Sun Directory MMR

Administration Guide

West Area IT Infrastructure

Version 1.0
4/30/2009

Revision History

Document version	Date	Change description	Modified by
1.0	04/30/2009	Initial version for review	Fawzia Begum

Table of Contents

Directory Server Replication

This document describes the Directory Server Replication and the tasks to be performed to set up various replication scenarios by using the command line. Replication is the mechanism by which directory contents are automatically copied from a Directory Server to one or more other Directory Servers. All write operations are automatically mirrored to other Directory Servers. In a replication topology, generally one suffix on a server is replicated to or from another suffix on a server. For this reason, the terms replica, replicated suffix and replicated server can be used interchangeably.

For a complete description of replication concepts, replication scenarios, and how to plan for replication in your directory deployment, see the Sun Java System Directory Server Enterprise Edition 6.2 Deployment Planning Guide.

The document covers the following topics:

- Planning Your Replication Deployment
- Recommended Interface for Configuring and Managing Replication
- Summary of Steps for Configuring Replication
- Enabling Replication on a Dedicated Consumer
- Enabling Replication on a Hub
- Enabling Replication on a Master Replica
- Configuring the Replication Manager
- Creating and Changing Replication Agreements
- Fractional Replication
- Replication Priority
- Initializing Replicas
- Indexing Replicated Suffixes
- Incrementally Adding Many Entries to Large Replicated Suffixes
- Maintaining Referential Integrity
- Replication Over SSL
- Replication Over a WAN
- Modifying the Replication Topology
- Replication with Releases Prior to Directory Server 6.2
- Using the Retro Change Log
- Getting Replication Status
- Solving Common Replication Conflicts

Planning Your Replication Deployment

You can configure a replication deployment with an unlimited number of masters. You are not required to include hubs or consumers in your deployment. Procedures for configuring replication for hubs and consumers are included in this document, but they are optional.

Before you begin configuring replication, you need to have a clear understanding of the way that replication will be deployed in your organization. You must understand the replication concepts described in *Sun Java System Directory Server Enterprise Edition 6.2 Reference*. You must also have carefully planned your future replication configuration using the design guidelines provided in the *Sun Java System Directory Server Enterprise Edition 6.2 Deployment Planning Guide*.

Recommended Interface for Configuring and Managing Replication

The easiest way to configure and manage replication is by using Directory Service Control Center (DSCC). By using DSCC, you can configure replication automatically. You can choose the level of automation you require for setting up your replication topology, for example, whether you want to initialize the suffixes during replication configuration or not. DSCC also provides checks that can prevent errors. In addition, DSCC provides a graphical view of the replication topology.

The DSCC online help provides procedures for setting up replication by using DSCC.

Summary of Steps for Configuring Replication

"Summary of Steps for Configuring Replication" assumes that you are replicating a single suffix. If you are replicating more than one suffix, you can configure the suffixes in parallel on each server. In other words, you can repeat each step to configure replication on multiple suffixes.

The rest of this section contains detailed instructions on how to configure replication.

Summary of Steps for Configuring Replication

To configure any replication topology, follow the general steps as outline in this procedure.

1. **Do the following on all servers that contain a dedicated consumer replica:**

    a. Create an empty suffix for the consumer replicated suffix.
    b. Enable the consumer replicated suffix.
    c. (Optional) Configure the advanced consumer settings.

2. **If applicable, do the following on all servers that contain a hub replicated suffix:**

    a. Create an empty suffix for the hub replicated suffix.
    b. Enable the hub replicated suffix.
    c. (Optional) Configure the advanced hub settings.

3. **Do the following on all servers that contain a master replicated suffix:**

    a. Create a suffix for the master replicated suffix.
    b. Enable the master replicated suffix.
    c. (Optional) Configure the advanced master settings.

Note – Make sure that you enable all replicas before you create a replication agreement so that you can initialize consumer replicas immediately after you create the replication agreement. Consumer initialization is always the last stage in setting up replication.

4. **Ensure your replication manager configuration is complete.**

    - If you plan to use the default manager, set the default replication manager password on all servers. See "To Change the Default Replication Manager Password."

