameters from the ndbd_shutdown_config file. A template for this file is in the /opt/SUNWscmys/ndbd_shutdown/util directory. To specify configuration parameters for the MySQL Cluster ndb shutdown controller resource, copy the ndbd_shutdown_config file to another directory and edit this ndbd_shutdown_config file.

Each configuration parameter in the ndbd_shutdown_config file is defined as a keyword-value pair. The ndbd_shutdown_config file already contains the required keywords and equals signs. For more information, see "ndb_shutdown_config File" on page 48. When you edit the /myplace/ndbd_shutdown_config file, add the required value to each keyword.

The keyword-value pairs in the ndbd_shutdown_config file are as follows.

TABLE 6 Keyword-Value Pairs in the ndbd_shutdown_config File

Parameter	Description	Example
RS=*ndb-shutdown-controller-resource*	Specifies the name that you are assigning to the MySQL Cluster ndb shutdown controller resource. You must specify a value for this keyword.	The name of the MySQL Cluster ndb daemon resource is ndb-shut-rs.
RG=*ndb-shutdown-controller-resource-group*	Specifies the name of the resource group where the MySQL Cluster ndb shutdown controller resource will reside. You must specify a value for this keyword.	The name of the MySQL Cluster ndb daemon resource group is ndb-rg.
HAS_RS=*ndbd-daemon-resource, dependency-list*	Specifies a comma-separated list of resources such that the MySQL Cluster ndb shutdown controller depends on. You must include the resource name of all such resources.	The MySQL Cluster ndb shutdown controller resource does depend on the ndb daemon resource ndb_rs. So the HAS_RS variable is set to ndb-rs.
RS_PROP=*resource-property-string*	Specifies a string that is formatted as -p *property-value=value*.	Additional standard and extension properties.
PARFILE=*ndbd-daemon-resource- parameter-file*	Specifies the name of the parameter file of one MySQL Cluster ndb daemon resource. You must specify a value for this keyword.	The parameter file is /pfile/ndbd-pfile, so the PARFILE variable is set to /pfile/ndbd-pfile.

▼ How to Create and Enable Resources for MySQL Cluster ndb Shutdown Controller

Before You Begin Ensure that you have edited the ndbd_shutdown_config file to specify configuration parameters for the HA for MySQL Clusterndb shutdown controller service. For more information, see "Specifying Configuration Parameters for the MySQL Cluster ndb Shutdown Controller Resource" on page 33.

1. **Register the resource for the MySQL Cluster ndb daemon on one node.**

```
# ksh /opt/SUNWscmys/ndbd_shutdown/util/ndb_shutdown_register \
-f /myplace/ndbd_shutdown_config
```

2. **Enable the MySQL Cluster management server resource group.**

```
# clresourcegroup online -eM ndb-rg
```

How to Suspend the `ndb` Shutdown Controller's Stop Action

If you want to take the `ndbd` daemon offline on one node only, you must suspend the stop action of the `ndb` shutdown controller. Otherwise, it restarts all `ndbd` daemons without loading data.

Perform the following task before you disable the shutdown controller resource on one node, or before you take offline the resource group that contains the `ndb` daemon resource and the `ndb` shutdown controller resource.

- **On one node, suspend the stop action of the `ndb` shutdown controller.**

  ```
  # touch /tmp/shutdown-controller-resourcename_noop
  ```

Note - During the stop, the `noop` file is removed.

How to Create and Enable Resources for the MySQL Cluster Server

Before You Begin Before you begin make sure you already bootstrapped the MySQL Cluster server on all nodes to host the MySQL Cluster server.

This procedure provides the general steps to register a MySQL Cluster server. For complete instructions, see "Registering and Configuring HA for MySQL Cluster" on page 27. However, the deployment example in Appendix B, "Deployment Example: Installing HA for MySQL Cluster in a Global Cluster" shows all the steps to their full extent.

If you want monitoring of an `ndb` table by each SQL server you are about to configure, set `NDB_CHECK=`*yes* in the configuration files `mysql_config` and `ha_mysql_config`. This setting creates a functional dependency between the MySQL Cluster servers and the MySQL Cluster data node processes. Consider deploying the MySQL Cluster server in a scalable or multiple-masters topology.

1. **Prepare the MySQL Cluster server on each node to run the MySQL Cluster server.**

a. **Execute the `mysqladmin` command.**

b. **Include the GRANT and UPDATE statements on each MySQL Cluster server you configured.**

   For more details about the GRANT and UPDATE statements for each of the MySQL Cluster server you have configured, see *Oracle Solaris Cluster Data Service for MySQL Guide* .

2. **Issue the `/opt/SUNWscmys/util/mysql_register` command on each MySQL Cluster server you configured.**

3. **(Optional) Make the `ha_mysql_config` file available on all nodes where the MySQL Cluster server is to run and execute.**

```
# /opt/SUNWscmys/util/ha_mysql_register -f config-file -e
```

-e (Optional) Specifies that you want password encryption.

4. **On one node, run the resource registration script.**

```
# /opt/SUNWscmys/util/ha_mysql_register -f config-file
```

Verifying the HA for MySQL Cluster Installation and Configuration

After you install, register, and configure HA for MySQL Cluster, verify this installation and configuration to determine whether the HA for MySQL Cluster data service makes your MySQL Cluster database highly available.

▼ How to Verify the HA for MySQL Cluster Installation and Configuration

1. **Assume the `root` role on a cluster node that is to host the MySQL Cluster component.**

2. **Determine whether resources are online by issuing the following commands for each resource.**

   If a resource is not online, use the `clresource enable` *resources* command to bring the resource online.

MySQL Cluster Component	Resource Name
MySQL Cluster management server	*mgm-rs*
MySQL Cluster `ndb` daemon	*ndb-rs*
MySQL Cluster `ndb` shutdown controller	*ndb-shut-rs*
MySQL Cluster server	*mysql-rs*

3. **Stop all MySQL Cluster resource groups.**

   a. **Stop the MySQL Cluster server resource group.**

   ```
   # clresourcegroup offline mysql-rg
   ```

   b. **Stop the MySQL Cluster `ndb` resource group.**

   ```
   # clresourcegroup offline ndb-rg
   ```

   c. **Stop the MySQL Cluster management server resource group.**

   ```
   # clresourcegroup offline mgm-rg
   ```

4. **Start all MySQL Cluster resource groups in reverse order.**

   a. **Start the MySQL Cluster management server resource group.**

   ```
   # clresourcegroup online mgm-rg
   ```

   b. **Start the MySQL Cluster `ndb` resource group.**

   ```
   # clresourcegroup online ndb-rg
   ```

   c. **Start the MySQL Cluster server resource group.**

   ```
    # clresourcegroup online mysql-rg
   ```

5. **Verify the status of all MySQL Cluster resource groups.**

```
# clresource status
# clresourcegroup status
```

Tuning the HA for MySQL Cluster Fault Monitor

The HA for MySQL Cluster fault monitor verifies that the data service is running in a healthy condition.

An HA for MySQL Cluster fault monitor is contained in each resource that represents the MySQL Cluster instance. You created these resources when you registered and configured HA for MySQL Cluster. For more information, see "Registering and Configuring HA for MySQL Cluster" on page 27.

System properties and extension properties of the MySQL Cluster resources control the behavior of the fault monitor. The default values of these properties determine the preset behavior of the fault monitor. Because the preset behavior should be suitable for most Oracle Solaris Cluster installations, tune the HA for MySQL Cluster fault monitor only if you need to modify this preset behavior.

Tuning the HA for MySQL Cluster fault monitor involves the following tasks, depending on the specific component:

- Setting the return value for failed MySQL Cluster monitor connections for the `ndb` daemon
- Setting the interval between fault monitor probes
- Setting the time out for fault monitor probes
- Defining the criteria for persistent faults
- Specifying the failover behavior of a resource

The fault monitor HA for MySQL Cluster `ndb` daemon differentiates between connection problems and definitive application failures. The value of `ERROR_ON_SHOW` in the MySQL Cluster `ndb` daemon parameter file specifies the return code for connection problems. This value results in a certain amount of ignored consecutive failed probes as long as they all return the value of `ERROR_ON_SHOW`. The first successful probe reverts this back to zero. The maximum number of failed probes is calculated as `100 / ERROR_ON_SHOW`. A definitive application failure will result in an immediate restart or failover.

The definition of the return value `ERROR_ON_SHOW` defines one of two behaviors for failed database connections of a MySQL Cluster `ndb` daemon resource.

- Retry the connection to the `ndb` management server several times before considering the MySQL Cluster `ndb` Daemon resource as failed and triggering a restart or failover.
- Complain at every probe that the connection to the test database failed. No restart or failover will be triggered.

To achieve either of these behaviors, use the standard resource properties `retry_interval` and `thorough_probe_interval`.

- A just complainingprobe is achieved as soon as the following equation is true: retry_interval < thorough_probe_interval *100/ERROR_ON_SHOW
- As soon as this equation is false, the MySQL Cluster ndb Daemon resource restarts after 100 / ERROR_ON_SHOW consecutive probe failures.