- If you plan to use a non-default replication manager, define the alternative replication manager entry on all servers. See “Using a Non-Default Replication Manager.”

5. **Create replication agreements on all master replicas as follows:**

    a. Between masters in a multimaster topology
    b. Between masters and their dedicated consumers
    c. Between masters and hub replicas

6. **(Optional) If you want to use fractional replication, configure it now.**
7. **(Optional) If you want to use replication priority, configure it now.**
8. **Configure replication agreements between the hub replicas and their consumers.**
9. **For multimaster replication, initialize all masters from the same master replica that contains the original copy of the data.**
10. **Initialize the hub and consumer replicas.**

Enabling Replication on a Dedicated Consumer

A dedicated consumer is a read-only copy of a replicated suffix. The dedicated consumer receives updates from servers that bind as the replication manager to make changes. Configuring the consumer server consists of preparing an empty suffix to hold the replicated suffix and enabling replication on that suffix. Optional advanced configuration can include setting referrals, changing the purge delay, and modifying properties.

The following sections explain how to configure one dedicated consumer replicated suffix on its server. Repeat all procedures on each server that will contain a dedicated consumer replicated suffix.

To Create a Suffix for a Consumer Replica

- **If an empty suffix does not already exist, create it on the consumer with the same DN as the intended master replica.**

For instructions, see “Creating Suffixes.”

Caution – If the suffix exists and is not empty, its contents will be lost when the replicated suffix is initialized from the master.

To Enable a Consumer Replica

After you have created an empty suffix, you need to enable the consumer replicated suffix.
You can use DSCC to perform this task.

- **Enable the consumer replicated suffix.**

    ```
    $ dsconf enable-repl -h host -p port consumer suffix-DN
    ```

    For example:

    ```
    $ dsconf enable-repl -h host1 -p 1389 consumer dc=example,dc=com
    ```

To Perform Advanced Consumer Configuration

If you want to configure your consumer replicated suffix for advanced features, do so now. You can use DSCC to perform this task.

1. **If you want to use SSL for referrals, set secure referrals.**

   ```
   $ dsconf set-suffix-prop -h host -p port suffix-DN referral-
   url:ldaps://servername:port
   ```

   For example:

   ```
   $ dsconf set-suffix-prop -h host1 -p 1389 dc=example, dc=com \
   referral-url:ldaps://server2:2389
   ```

The replication mechanism automatically configures consumers to return referrals for all known masters in the replication topology. These default referrals assume that clients will use simple authentication over a regular connection. If you want to give clients the option of binding to masters using SSL for a secure connection, add referrals of the form *ldaps://servername :port* that use a secure port number. Note that if the masters are configured for secure connections only, the URLs will point to the secure ports by default.

If you have added one or more LDAP URLs as referrals, you can force the consumer to send referrals exclusively for these LDAP URLs and not for the master replicas. For example, suppose that you want clients to always be referred to the secure port on the master servers and not to the default port. Create a list of LDAP URLs for these secure ports, and set the property for using these referrals. You can also use an exclusive referral if you want to designate a specific master or a Directory Server proxy to handle all updates.

2. **If you want to change the replication purge delay for the consumer, use this command:**

   ```
   $ dsconf set-suffix-prop -h host -p port suffix-DN repl-purge-delay:time
   ```

   For example, to set the purge delay to 2 days, type:

   ```
   $ dsconf set-suffix-prop -h host1 -p 1389 edc=example,dc=com repl-purge-
   delay:2d
   ```

The consumer server stores internal information about updates to the replicated suffix contents, and the purge delay parameter specifies how long it must keep this information. The purge delay determines in part how long replication between the consumer and its master can be interrupted, and still recover normally. It is related to the MaxAge parameter of the change log on its supplier server. The shorter of these two parameters determines the longest time that replication between the two servers can be disabled or down and still recover normally. The default value of 7 days is sufficient in most cases.