  The value 100/ERROR_ON_SHOW defines the maximum number of retries for the probe in the case of a failed connection.

  Assume that the following resource parameters are set:

  - thorough_probe_interval=90
  - retry_interval=660
  - ERROR_ON_SHOW=25

  If you encounter, for example, unresponsive management servers for 4.5 minutes, you will see three complaints in /var/adm/messages, but no resource restart. If the shortage lasts 6 minutes, you will have a restart of the MySQL Cluster ndb Daemon resource after the fourth probe.

  If you do not want a resource restart in the previous example, set the value of ERROR_ON_SHOW to 15 or less.

  For more information, see "Tuning Fault Monitors for Oracle Solaris Cluster Data Services" in *Oracle Solaris Cluster 4.3 Data Services Planning and Administration Guide*

This section contains the following additional information:

- "Operation of the HA for MySQL Cluster Management Server Parameter File" on page 39
- "Operation of the HA for MySQL Cluster ndb Daemon Parameter File" on page 40
- "Operation of the Fault Monitor for HA for MySQL Cluster Management Server" on page 41
- "Operation of the Fault Monitor for HA for MySQL Cluster ndb Daemon" on page 41

Operation of the HA for MySQL Cluster Management Server Parameter File

The HA for MySQL Cluster management server resources use a parameter file to pass parameters to the start, stop, and probe commands. Changes to these parameters take effect at least at every restart, or enabling, or disabling of the resource.

Changing one of the following parameters, takes effect at the next probe of the MySQL Cluster management server resource:

- BASEDIR

- USER
- TRY_RECONNECT
- CONNECT_STRING
- CONFIG_DIR ID

Note - An unexpected change of the parameters with an enabled MySQL Cluster management server resource might result in an unplanned service outage. To avoid such an outage, first disable the MySQL Cluster management server resource, execute the change, and then re-enable the resource.

Operation of the HA for MySQL Cluster ndb Daemon Parameter File

The HA for MySQL Cluster ndb daemon resources use a parameter file to pass parameters to the start, stop, and probe commands. Changes to these parameters take effect at least at every restart, or enabling, or disabling of the resource.

Changing one of the following parameters, takes effect at the next probe of the MySQL Cluster ndb daemon resource:

- BASEDIR
- USER
- TRY_RECONNECT
- CONNECT_STRING
- ID
- MULTI_THREAD
- DATA_DIR
- ERROR_ON_SHOW

⚠ **Caution -** Do not lower the Probe_timeout property of the ndbd daemon resource below 70 seconds. The probe algorithm relies on the presence of a management server. If the first physical node specified in the CONNECT_STRING is down, you will get a 60 seconds timeout. There must be enough time left, to run the probe request on the second node specified in the CONNECT_STRING.

Note - An unexpected change of the parameters with an enabled MySQL Cluster ndb daemon resource might result in an unplanned service outage. Therefore, disable the MySQL Cluster ndb Daemon resource first, execute the change, and then re-enable the resource.

Operation of the Fault Monitor for HA for MySQL Cluster Management Server

The fault monitor for HA for MySQL Cluster management server ensures that all the requirements for the MySQL Cluster management server component to run are met. These requirements include the following:

- The HA for MySQL Cluster management server `ndb_mgmd` process is running. If this process is not running, the fault monitor restarts the MySQL Cluster management server. If the fault persists, the fault monitor gives up on the resource group that contains the resource for the MySQL Cluster management server because it is a scalable or multiple-master resource.
- Connections to the MySQL Cluster management server are possible, and the `ndb_mgm STATUS` command does not show the value "not connected" for the selected server ID.

Operation of the Fault Monitor for HA for MySQL Cluster `ndb` Daemon

The fault monitor for HA for MySQL Cluster `ndb` daemon ensures that all the requirements for the MySQL Cluster `ndb` daemon component to run are met. These requirements include the following:

- The HA for MySQL Cluster `ndb` daemon `ndbd` or `ndbmtd` process is running, depending on the `MULTITHREAD` value at resource start time.
- If this process is not running, the fault monitor restarts the MySQL Cluster `ndb` daemon. If the fault persists, the fault monitor gives up the resource group that contains the resource for the MySQL Cluster `ndb` daemon, because it is a multiple-master resource.
- Connections to the MySQL Cluster `ndb` daemon management server are possible, and the `ndb_mgm STATUS` command show the value "started" or "starting" for the selected server ID. If the resource is waiting to be put online, only "started" is a legal value for the selected server ID.

  If the connection to the management server fails, the probe exits with the connection failed return code `ERROR_ON_SHOW`. If the `ndb_mgm` status command shows an illegal value, the fault monitor restarts the MySQL Cluster `ndb` daemon resource, if it is not in its wait for online phase.

Debugging HA for MySQL Cluster

Each HA for MySQL Cluster component has a file named `config` that enables you to activate debugging for MySQL Cluster resources. This file is in the `/opt/SUNWscmys/`*component*`/etc` directory.

▼ How to Activate Debugging for HA for MySQL Cluster

1. **Determine whether debugging for HA for MySQL Cluster is active.**

```
# grep daemon /etc/syslog.conf
*.err;kern.debug;daemon.notice;mail.crit    /var/adm/messages
*.alert;kern.err;daemon.err           operator
#
```

2. **Determine whether debugging is active.**

   - **If `daemon.debug` appears in the `/etc/syslog.conf` file of the global zone or the zone cluster's zone, debugging is active.**

     You do not need to continue with this procedure.

   - **If `daemon.notice` appears in the `/etc/syslog.conf` file of the appropriate zone, debugging is inactive.**

     Continue with the rest of this procedure.

3. **In the `/etc/syslog.conf` file in the appropriate zone, change `daemon.notice` to `daemon.debug`.**

4. **Restart the `syslogd` daemon in the appropriate zone.**

```
 # svcadm refresh svc:/system/system-log:default
```

5. **Edit the appropriate `/opt/SUNWscmys/`*component-name*`/etc/config` file to add a value to the `DEBUG=` parameter:**

   The value of the *component-name* can be `ndb_mgmd`, `ndbd`, or `ndbd_shutdown`.

   - To debug all resources, use `DEBUG=`*ALL*
   - To debug specific resources, use `DEBUG=`*resource-name*`, `*resource-name*

Example 1 Editing the Debug Configuration File

```
# cat /opt/SUNWscmys/ndb_mgmd/etc/config
#
# Copyright 2015 Oracle and/or its affiliates. All rights reserved.
# Use is subject to license terms.
#
# Usage:
#    DEBUG=RESOURCE-NAME or ALL
#
DEBUG=ALL
#
```

♦ ♦ ♦ APPENDIX A

Files for Configuring HA for MySQL Cluster

The /opt/SUNWscmys/*component-name*/util directory contains files that automate the process of configuring HA for MySQL Cluster resources. This appendix shows the contents of the following configuration files:

- "mysql_ndb_mgmd_config File" on page 45
- "mysql_ndbd_config File" on page 47
- "ndb_shutdown_config File" on page 48

mysql_ndb_mgmd_config File

```
#
# Copyright 2009, 2012 Oracle and/or its affiliates. All rights reserved.
# Use is subject to license terms.
#

#ident "@(#)mysql_ndb_mgmd_config.ksh 1.3   12/08/20"

# This file will be sourced in by mysql_ndb_mgmd_register and the parameters
# listed below will be used.
#
# These parameters can be customized in (key=value) form
#
#    RS - name of the resource for the application
#    RG - name of the resource group containing RS
#
# MySQL Cluster cluster ndb_nmbd specific variables
#
# SCALABLE - Any value here triggers a scalable resource creation
# LB_POLICY - Set the loadbalancing policy for a scalable MySQL Cluster service.
#        Use the values defined for the standare resource property
#        Load_balancing_policy. If you do not specify it, the defaults are
#        used.
#    LH - Name of the Shared Address SC resource
#  HAS_RS - Name of the MySQL Cluster HAStoragePlus SC resource
#  RS_PROP - Additional resource properties in the format for clresource create,
#     example "-p start_timeout=600"
#  PARFILE - Absolute path to the management server resource parameter file
#
#    The following examples illustrate sample parameters
#    for a multiple master MySQL Cluster management server resource.
#
#    RS=mgm-rs
#    RG=mgm-rg
#    PORT=1186
#    SCALABLE=
#    LB_POLICY=
#    HAS_RS=
#    PARFILE=/ndb-mgmd-data/pfile
#

RS=
RG=
PORT=
LH=
SCALABLE=
LB_POLICY=
HAS_RS=
RS_PROP=
PARFILE=


#    This is the template for a MySQL Cluster cluster's management server resource.
#    The variables must be specified in the key value form.
#    BASEDIR    Directory where MySQL Cluster cluster is installed, to find the
 binaries.
#    USER      User under which the management server will be run, an empty value
#           stands for the root user.
#    TRY_RECONNECT Value for the try-reconnect parameter in the mgm command.
#    CONNECT_STRING A connect string which is valid for any connection to the management
#           server.
#    CONFIG_DIR   Directory where the management server's config.ini file and its cache
#           files are stored.
#    ID       Unique Id for this management server. This value must match the entry
#           in the config.ini file.
#
#    Examples:
#    BASEDIR=/usr/local/mysql
#    USER=
#    TRY_RECONNECT=1
#    CONNECT_STRING=phys-node-1-priv,phys-node-2-priv
#    CONFIG_DIR=/ndb-nmbd-data
#    ID=1
#

BASEDIR=
USER=
TRY_RECONNECT=1
CONNECT_STRING=
CONFIG_DIR=
ID=
```

mysql_ndbd_config File

```
#
# Copyright 2009, 2012 Oracle and/or its affiliates. All rights reserved.
# Use is subject to license terms.
#

#ident "@(#)mysql_ndbd_config.ksh 1.3   12/08/20"

# This file will be sourced in by mysql_ndbd_register and the parameters
# listed below will be used.
#
# These parameters can be customized in (key=value) form
#
#    RS - Name of the resource for the application
#    RG - Name of the resource group containing RS
#
# MySQL Cluster cluster ndb_nmbd specific variables
#
#  HAS_RS - Name of the MySQL Cluster HAStoragePlus SC resource
#  RS_PROP - Additional resource properties in the format for clresource create,
#     example "-p start_timeout=600"
#  PARFILE - Absolute path to the data node resource parameter file
#
#    The following examples illustrate sample parameters
#    for the MySQL Cluster data node resource.
#
#    RS=ndbd-rs
#    RG=ndbd-rg
#    HAS_RS=
#    PARFILE=/ndb-mgmd-data/pfile
#

RS=
RG=
HAS_RS=
RS_PROP=
PARFILE=


#    This is the template for a MySQL Cluster cluster's management server resource.
#    The variables must be specified in the key value form.
#    BASEDIR    Directory where MySQL Cluster cluster is installed, to find the
 binaries.
#    USER      User under which the management server will be run. An empty value
#           stands for the root user.
#    TRY_RECONNECT Value for the try-reconnect parameter in the mgm command.
#    CONNECT_STRING A connect string which is valid for any connection to the management
#           server.
#    ID       Unique Id for this management server. This value must match the entry
#           in the config.ini file.
#    MULTI_THREAD  Any entry here will trigger the start of the multithreaded version
 ndbmtd
#           of the ndbd instead of the single-threaded version ndbd.
#    DATA_DIR    Data directory of the ndb process.
#    ERROR_ON_SHOW Return code if the probe is unable to connect to the management
 server.
#
#    Examples:
#    BASEDIR=/usr/local/mysql/bin
#    USER=
#    TRY_RECONNECT=1
#    CONNECT_STRING=phys-node-1-priv,phys-node-2-priv
#    ID=1
#    DATAG_DIR=/ndb-data
#    MULTI_THREAD=yes
#    ERROR_ON_SHOW=25
#

BASEDIR=
USER=
TRY_RECONNECT=1
CONNECT_STRING=
ID=
MULTI_THREAD=
DATA_DIR=
ERROR_ON_SHOW=
```

ndb_shutdown_config File

```
#
# Copyright 2009, 2012 Oracle and/or its affiliates. All rights reserved.
# Use is subject to license terms.
#

#ident "@(#)ndbd_shutdown_config.ksh 1.3   12/08/20"

# This file will be sourced in by ndbd_shutdown_register and the parameters
# listed below will be used.
#
# These parameters can be customized in (key=value) form
#
#    RS - Name of the resource for the application
#    RG - Name of the resource group containing RS
#
# MySQL Cluster cluster ndb_nmbd specific variables
#
#  HAS_RS - Name of the MySQL Cluster HAStoragePlus SC resource
#  RS_PROP - Additional resource properties in the format for clresource create,
#     example "-p start_timeout=600"
#  PARFILE - Absolute path to the data node resource parameter file
#       IMPORTANT: it is essential to specify the data node
#       parameter file.
#
#    The following examples illustrate sample parameters
#    for the MySQL Cluster ndb shutdown resource.
#
#    RS=ndbd-shut-rs
#    RG=ndbd-rg
#    HAS_RS=ndbd-rs
#    RS_PROP=
#    PARFILE=/ndbd-data/pfile
#

RS=
RG=
HAS_RS=
RS_PROP=
PARFILE=
```

♦ ♦ ♦ APPENDIX B

Deployment Example: Installing HA for MySQL Cluster in a Global Cluster

This appendix presents a complete example of how to install and configure the HA for MySQL Cluster data service in a global cluster. It presents a simple two-node cluster configuration. If you need to install the application in any other configuration, refer to the general-purpose procedures presented elsewhere in this manual.

Target Cluster Configuration

This example uses a two-node cluster with the following node names:

- `phys-schost-1` (a global-cluster node)
- `phys-schost-2` (a global-cluster node)

This configuration also uses the shared address host name `sa-host-1`.

The `clprivnet` addresses used by the MySQL Cluster configuration use the following names:

- `phys-schost-1-p` running on `phys-schost-1`
- `phys-schost-2-p` running on `phys-schost-2`

Software Configuration

This deployment example uses the following software products and versions:

- Oracle Solaris 11 software for SPARC or x86 platforms
- Oracle Solaris Cluster 4 core software
- Oracle Solaris Cluster HA for MySQL Cluster

- MySQL Cluster version 7.0.7 tar file

This example assumes that you have already installed and established your cluster. It illustrates installation and configuration of the data service application only.

The projected configuration is as follows:

- A scalable service for the management servers
- A multiple-master service for the data nodes
- A scalable service for the SQL nodes

The general tasks are:

1. Install the MySQL Cluster software.
2. Set up cluster control.
3. Initialize the MySQL Cluster server and shut down the components.
4. Create the resource groups and resources.

The following example procedures are provided in this section:

- "How to Install MySQL Cluster Software" on page 52
- "Setting up the MySQL Cluster Control" on page 53
- "How to Create the HA for MySQL Cluster Configuration With Scalable Services" on page 59

You will also need several configuration files, which are shown in Appendix A, "Files for Configuring HA for MySQL Cluster".

On both nodes there is a file containing the MySQL Cluster available in `/temp/mysql-7.0.7-solaris11-sparc.tar.gz`.

How to Install MySQL Cluster Software

1. **Create a group and user on both nodes.**

```
phys-schost-1:/ # groupadd -g 200 dba
phys-schost-2:/ # groupadd -g 200 dba

phys-schost-1:/ # useradd -u 1000 -g dba -d /export/mysql -s /usr/bin/bash mysql
phys-schost-2:/ # useradd -u 1000 -g dba -d /export/mysql -s /usr/bin/bash mysql
```

2. **Install the tar file on both nodes.**

```
phys-schost-1:/ # mkdir /usr/local
```

```
phys-schost-2:/ # mkdir /usr/local

phys-schost-1:/ # cd /usr/local
phys-schost-2:/ # cd /usr/local

phys-schost-1:/usr/local # /usr/sfw/bin/gtar xzf /temp/mysql-7.0.7-solaris10-
sparc.tar.gz
phys-schost-2:/usr/local # /usr/sfw/bin/gtar xzf /temp/mysql-7.0.7-solaris10-
sparc.tar.gz

phys-schost-1:/usr/local # ln -s ./mysql-7.0.7-solaris10-sparc ./mysql
phys-schost-2:/usr/local # ln -s ./mysql-7.0.7-solaris10-sparc ./mysql
```

3. **Set the ownership.**

```
phys-schost-1:/usr/local # chown -RL mysql:dba ./mysql
phys-schost-2:/usr/local # chown -RL mysql:dba ./mysql
```

Setting up the MySQL Cluster Control

This section contains the following example procedures:

- "How to Configure the Management Server on Both Nodes" on page 53
- "How to Configure the Data Nodes on Both Nodes" on page 55
- "How to Initialize the MySQL Cluster Server" on page 56

▼ How to Configure the Management Server on Both Nodes

1. **Create the configuration.**

   a. **On both nodes, create the data directory for the management server.**

   ```
   phys-schost-1:/ # mkdir /mgm-data
   phys-schost-2:/ # mkdir /mgm-data
   ```

   b. **On both nodes, copy the `config.ini` file from `/temp/cluconfig` to the `mgm-data` directory.**

   ```
   phys-schost-1:/ # cp /temp/cluconfig/config.ini /mgm-data
   phys-schost-2:/ # cp /temp/cluconfig/config.ini /mgm-data
   ```

   c. **Modify the `config.ini` file from `/temp/cluconfig`.**

Alternatively, copy the content from "config.ini File for Both Nodes to Store in /mgm-data" on page 63 and overwrite the copied file.

The configuration in the config.ini file for this example is shown in the following table:

Server ID	Node Type	Global-Cluster Node to Run On	Private Net Alias
1	Management node	phys-schost-1	
2	Management node	phys-schost-2	
3	Date node	phys-schost-1	phys-schost-1-p
4	Date node	phys-schost-2	phys-schost-2-p
7	SQL node	phys-schost-1	
8	SQL node	phys-schost-2	

d. **Configure the data nodes to communicate over the private interconnect clprivnet addresses.**

Create aliases in the /etc/hosts table for the clprivnet addresses and use them in the config.ini file as the host names.

e. **Set Arbitration=WaitExternal and an appropriate value for ArbitrationTimeout in the config.inifile.**

2. **On one node, set the heartbeat timeouts for Oracle Solaris Cluster software.**

```
phys-schost-1:/ # cluster set -p heartbeat_quantum=500 -p heartbeat_timeout=5000
```

Note - The heartbeat timeout must be half of the ArbitrationTimeout in the config.inifile.