Enabling Replication on a Hub

Hub replicas act as both consumers and masters to further distribute replicated data to a larger number of consumers. Hub replicas receive replication updates from their suppliers and send replication updates to their consumers. They do not accept modifications, but instead return referrals to the masters.

Configuring a hub server consists of preparing an empty suffix to hold the replicated suffix and enabling replication on that suffix. Optional advanced configuration can include choosing a different replication manager, setting referrals, setting the purge delay, and modifying change log parameters.

The following sections explain how to configure one hub server. Repeat all procedures on each server that will contain a hub replicated suffix.

To Create a Suffix for a Hub Replica

- **If an empty suffix does not already exist, create it on the hub server with the same DN as the intended master replica.**

  For instructions, see “Creating Suffixes.”

  If the suffix exists and is not empty, its contents will be lost when the replicated suffix is initialized from the master.

To Enable a Hub Replica

If you have hub replicas, enable them now. You can use DSCC to perform this task.

- **Enable the hub replicated suffix.**

  ```
  $ dsconf enable-repl -h host -p port hub suffix-DN
  ```

  For example:

  ```
  $ dsconf enable-repl -h host1 -p 1389 hub dc=example,dc=com
  ```

To Modify Change Log Settings on a Hub Replica

For advanced hub configuration, the only parameters that you might want to modify are related to the change log. As a supplier, a hub server requires a change log.

To modify a change log setting on a hub, use one of the following commands:

```
$ dsconf set-server-prop -h host -p port suffix-DN repl-cl-max-age:value
$ dsconf set-server-prop -h host -p port suffix-DN repl-cl-max-entry-count:value
```

Enabling Replication on a Master Replica

Master replicas contain the master copy of the data and centralize all modifications before propagating updates to all other replicas. A master records all changes, checks the status of its consumers and sends updates to them when necessary. In multimaster replication, master replicas also receive updates from other masters.

Configuring a master server consists of defining the suffix that contains the master replica, enabling the master replica, and configuring it for advanced replication, if necessary.

The following sections explain how to configure one master server. Repeat all procedures on each server that will contain a master replicated suffix.

To Create a Suffix for a Master Replica

- **Choose or create a suffix on the master server that will contain the entries that you want to replicate.**

  For instructions, see to “Creating Suffixes.”

To ensure correct multimaster configuration and initialization, only load one of the masters with the data. Any data on other replicated suffixes will be overwritten.

To Enable a Master Replica

When you enable replication on a master, you must assign a replication ID. The replication ID is used to distinguish the owner of update statements and to resolve conflicts that might occur with multimaster replication. Therefore, the replication ID must be unique for all master replicas of this suffix. Once set, the replication ID must not be changed. You can use DSCC to perform this task.

- **Enable the master replicated suffix.**

  ```
  $ dsconf enable-repl -h host -p port -d ReplicaID master suffix-DN
  ```

  where *ReplicaID* is an integer between 1 and 65534.

  For example, to create a master replicated suffix with replica ID 1, use this command:

  ```
  $ dsconf enable-repl -h host1 -p 1389 -d 1 master dc=example, dc=com
  ```

To Modify Change Log Settings on a Master Replica

For advanced master configuration, you might want to modify the change log settings.

You can use DSCC to perform this task.

- **If you want to modify a change log setting on a master, use one of the following commands**:

  ```
  $ dsconf set-server-prop -h host -p port suffix-DN repl-cl-max-age:value
  $ dsconf set-server-prop -h host -p port suffix-DN repl-cl-max-entry-count:value
  ```

Configuring the Replication Manager

This section describes how to configure a non-default replication manager and how to set the default replication manager password.

Using a Non-Default Replication Manager

The replication manager is the user that suppliers will use to bind to a consumer server when sending replication updates. All servers that contain suffixes receiving updates must have at least one replication manager entry.

Directory Server has a default replication manager entry that you can use on every server, especially for simple replication scenarios: *cn=replication manager,cn=replication,cn=config*. The replication mechanism automatically configures consumer replicas with this user, simplifying the deployment of replicas.

If you have a more complex replication scenario, you might want several replication managers with a different password for each replicated suffix. You can replace the existing default replication manager with one or more new replication managers.