3. **Start the management server.**

```
phys-schost-1:/ # cd /mgm-data
phys-schost-2:/ # cd /mgm-data

phys-schost-1:/mgm-data # /usr/local/mysql/bin/ndb_mgmd \
--configdir=/mgm-data -f /mgm-data/config.ini \
--ndb-nodeid=1

phys-schost-2:/mgm-data # /usr/local/mysql/bin/ndb_mgmd \
--configdir=/mgm-data -f /mgm-data/config.ini \
--ndb-nodeid=2
```

4. **Verify that the management server is running.**

Run the `ndb_mgm show` command on both nodes until the data nodes are connected to the management server.

```
phys-schost-1:/mgm-data # /usr/local/mysql/bin/ndb_mgm \
--ndb-connectstring=phys-schost-1-p,phys-schost-2-p -e show

phys-schost-2:/mgm-data # /usr/local/mysql/bin/ndb_mgm \
--ndb-connectstring=phys-schost-2-p,phys-schost-1-p -e show
```

▼ How to Configure the Data Nodes on Both Nodes

1. **Create the configuration.**

   a. **On both nodes create the data directory for the management server.**

   ```
   phys-schost-1:/ # mkdir /ndbd-data
   phys-schost-2:/ # mkdir /ndbd-data
   ```

   b. **Copy the `my.cnf_ndbd` file from `/temp/cluconfig` in the `ndbd-data` directory.**

   ```
   phys-schost-1:/ # cp /temp/cluconfig/my.cnf_ndbd /ndbd-data/my.cnf
   phys-schost-2:/ # cp /temp/cluconfig/my.cnf_ndbd /ndbd-data/my.cnf
   ```

   c. **Modify the `my.cnf_ndbd` file from `/temp/cluconfig`.**

   Alternatively, copy the content from "`my.cnf` File for the Data Nodes to Store in `/ndbd-data`" on page 66 and overwrite the copied file.

2. **Start the data nodes and verify the settings.**

```
phys-schost-1:/ # cd /ndbd-data
phys-schost-2:/ # cd /ndbd-data

phys-schost-1:/ndbd-data # /usr/local/mysql/bin/ndbd \
--defaults-file=/ndbd-data/my.cnf \
--ndb-connectstring=phys-schost-1-p:1186,phys-schost-2-p:1186 --ndb-nodeid=3

phys-schost-2:/ndbd-data # /usr/local/mysql/bin/ndbd \
--defaults-file=/ndbd-data/my.cnf \
--ndb-connectstring=phys-schost-1-p:1186,phys-schost-2-p:1186 --ndb-nodeid=4
```

3. **On one node, check the data nodes.**

```
phys-schost-1:/ndbd-data # /usr/local/mysql/bin/ndb_mgm \
--ndb-connectstring=phys-schost-1-p,phys-schost-2-p -e show
```

Note - Repeat the `show` command until both `ndbd` processes are fully up and running.

Example output:

```
phys-schost-1:/ # /usr/local/mysql/bin/ndb_mgm
--ndb-connectstring=phys-schost-1-p,phys-schost-2-p -e show
Connected to Management Server at: phys-schost-1-p:1186
Cluster Configuration
---------------------
[ndbd(NDB)]   2 node(s)
id=3  @172.16.4.2 (mysql-5.1.35 ndb-7.0.7, Nodegroup: 0, Master)
id=4  @172.16.4.1 (mysql-5.1.35 ndb-7.0.7, Nodegroup: 0)

[ndb_mgmd(MGM)] 2 node(s)
id=1  @172.16.4.2 (mysql-5.1.35 ndb-7.0.7)
id=2  @172.16.4.1 (mysql-5.1.35 ndb-7.0.7)

[mysqld(API)]  2 node(s)
id=7 (not connected, accepting connect from phys-schost-1)
id=8 (not connected, accepting connect from phys-schost-2)
```

▼ How to Initialize the MySQL Cluster Server

Create the directory structure and configuration by performing the following steps on both nodes

1. **Create the directory structure.**

```
phys-schost-1:/ # mkdir -p /mysql-data/logs /mysql-data/innodb
phys-schost-2:/ # mkdir -p /mysql-data/logs /mysql-data/innodb
```

2. **Initialize the MySQL Cluster database.**

```
phys-schost-1: # cd /usr/local/mysql
phys-schost-1:/usr/local/mysql # ./scripts/* --datadir=/mysql-data

phys-schost-2: # cd /usr/local/mysql
phys-schost-2:/usr/local/mysql # ./scripts/* --datadir=/mysql-data
```

3. **Prepare the `my.cnf-serv` file in `/temp/cluconfig`.**

   Use the example values in the “`my.cnf` File for the First SQL Node `phys-schost-1` to Store in `/mysql-data`” on page 66 for `phys-schost-1`, and “`my.cnf` File for the Second SQL Node `phys-schost-2` to Store in `/mysql-data`” on page 68 for `phys-schost-2`.

```
phys-schost-1:/ # cp /temp/cluconfig/my.cnf-serv /mysql-data/my.cnf
phys-schost-1:/ # chmod 644 /mysql-data/my.cnf

phys-schost-2:/ # cp /temp/cluconfig/my.cnf-serv /mysql-data/my.cnf
phys-schost-2:/ # chmod 644 /mysql-data/my.cnf
```

4. **Copy `my.cnf` file from `/temp/cluconfig/my.cnf-serv`, adjust the parameters as appropriate, and change the permissions.**

5. **Change the ownership of the data directory.**

   ```
   phys-schost-1:/ # chown -R mysql:dba /mysql-data
   phys-schost-2:/ # chown -R mysql:dba /mysql-data
   ```

6. **Start the MySQL Cluster server for the first time and specify the grants.**

   a. **Create a start script on both nodes with the appropriate values.**

   ```
   phys-schost-1:/ # cat >/temp/cluconfig/first <<EOF
   /usr/local/mysql/bin/mysqld --defaults-file=/mysql-data/my.cnf \
   --basedir=/usr/local/mysql --datadir=/mysql-data \
   --pid-file=/mysql-data/mysqld.pid \
   --user=mysql >> /mysql-data/logs/phys-schost-1.log 2>&1 &
   EOF

   phys-schost-2:/ # cat >/temp/cluconfig/first <<EOF
   /usr/local/mysql/bin/mysqld --defaults-file=/mysql-data/my.cnf \
   --basedir=/usr/local/mysql --datadir=/mysql-data \
   --pid-file=/mysql-data/mysqld.pid \
   --user=mysql >> /mysql-data/logs/phys-schost-2.log 2>&1 &
   EOF
   ```

   b. **Execute the start script `/temp/cluconfig/first` on both nodes.**

   ```
   phys-schost-1:/ # . /temp/cluconfig/first
   phys-schost-2:/ # . /temp/cluconfig/first
   ```

7. **Wait 60 seconds and verify that the MySQL Cluster servers connect to the `ndb` data nodes.**

   a. **On one node, issue the following command.**

   ```
   phys-schost-2:/ # /usr/local/mysql/bin/ndb_mgm \
   --ndb-connectstring=phys-schost-1,phys-schost-2 -e show
   ```

   b. **Set the administrative password on both nodes.**

```
phys-schost-1:/ # /usr/local/mysql/bin/mysqladmin -S /tmp/phys-schost-1.sock \
-uroot password 'root'

phys-schost-2:/ # /usr/local/mysql/bin/mysqladmin -S /tmp/phys-schost-2.sock \
-uroot password 'root'
```

c. **Define the administrative user.**

```
phys-schost-1:/ # /usr/local/mysql/bin/mysql -S /tmp/phys-schost-1.sock -uroot -
proot
mysql> use mysql;
mysql> grant all on *.* to 'root'@'phys-schost-1' identified by 'root';
mysql> UPDATE user SET Grant_priv='Y' WHERE User='root' AND Host='phys-schost-1';
mysql> exit

phys-schost-2:/ # /usr/local/mysql/bin/mysql -S /tmp/phys-schost-2.sock -uroot -
proot
mysql> use mysql;
mysql> grant all on *.* to 'root'@'phys-schost-2' identified by 'root';
mysql> UPDATE user SET Grant_priv='Y' WHERE User='root' AND Host='phys-schost-2';
mysql> exit
```

8. **Prepare the MySQL Cluster server for Oracle Solaris Cluster usage.**

a. **Prepare a `mysql_config` file under `/temp/cluconfig`.**

Use the content from "`mysql_config` File for the First SQL Node `phys-schost-1` to Store in `/temp/cluconfig`" on page 69 on `phys-schost-1`, and the content from "`mysql_config` File for the Second SQL Node `phys-schost-2` to Store in `/temp/cluconfig`" on page 70 on `phys-schost-2`.

b. **Set the `MYSQL_NIC_HOSTNAME` values.**

On `phys-schost-1`

```
MYSQL_NIC_HOSTNAME=" phys-schost-1 "
```

On `phys-schost-2`

```
MYSQL_NIC_HOSTNAME=" phys-schost-2 "
```

c. **On both nodes, execute the following commands.**

```
phys-schost-1:/ # ksh /opt/SUNWscmys/util/mysql_register \
-f /temp/cluconfig/mysql_config

phys-schost-2:/ # ksh /opt/SUNWscmys/util/mysql_register \
-f /temp/cluconfig/mysql_config
```

d. **Shut down the MySQL Cluster server on both nodes.**

```
phys-schost-1:/ # pkill -f mysqld
phys-schost-2:/ # pkill -f mysqld
```

e. **From the global zone of one node, shut down the MySQL Cluster components.**

```
phys-schost-1:/ # /usr/local/mysql/bin/ndb_mgm \
--ndb-connectstring=phys-schost-1-p,phys-schost-2-p -e shutdown
```

f. **Verify the shutdown on both nodes.**

```
phys-schost-1:/ # /usr/local/mysql/bin/ndb_mgm \
--ndb-connectstring=phys-schost-1-p,phys-schost-2-p -e show
phys-schost-2:/ # /usr/local/mysql/bin/ndb_mgm \
--ndb-connectstring=phys-schost-2-p,phys-schost-1-p -e show
```

g. **Shut down potentially running daemons.**

```
phys-schost-1:/ # /usr/local/mysql/bin/ndb_mgm \
--ndb-connectstring=phys-schost-1-p,phys-schost-2-p -e "id stop"
```

▼ How to Create the HA for MySQL Cluster Configuration With Scalable Services

1. **On one node, create the resource groups.**

```
phys-schost-1:/ # clresourcegroup create access-rg
phys-schost-1:/ # clressharedaddress create -g access-rg sa_host_1
phys-schost-1:/ # clresourcegroup online -eM access-rg
phys-schost-1:/ # clresourcegroup create -p Maximum_primaries=2 -p Desired_primaries=2
 mgm-rg
phys-schost-1:/ # clresourcegroup create -p Maximum_primaries=2 -p Desired_primaries=2
 ndbd-rg
phys-schost-1:/ # clresourcegroup create -p Maximum_primaries=2 -p Desired_primaries=2
 mysql-rg
phys-schost-1:/ # clresourcegroup set -p RG_affinities=++ndbd-rg mysql-rg
```

Note - Setting the ++ affinity ensures that on a restart of a single node, the start order of the resources is maintained as set within the resource dependencies.

2. **On both nodes, create a configuration directory for the parameter file.**

```
phys-schost-1:/ # mkdir /cluster-pfiles
phys-schost-2:/ # mkdir /cluster-pfiles
```

3. **On one node, register `gds`.**

```
phys-schost-1:/ # clresourcetype register gds
```

4. **Create the resource for the management daemon.**

   a. **Create a configuration file on both nodes under `/temp/cluconfig/mysql_ndb_mgmd_config`.**

   Use the content of "`mysql_ndb_mgmd_config` File for the First Node `phys-schost-1`" on page 71 for `phys-schost-1` and "`mysql_ndb_mgmd_config` File for the Second Node `phys-schost-2`" on page 72 for `phys-schost-2`.

   b. **Ensure that the ID parameter on each node reflects the ID in the `config.ini` file.**

```
ID=1 for phys-schost-1
ID=2 for phys-schost-2
```

   c. **Ensure that the connect string contains the global-cluster node name.**

   Value for `phys-schost-1`:

```
CONNECT_STRING=phys-schost-1,phys-schost-2
```

   Value for `phys-schost-2`:

```
CONNECT_STRING=phys-schost-2,phys-schost-1
```

   d. **On both nodes, create the parameter file.**

```
phys-schost-1:/ # ksh /opt/SUNWscmys/ndb_mgmd/util/mysql_ndb_mgmd_register \
-f /temp/cluconfig/mysql_ndb_mgmd_config -p

phys-schost-2:/ # ksh /opt/SUNWscmys/ndb_mgmd/util/mysql_ndb_mgmd_register \
-f /temp/cluconfig/mysql_ndb_mgmd_config -p
```

   e. **On one node, create the resource and start the `mgm-rg`.**

```
phys-schost-1:/ # ksh /opt/SUNWscmys/ndb_mgmd/util/mysql_ndb_mgmd_register \
-f /temp/cluconfig/mysql_ndb_mgmd_config
phys-schost-1:/ # clresourcegroup online -eM mgm-rg
```

```
phys-schost-1:/ # /usr/local/mysql/bin/ndb_mgm \
--ndb-connectstring=phys-schost-1-p,phys-schost-2-p -e show
phys-schost-1:/ # /usr/local/mysql/bin/ndb_mgm \
--ndb-connectstring=phys-schost-2-p,phys-schost-1-p -e show
```

5. **Create the resource for the `ndbd` daemon.**

   a. **Create a configuration file on both nodes under `/temp/cluconfig/mysql_ndbd_config`.**

      Use the content of "`mysql_ndbd_config` File for the First Node `phys-schost-2`" on page 74 for `phys-schost-1` and "`mysql_ndbd_config` File for the Second Node `phys-schost-2`" on page 75 for `phys-schost-2`.

   b. **Ensure that the ID parameter on each node reflects the ID in the `config.ini` file.**

      ```
      ID=3 for phys-schost-1
      ID=4 for phys-schost-2
      ```

   c. **On both nodes, create the parameter file.**

      ```
      phys-schost-1:/ # ksh /opt/SUNWscmys/ndbd/util/mysql_ndbd_register \
      -f /temp/cluconfig/mysql_ndbd_config -p

      phys-schost-2:/ # ksh /opt/SUNWscmys/ndbd/util/mysql_ndbd_register \
      -f /temp/cluconfig/mysql_ndbd_config -p
      ```

   d. **On one node, create the resource and start the `ndbd-rg` resource.**

      ```
      phys-schost-1:/ # ksh /opt/SUNWscmys/ndbd/util/mysql_ndbd_register \
      -f /temp/cluconfig/mysql_ndbd_config
      phys-schost-1:/ # clresourcegroup online -eM ndbd-rg
      ```

Note - Do not take the `ndbd-rg` offline until you create and enable the shutdown controller resource.

6. **Create the resource for the shutdown controller.**

   a. **On one node, create a configuration file under `/temp/cluconfig/ndbd_shutdown_config`.**

      Use the content of "`ndbd_shutdown_config` File for One Node" on page 77.

b. **On one node, create the resource and start the `ndbd-rg` resource.**

```
phys-schost-1:/ # ksh /opt/SUNWscmys/ndbd_shutdown/util/ndbd_shutdown_register \
-f /temp/cluconfig/ndbd_shutdown_config
phys-schost-1:/ # clresourcegroup online -e ndbd-rg
```

Note - From this point, never take offline on all the servers only the `ndbd` resource. If you want to shut down the `ndbd` completely, either use the `clresourcegroup` command to take `ndbd-rg` offline or first disable the shutdown controller resource.

If you want to shut down an `ndbd` resource on one node only (performing a rolling restart), you can disable it with the `clresource disable -n phys-schost-1 ndbd-rs` command. In this case, re-enable the `ndbd` resource before you shut down another resource.

For a rolling restart, do not disable the shutdown controller resource. Doing so would lead to a restart of the `ndbd` without loading data, in which case your database would be unavailable.

7. **On one node, create the resource for the MySQL Cluster server.**

a. **Create a configuration file under `/temp/cluconfig/ha_mysql_config` using the content shown in "`ha_mysql_config` File for One Node" on page 78.**

b. **Create the resource and start the `ndbd-rg` resource group.**

```
phys-schost-1:/ # ksh /opt/SUNWscmys/util/ha_mysql_register \
-f /temp/cluconfig/ha_mysql_config
phys-schost-1:/ # clresourcegroup online -eM mysql-rg
```

Example Configuration Files for Installation in a Global Cluster

This section contains example contents for the following files:

- "`config.ini` File for Both Nodes to Store in `/mgm-data`" on page 63
- "`my.cnf` File for the Data Nodes to Store in `/ndbd-data`" on page 66
- "`my.cnf` File for the First SQL Node `phys-schost-1` to Store in `/mysql-data`" on page 66
- "`my.cnf` File for the Second SQL Node `phys-schost-2` to Store in `/mysql-data`" on page 68

- "`mysql_config` File for the First SQL Node `phys-schost-1` to Store in `/temp/cluconfig`" on page 69
- "`mysql_config` File for the Second SQL Node `phys-schost-2` to Store in `/temp/cluconfig`" on page 70
- "`mysql_ndb_mgmd_config` File for the First Node `phys-schost-1`" on page 71
- "`mysql_ndb_mgmd_config` File for the Second Node `phys-schost-2`" on page 72
- "`mysql_ndbd_config` File for the First Node `phys-schost-2`" on page 74
- "`mysql_ndbd_config` File for the Second Node `phys-schost-2`" on page 75
- "`ndbd_shutdown_config` File for One Node" on page 77
- "`ha_mysql_config` File for One Node" on page 78