Caution – Never bind or perform operations on the server using the DN and password of the replication manager. The replication manager is for use only by the replication mechanism.

Any other use might require reinitializing the replicas.

Never use the Directory Manager as the replication manager. Because the *cn=admin,cn=Administrators,cn=config* entry is used for other administrative tasks, you must also not use this user or any other user in the administrator group as the replication manager.

After you have chosen the replication manager for each consumer, ensure that you remember the replication manager DN that you chose or created. You will need this DN and its password later when creating the replication agreement with this consumer on its supplier.

To Set A Non-Default Replication Manager

You can use DSCC to perform this task.

1. **On all consumer (destination) replicated suffixes, create a new replication manager and password.**

   ```
   $ ldapmodify -a -h host -p port -D cn=admin,cn=Administrators,cn=config -w -

   Enter bind password:
   dn:"cn=new-replication-manager,cn=replication,cn=config"
   objectclass: top
   objectclass: person
   userpassword:password
   sn:new-replication-manager
   ```

   For example:

   ```
   $ ldapmodify -a -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w
   -
   Enter bind password:
   dn:"cn=ReplicationManager3,cn=replication,cn=config"
   objectclass: top
   objectclass: person
   userpassword:secret
   sn:ReplicationManager3
   ```

2. **On all consumer (destination) replicated suffixes, set the replication manager bind DN.**

   ```
   $ dsconf set-suffix-prop -h host -p port suffix-DN \
   repl-manager-bind-dn:"cn=new-replication-manager,cn=replication,cn=config"
   ```

   For example:

   ```
   $ dsconf set-suffix-prop -h host1 -p 1389 dc=example,dc=com \
   repl-manager-bind-dn:"cn=ReplicationManager3,cn=replication,cn=config"
   ```

3. **For all replication agreements that you have created on all supplier (source) replicated suffixes, set the replication manager bind DN.**

   a. **Create a temporary file for setting the new replication manager password. This file is read once, and the password is stored for future use.**

      ```
      $ echo password > password-file
      ```

b. **Set the replication manager bindDN and password to be used by the replication mechanism when performing updates.**

```
$ dsconf set-repl-agmt-prop -h host -p port suffix-DN host:port \
auth-bind-dn:"cn=new-replication-manager,cn=replication,cn=config" \
auth-pwd-file:password-file
```

For example:

```
$ dsconf set-repl-agmt-prop -h host2 -p 1389 dc=example,dc=com
host1:1389 \
auth-bind-dn:"cn=ReplicationManager3,cn=replication,cn=config" \
auth-pwd-file:pwd.txt
```

c. **Remove the temporary password file.**

```
$ rm password-file
```

To Change the Default Replication Manager Password

1. **Create a temporary file for setting the replication manager password.**
   This file is read once, and the password is stored for future use.

   ```
   $ echo password > password-file
   ```

2. **On all consumer (destination) servers in the replication topology, set the replication manager bind password.**

   ```
   $ dsconf set-server-prop -h host -p port def-repl-manager-pwd-file:password-
   file
   ```

   For example:

   ```
   $ dsconf set-server-prop -h host1 -p 1389 def-repl-manager-pwd-file:pwd.txt
   ```

3. **Remove the temporary password file.**

   ```
   $ rm password-file
   ```

Creating and Changing Replication Agreements

A *replication agreement* is a set of parameters on a supplier that configures and controls how updates are sent to a given consumer. The replication agreement must be created on the supplier replicated suffix that is sending updates to its consumer. You must create a replication agreement on the supplier for every consumer that you want updated.

To Create a Replication Agreement

You can use DSCC to perform this task. If you use DSCC to create a new replication agreement, you can choose to copy some or all replication agreement configuration settings from an existing replication agreement.

1. **From your master server, create a replication agreement for each consumer that you want to replicate to.**

```
$ dsconf create-repl-agmt -h host -p port suffix-DN consumer-host:consumer-port [consumer-host:consumer-port]
```

For example:

```
$ dsconf create-repl-agmt -h host1 -p 1389 dc=example,dc=com host2:1389
```

To list existing replication agreements by using the command line, use the `dsconf list-repl-agmts` command.