`config.ini` File for Both Nodes to Store in `/mgm-data`

```
[TCP DEFAULT]
SendBufferMemory=2M
ReceiveBufferMemory=2M

[NDB_MGMD DEFAULT]
PortNumber=1186
Datadir=/mgm-data/


[NDB_MGMD]
Id=1
Hostname=phys-schost-1-p

[NDB_MGMD]
Id=2
Hostname=phys-schost-2-p

[NDBD DEFAULT]
NoOfReplicas=2
Datadir=/ndbd-data/
DataMemory=256M
IndexMemory=32M
LockPagesInMainMemory=0
StopOnError=FALSE
Arbitration=WaitExternal
ArbitrationTimeout=10000

MaxNoOfConcurrentOperations=100000
```

```
StringMemory=25
MaxNoOfTables=4096
MaxNoOfOrderedIndexes=2048
MaxNoOfUniqueHashIndexes=512
MaxNoOfAttributes=24576
DiskCheckpointSpeedInRestart=100M
FragmentLogFileSize=256M
InitFragmentLogFiles=FULL
NoOfFragmentLogFiles=3
RedoBuffer=32M

TimeBetweenLocalCheckpoints=20
TimeBetweenGlobalCheckpoints=1000
TimeBetweenEpochs=100

MemReportFrequency=30
BackupReportFrequency=10

### Params for setting logging
LogLevelStartup=15
LogLevelShutdown=15
LogLevelCheckpoint=8
LogLevelNodeRestart=15

### Params for increasing disk throughput
BackupMaxWriteSize=1M
BackupDataBufferSize=16M
BackupLogBufferSize=4M
BackupMemory=20M
#Reports indicate that odirect=1 can cause io errors (os err code 5) on some systems.
 You must test.
#ODirect=1

### Watchdog
TimeBetweenWatchdogCheckInitial=30000

### TransactionInactiveTimeout - should be enabled in Production
#TransactionInactiveTimeout=30000
### CGE 6.3 - REALTIME EXTENSIONS
#RealTimeScheduler=1
#SchedulerExecutionTimer=80
#SchedulerSpinTimer=40

### DISK DATA
#SharedGlobalMemory=384M
#DiskPageBufferMemory=3072M
```

```
### Multithreading
MaxNoOfExecutionThreads=2
BatchSizePerLocalScan=512
[NDBD]
Id=3
Hostname=phys-schost-1-p

### CGE 6.3 - REALTIME EXTENSIONS
### PLEASE NOTE THAT THE BELOW ONLY WORKS IF YOU HAVE >1 CORE.
### YOU SHOULD CHECK cat /proc/interrupts AND CHOOSE THE CPUs
### THAT GENERATE THE LEAST INTERRUPS. TYPICALLY THE CPU HANDLING
### THE INTERRUPTS FOR THE COMMUNICATION INTERFACE USED FOR THE DATA NODE SHOULD
### BE AVOIDED FOR THE LockExecuteThreadToCPU, BUT YOU CAN
### LockMaintThreadsToCPU TO THAT CPU SINCE IT DOES NOT AFFECT THE
### REALTIME ASPECTS (THIS IS TRUE FOR UP TO TWO DATA NODES ON ONE COMPUTER.
#LockExecuteThreadToCPU=X
#LockMaintThreadsToCPU=Y

[NDBD]
Id=4
Hostname=phys-schost-2-p

### CGE 6.3 - REALTIME EXTENSIONS
### PLEASE NOTE THAT THE BELOW ONLY WORKS IF YOU HAVE >1 CORE.
### YOU SHOULD CHECK cat /proc/interrupts AND CHOOSE THE CPUs
### THAT GENERATE THE LEAST INTERRUPS. TYPICALLY THE CPU HANDLING
### THE INTERRUPTS FOR THE COMMUNICATION INTERFACE USED FOR THE DATA NODE SHOULD
### BE AVOIDED FOR THE LockExecuteThreadToCPU, BUT YOU CAN
### LockMaintThreadsToCPU TO THAT CPU SINCE IT DOES NOT AFFECT THE
### REALTIME ASPECTS (THIS IS TRUE FOR UP TO TWO DATA NODES ON ONE COMPUTER.
#LockExecuteThreadToCPU=X
#LockMaintThreadsToCPU=Y

##   BELOW ARE TWO (INACTIVE) SLOTS FOR DATA NODES TO ALLOW FOR GROWTH
#[NDBD]
#Id=5
#Hostname=

### CGE 6.3 - REALTIME EXTENSIONS
### PLEASE NOTE THAT THE BELOW ONLY WORKS IF YOU HAVE >1 CORE.
### YOU SHOULD CHECK cat /proc/interrupts AND CHOOSE THE CPUs
### THAT GENERATE THE LEAST INTERRUPS. TYPICALLY THE CPU HANDLING
### THE INTERRUPTS FOR THE COMMUNICATION INTERFACE USED FOR THE DATA NODE SHOULD
### BE AVOIDED FOR THE LockExecuteThreadToCPU, BUT YOU CAN
### LockMaintThreadsToCPU TO THAT CPU SINCE IT DOES NOT AFFECT THE
### REALTIME ASPECTS (THIS IS TRUE FOR UP TO TWO DATA NODES ON ONE COMPUTER.
#LockExecuteThreadToCPU=X
#LockMaintThreadsToCPU=Y
```

```
#[NDBD]
#Id=6
#Hostname=

### CGE 6.3 - REALTIME EXTENSIONS
### PLEASE NOTE THAT THE BELOW ONLY WORKS IF YOU HAVE >1 CORE.
### YOU SHOULD CHECK cat /proc/interrupts AND CHOOSE THE CPUs
### THAT GENERATE THE LEAST INTERRUPS. TYPICALLY THE CPU HANDLING
### THE INTERRUPTS FOR THE COMMUNICATION INTERFACE USED FOR THE DATA NODE SHOULD
### BE AVOIDED FOR THE LockExecuteThreadToCPU, BUT YOU CAN
### LockMaintThreadsToCPU TO THAT CPU SINCE IT DOES NOT AFFECT THE
### REALTIME ASPECTS (THIS IS TRUE FOR UP TO TWO DATA NODES ON ONE COMPUTER.
#LockExecuteThreadToCPU=X
#LockMaintThreadsToCPU=Y

[MYSQLD DEFAULT]
BatchSize=512
#BatchByteSize=2048K
#MaxScanBatchSize=2048K

[MYSQLD]
Id=7
Hostname=phys-schost-1

[MYSQLD]
Id=8
Hostname=phys-schost-2
```

`my.cnf` File for the Data Nodes to Store in `/ndbd-data`

```
# Options for ndbd process:
[mysql_cluster]
ndb-connectstring=phys-schost-1-p:1186,phys-schost-2-p:1186 # location of management
 server
```

`my.cnf` File for the First SQL Node `phys-schost-1` to Store in `/mysql-data`

```
[mysqld]
server-id=1
```

```
#port=3306
socket=/tmp/phys-schost-1.sock
log=/mysql-data/logs/log1
log-bin=/mysql-data/logs/bin-log
binlog-ignore-db=sc3_test_database
log-slow-queries=/mysql-data/logs/log-slow-queries
#log-update=/mysql-data/logs/log-update

# Innodb
#skip-innodb
loose-innodb_data_home_dir = /mysql-data/innodb
loose-innodb_data_file_path = ibdata1:10M:autoextend
loose-innodb_log_group_home_dir = /mysql-data/innodb
#innodb_log_arch_dir = /mysql-data/innodb
# You can set .._buffer_pool_size up to 50 - 80 %
# of RAM but beware of setting memory usage too high
loose-set-variable = innodb_buffer_pool_size=50M
loose-set-variable = innodb_additional_mem_pool_size=20M
# Set .._log_file_size to 25 % of buffer pool size
loose-set-variable = innodb_log_file_size=12M
loose-set-variable = innodb_log_buffer_size=4M
loose-innodb_flush_log_at_trx_commit=1
loose-set-variable = innodb_lock_wait_timeout=50

# MySQL Cluster 4.x
relay-log=/mysql-data/logs/slave-bin.log
relay-log-info-file=/mysql-data/logs/slave-info

# changes for cluster
#[mysqld]
ndbcluster
ndb-connectstring=phys-schost-1,phys-schost-2

# provide connectstring for management server host (default port: 1186)
[ndbd]
connect-string=phys-schost-1,phys-schost-2

# provide connectstring for management server host (default port: 1186)
[ndb_mgm]
connect-string=phys-schost-1,phys-schost-2

# provide location of cluster configuration file
[ndb_mgmd]
config-file=/mgm-data/config.ini
```

`my.cnf` File for the Second SQL Node `phys-schost-2` to Store in `/mysql-data`

```
[mysqld]
server-id=1
#port=3306
socket=/tmp/phys-schost-2.sock
log=/mysql-data/logs/log1
log-bin=/mysql-data/logs/bin-log
binlog-ignore-db=sc3_test_database
log-slow-queries=/mysql-data/logs/log-slow-queries
#log-update=/mysql-data/logs/log-update

# Innodb
#skip-innodb
loose-innodb_data_home_dir = /mysql-data/innodb
loose-innodb_data_file_path = ibdata1:10M:autoextend
loose-innodb_log_group_home_dir = /mysql-data/innodb
#innodb_log_arch_dir = /mysql-data/innodb
# You can set .._buffer_pool_size up to 50 - 80 %
# of RAM but beware of setting memory usage too high
loose-set-variable = innodb_buffer_pool_size=50M
loose-set-variable = innodb_additional_mem_pool_size=20M
# Set .._log_file_size to 25 % of buffer pool size
loose-set-variable = innodb_log_file_size=12M
loose-set-variable = innodb_log_buffer_size=4M
loose-innodb_flush_log_at_trx_commit=1
loose-set-variable = innodb_lock_wait_timeout=50

# MySQL Cluster 4.x
relay-log=/mysql-data/logs/slave-bin.log
relay-log-info-file=/mysql-data/logs/slave-info

# changes for cluster
#[mysqld]
ndbcluster
ndb-connectstring=phys-schost-1,phys-schost-2

# provide connectstring for management server host (default port: 1186)
[ndbd]
connect-string=phys-schost-1,phys-schost-2

# provide connectstring for management server host (default port: 1186)
[ndb_mgm]
connect-string=phys-schost-1,phys-schost-2
```

```
# provide location of cluster configuration file
[ndb_mgmd]
config-file=/mgm-data/config.ini
```

`mysql_config` File for the First SQL Node `phys-schost-1` to Store in `/temp/cluconfig`

```
# Where is MySQL Cluster installed (BASEDIR)
MYSQL_BASE=/usr/local/mysql

# MySQL Cluster admin-user for localhost (Default is root)
MYSQL_USER=root

# Password for MySQL Cluster admin user
MYSQL_PASSWD=root

# Configured logicalhost
MYSQL_HOST=phys-schost-1

# Specify a username for a faultmonitor user
FMUSER=fmuser

# Pick a password for that faultmonitor user
FMPASS=

# Socket name for mysqld ( Should be /tmp/logical-host.sock )
MYSQL_SOCK=/tmp/phys-schost-1.sock

# Specify the physical hostname for the
# physical NIC that this logicalhostname belongs to for every node in the
# cluster this resource group is located on.
# IE: The logicalhost lh1 belongs to hme1 for physical-node phys-1 and
# hme3 for physical-node phys-2. The hostname for hme1 is phys-1-hme0 and
# for hme3 on phys-2 it is phys-2-hme3.
# IE: MYSQL_NIC_HOSTNAME="zone1"
MYSQL_NIC_HOSTNAME="phys-schost-1 phys-schost-2"

MYSQL_DATADIR=/mysql-data
# Is MySQL Cluster installed?
# Any entry here triggers the ndb engine check. If no MySQL Cluster cluster should be
 checked
# leave it empty.
NDB_CHECK=y
```

`mysql_config` File for the Second SQL Node `phys-schost-2` to Store in `/temp/cluconfig`

```
# Where is MySQL Cluster installed (BASEDIR)
MYSQL_BASE=/usr/local/mysql

# MySQL Cluster admin-user for localhost (Default is root)
MYSQL_USER=root

# Password for MySQL Cluster admin user
MYSQL_PASSWD=root

# Configured logicalhost
MYSQL_HOST=phys-schost-2

# Specify a username for a faultmonitor user
FMUSER=fmuser

# Pick a password for that faultmonitor user
FMPASS=

# Socket name for mysqld ( Should be /tmp/logical-host.sock )
MYSQL_SOCK=/tmp/phys-schost-2.sock

# Specify the physical hostname for the
# physical NIC that this logicalhostname belongs to for every node in the
# cluster this resource group is located on.
# IE: The logicalhost lh1 belongs to hme1 for physical-node phys-1 and
# hme3 for physical-node phys-2. The hostname for hme1 is phys-1-hme0 and
# for hme3 on phys-2 it is phys-2-hme3.
# IE: MYSQL_NIC_HOSTNAME="zone1"
MYSQL_NIC_HOSTNAME="phys-schost-1 phys-schost-2"

MYSQL_DATADIR=/mysql-data
# Is MySQL Cluster installed?
# Any entry here triggers the ndb engine check. If no MySQL Cluster cluster should be
 checked
# leave it empty.
NDB_CHECK=y
```

mysql_ndb_mgmd_config File for the First Node phys-schost-1

```
# This file will be sourced in by mysql_ndb_mgmd_register and the parameters
# listed below will be used.
#
# These parameters can be customized in (key=value) form
#
#    RS - Name of the resource for the application
#    RG - Name of the resource group containing RS
#
# MySQL Cluster ndb_nmbd specific variables
#
# SCALABLE - Any value here triggers a scalable resource creation
# LB_POLICY - Set the loadbalancing policy for a scalable mysql service.
#       Use the values defined for the standard resource property
#       Load_balancing_policy. If you do not specify it, the defaults are
#       used.
#    LH - Name of the Shared Address SC resource
#  HAS_RS - Name of the MySQL Cluster HAStoragePlus SC resource
#  RS_PROP - Additional resource properties in the format for clresource create,
#     example "-p start_timeout=600"
#  PARFILE - Absolute path to the management server resource parameter file
#
#    The following examples illustrate sample parameters
#    for a multiple-master MySQL Cluster management server resource.
#
#    RS=mgm-rs
#    RG=mgm-rg
#    PORT=1186
#    SCALABLE=
#    HAS_RS=
#    PARFILE=/ndb-mgmd-data/pfile
#

RS=mgm-rs
RG=mgm-rg
PORT=1186
LH=sa_host_1
SCALABLE=yes
LB_POLICY=
HAS_RS=
RS_PROP=
PARFILE=/cluster-pfiles/mgmd-pfile
```

```
#    This is the template for a MySQL Cluster cluster's management server resource.
#    The variables must be specified in the key value form.
#    BASEDIR    Directory where MySQL Cluster cluster is installed, to find the
 binaries.
#    USER      User under which the management server will be run. An empty value
#           stands for the root user.
#    TRY_RECONNECT Value for the try-reconnect parameter in the mgm command.
#    CONNECT_STRING A connect string which is valid for any connection to the management
#           server.
#    CONFIG_DIR   Directory where the management server's config.ini file and its cache
#           files are stored.
#    ID       Unique ID for this management server. This value must match the entry
#           in the config.ini file.
#
#    Examples:
#    BASEDIR=/usr/local/mysql
#    USER=
#    TRY_RECONNECT=1
#    CONNECT_STRING=phys-node-1,phys-node-2
#    CONFIG_DIR=/ndb-nmbd-data
#    ID=1
#

BASEDIR=/usr/local/mysql
USER=
TRY_RECONNECT=1
CONNECT_STRING=phys-schost-1,phys-schost-2
CONFIG_DIR=/mgm-data
ID=1
```

`mysql_ndb_mgmd_config` File for the Second Node `phys-schost-2`

```
# This file will be sourced in by mysql_ndb_mgmd_register and the parameters
# listed below will be used.
#
# These parameters can be customized in (key=value) form
#
#    RS - Name of the resource for the application
#    RG - Name of the resource group containing RS
#
# Mysql cluster ndb_nmbd specific Variables
#
# SCALABLE - Any value here triggers a scalable resource creation
# LB_POLICY - Set the loadbalancing policy for a scalable mysql service.
#       Use the values defined for the standare resource property
#       Load_balancing_policy. If you do not specify it, the defaults are
#       used.
#    LH - Name of the LogicalHostname SC resource
#  HAS_RS - Name of the MySQL Cluster HAStoragePlus SC resource
#  RS_PROP - Additional resource properties in the format for clresource create,
#     example "-p start_timeout=600"
#  PARFILE - Absolute path to the management server resource parameter file
#
#    The following examples illustrate sample parameters
#    for the MySQL Cluster management server resource.
#
#    SCALABLE=yes
#    NETWORK=
#    HAS_RS=
#    RS_PROP=
#    PARFILE=/ndb-mgmd-data/pfile
#

RS=mgm-rs
RG=mgm-rg
PORT=1186
LH=sa_host_1
SCALABLE=yes
LB_POLICY=
HAS_RS=
RS_PROP=
PARFILE=/cluster-pfiles/mgmd-pfile


#    This is the template for a MySQL Cluster cluster's management server resource.
#    The variables must be specified in the key value form.
#    BASEDIR    Directory where MySQL Cluster cluster is installed, to find the
 binaries.
#    USER      User under which the management server will be run. An empty value
#           stands for the root user.
#    TRY_RECONNECT Value for the try-reconnect parameter in the mgm command.
#    CONNECT_STRING A connect string which is valid for any connection to the management
#           server.
#    CONFIG_DIR   Directory where the management server's config.ini file and its cache
#           files are stored.
#    ID       Unique Id for this management server, this value must match the entry
#           in the config.ini file.
#
#    Examples:
#    BASEDIR=/usr/local/mysql/bin
#    USER=
#    TRY_RECONNECT=1
#    CONNECT_STRING=phys-node-1,phys-node-2
#    CONFIG_DIR=/ndb-nmbd-data
#    ID=1
#

BASEDIR=/usr/local/mysql
USER=
TRY_RECONNECT=1
CONNECT_STRING=phys-schost-2,phys-schost-1
CONFIG_DIR=/mgm-data
ID=2
```

`mysql_ndbd_config` File for the First Node `phys-schost-2`

```
# This file will be sourced in by mysql_ndbd_register and the parameters
# listed below will be used.
#
# These parameters can be customized in (key=value) form
#
#    RS - Name of the resource for the application
#    RG - Name of the resource group containing RS
#
# MySQL Cluster cluster ndb_nmbd specific variables
#
#  HAS_RS - Name of the MySQL Cluster HAStoragePlus SC resource
#  RS_PROP - Additional resource properties in the format for clresource create,
#     example "-p start_timeout=600"
#  PARFILE - Absolute path to the data node resource parameter file
#
#    The following examples illustrate sample parameters
#    for the MySQL Cluster data node resource.
#
#    RS=ndbd-rs
#    RG=ndbd-rg
#    HAS_RS=
#    RS_PROP=
#    PARFILE=/ndb-mgmd-data/pfile
#

RS=ndbd-rs
RG=ndbd-rg
HAS_RS=mgm-rs
RS_PROP=
PARFILE=/cluster-pfiles/ndbd-pfile

#    This is the template for a MySQL Cluster cluster's management server resource.
#    The variables must be specified in the key value form.
#    BASEDIR    Directory where MySQL Cluster cluster is installed, to find the
 binaries.
#    USER      User under which the management server will be run. An empty value
#           stands for the root user.
#    TRY_RECONNECT Value for the try-reconnect parameter in the mgm command.
#    CONNECT_STRING A connect string which is valid for any connection to the management
#           server.
#    ID       Unique ID for this management server. This value must match the entry
#           in the config.ini file.
#    MULTI_THREAD  Any entry here will trigger the start of the multithreaded version
 ndbmtd
#           of the ndbd instead of the single-threaded version ndbd.
#    DATA_DIR    Data directory of the ndb process.
#    ERROR_ON_SHOW Return code if the probe is unable to connect to the management
 server.
#
#    Examples:
#    BASEDIR=/usr/local/mysql/bin
#    USER=
#    TRY_RECONNECT=1
#    CONNECT_STRING=phys-node-1,phys-node-2
#    ID=1
#    DATAG_DIR=/ndb-data
#    MULTI_THREAD=yes
#    ERROR_ON_SHOW=25
#

BASEDIR=/usr/local/mysql
USER=
TRY_RECONNECT=1
CONNECT_STRING=phys-schost-1-p,phys-schost-2-p
ID=3
MULTI_THREAD=y
DATA_DIR=/ndbd-data
ERROR_ON_SHOW=25
```

`mysql_ndbd_config` File for the Second Node `phys-schost-2`

```
# This file will be sourced in by mysql_ndbd_register and the parameters
# listed below will be used.
```

```
#
# These parameters can be customized in (key=value) form
#
#    RS - Name of the resource for the application
#    RG - Name of the resource group containing RS
#
# MySQL Cluster cluster ndb_nmbd specific variables
#
#  HAS_RS - Name of the MySQL Cluster HAStoragePlus SC resource
#  RS_PROP - Additional resource properties in the format for clresource create,
#     example "-p start_timeout=600"
#  PARFILE - Absolute path to the data node resource parameter file
#
#    The following examples illustrate sample parameters
#    for the MySQL Cluster data node resource.
#
#    HAS_RS=
#    RS_PROP=
#    PARFILE=/ndb-mgmd-data/pfile
#

RS=ndbd-rs
RG=ndbd-rg
HAS_RS=mgm-rs
RS_PROP=
PARFILE=/cluster-pfiles/ndbd-pfile


#    This is the template for a MySQL Cluster clusters management server resource.
#    The variables must be specified in the key value form.
#    BASEDIR    Directory where MySQL Cluster cluster is installed, to find the
 binaries.
#    USER      User under which the management server will be run, an empty value
#           stands for the root user.
#    TRY_RECONNECT Value for the try-reconnect parameter in the mgm command.
#    CONNECT_STRING A connect string which is valid for any connection to the management
#           server.
#    ID       Unique ID for this management server, this value must match the entry
#           in the config.ini file.
#    MULTI_THREAD  Any entry here will trigger the start of the multithreaded version
 ndbmtd
#           of the ndbd instead of the single-threaded version ndbd.
#    DATA_DIR    Data directory of the ndb process.
#    ERROR_ON_SHOW Return code if the probe is unable to connect to the management
 server.
#
#    Examples:
#    BASEDIR=/usr/local/mysql/bin
#    USER=
#    TRY_RECONNECT=1
#    CONNECT_STRING=phys-node-1,phys-node-2
#    ID=1
#    DATA_DIR=/ndb-data
#    MULTI_THREAD=yes
#    ERROR_ON_SHOW=25
#

BASEDIR=/usr/local/mysql
USER=
TRY_RECONNECT=1
CONNECT_STRING=phys-schost-1-p,phys-schost-2-p
ID=4
MULTI_THREAD=y
DATA_DIR=/ndbd-data
ERROR_ON_SHOW=25
```

`ndbd_shutdown_config` File for One Node

```
# This file will be sourced in by mysql_ndbd_register and the parameters
# listed below will be used.
#
# These parameters can be customized in (key=value) form
#
#    RS - Name of the resource for the application
#    RG - Name of the resource group containing RS
#
# MySQL Cluster cluster ndb_nmbd specific variables
#
#  HAS_RS - Name of the MySQL Cluster HAStoragePlus SC resource
#  RS_PROP - Additional resource properties in the format for clresource create,
#     example "-p start_timeout=600"
#  PARFILE - Absolute path to the data node resource parameter file
#       IMPORATANT: it is essential to specify the data node
#       parameter file.
#
#    The following examples illustrate sample parameters
#    for the MySQL Cluster ndb shutdown resource.
#
#    HAS_RS=ndbd-rs
#    RS_PROP=
#    PARFILE=/ndb-mgmd-data/pfile
#

RS=ndbd-shut-rs
```

```
RG=ndbd-rg
HAS_RS=ndbd-rs
RS_PROP=
PARFILE=/cluster-pfiles/ndbd-pfile
```

`ha_mysql_config` File for One Node

```
# This file will be sourced in by ha_mysql_register and the parameters
# listed below will be used.
#
# These parameters can be customized in (key=value) form
#
#    RS - Name of the resource for the application
#    RG - Name of the resource group containing RS
# SCALABLE - Flag to indicate a scalable resource creation.
#       The default is no, so any entry here triggers a scalable resource.
# LB_POLICY - Set the loadbalancing policy for a scalable MySQL Cluster service.
#       Use the values defined for the standare resource property
#       Load_balancing_policy. If you do not specify it, the defaults are
#       used.
#  RS_PROP - Additional resource properties in the format for clresource create,
#     example "-p start_timeout=600"
#  PROJECT - A project in the zone that will be used for this service
#       Specify it if you have an su - in the start stop or probe,
#       or to define the smf credentials. If the variable is not set,
#       it will be translated as :default for the sm and default
#       for the zsh component
#       Optional
#   ZUSER - A user in the the zone which is used for the smf method
#       credentials. Your smf service will run under this user
#       Optional
#
#  BASEDIR - Name of the MySQL Cluster bin directory
#  DATADIR - Name of the MySQL Cluster Data directory
# MYSQLUSER - Name of the user Mysql should be started of
#    LH - Name of the LogicalHostname SC resource
# MYSQLHOST - Name of the host in /etc/hosts
#  FMUSER - Name of the MySQL Cluster fault monitor user
#  FMPASS - Name of the MySQL Cluster fault monitor user password
#  LOGDIR - Name of the directory mysqld should store it's logfile.
#  CHECK - Should HA-MySQL Cluster check MyISAM index files before start YES/NO.
#  HAS_RS - Name of the MySQL Cluster HAStoragePlus SC resource
#
#    The following examples illustrate sample parameters
#    for MySQL Cluster
#
```

```
#    BASEDIR=/usr/local/mysql
#    DATADIR=/global/mysqldata
#    MYSQLUSER=mysql
#    LH=mysqllh
#    MYSQLHOST=mysqllh
#    FMUSER=fmuser
#    FMPASS=
#    LOGDIR=/global/mysqldata/logs
#    CHECK=YES
#

RS=mys-rs
RG=mysql-rg
PORT=3306
SCALABLE=yes
LB_POLICY=
LH=sa_host_1
HAS_RS=ndbd-shut-rs
RS_PROP=
# Local zone specific options

ZONE=
ZONE_BT=
ZUSER=
PROJECT=

# MySQL Cluster specifications

BASEDIR=/usr/local/mysql
DATADIR=/mysql-data
MYSQLUSER=mysql
MYSQLHOST=
FMUSER=fmuser
FMPASS=
LOGDIR=/mysql-data/logs
CHECK=YES
NDB_CHECK=YES
```

Index