Note – If you change the port number on a master when replication is running, you do not need to reinitialize the servers. However, the old replication agreement that pointed to the old address (*host:old-port*) is no longer useful. If you want replication to continue as it did before the port number was changed, you must create a new agreement with the new address (*host:new-port*).

2. **Check that the replication agreement has been created correctly.**

```
$ dsconf show-repl-agmt-status -h host -p port suffix-DN consumer-host:consumer-port
```

If the authentication status is not OK, run the `dsconf accord-repl-agmt` command.

Note – Only use the command `dsconf accord-repl-agmt` if you are using the default replication manager. If you have created a new replication manager, do not use this command because it overwrites some required settings.

The `dsconf accord-repl-agmt` command ensures that both the supplier and destination servers share the same replication authentication settings.

```
$ dsconf accord-repl-agmt -h host -p port suffix-DN consumer-host:consumer-port
```

For example:

```
$ dsconf accord-repl-agmt -h host2 -p 1389 dc=example,dc=com host1:1389
```

To Change the Destination of a Replication Agreement

This procedure changes the remote replica pointed to by an existing replication agreement. The Suffix DN and configuration of the existing agreement remain the same.

- **Change the host name and port number of the remote replica in the replication agreement.**

  ```
  $ dsconf change-repl-dest -h host -p port suffix-DN host:port new-host:new-port
  ```

  If this command is run with the -A protocol option, you can change the authentication protocol that is used by replication.

Fractional Replication

By default, the replication operation copies entire entries in the replicated suffix to consumer replicas. With the fractional replication feature, you can select the suffix that you want to use, and which attributes you want to include or exclude. Fractional replication is configured in the replication agreement, allowing you to define the attribute set for each consumer replicated suffix of a master. You can control which data is distributed and use replication bandwidth and consumer resources more efficiently.

For example, if you want to reduce replication bandwidth, you can choose not to replicate attributes with typically large values such as photo, jpegPhoto, and audio. As a result, these attributes will not be available on consumers. As another example, you can choose to replicate only the uid and userpassword attributes to a consumer server that is dedicated to performing authentication.

Considerations for Fractional Replication

Note – Fractional replication cannot be used in versions of the product prior to Directory Server 5.2. When configuring a fractional replication agreement, both the master and consumer replicas must use at least Directory Server 5.2.

Enabling or modifying a fractional set of attributes requires you to reinitialize the consumer replica. Therefore, you need to determine your fractional replication needs before deployment and define your attribute set before you initialize your replicated suffixes for the first time.

You need to proceed with caution when replicating a small set of attributes, given the dependency of complex features such as ACIs, roles, and CoS on certain attributes. In addition, not replicating other attributes that are mentioned in specifiers or filters of the ACI, roles, or CoS mechanisms might compromise the security of the data. Not replicating might also result in different sets of attributes being returned in searches. Managing a list of attributes to exclude is safer, and less prone to human error, than managing a list of attributes to include.

You need to turn off schema checking in the consumer server if the attribute set that you replicate does not allow all replicated entries to follow the schema. Replication of non-conforming entries does not cause errors because the replication mechanism bypasses schema checking on the consumer. However, the consumer will contain non-conforming entries and should have schema checking turned off to expose a coherent state to its clients.

Fractional replication is configured in the replication agreement of master replicas with hubs and dedicated consumers. Configuration of fractional replication between two master replicas in a multimaster replication environment is not supported. Also, if several masters have replication agreements with the same replica, all these agreements must replicate the same set of attributes.

To Configure Fractional Replication

To configure fractional replication, you must specify the suffix, determine whether to include or exclude attributes on that suffix, then choose which attributes to include or exclude. If you choose to exclude attributes on a suffix, all other attributes are automatically included. Likewise, if you choose to include certain attributes on a suffix, all other attributes are automatically excluded.

- **Configure fractional replication on a replication agreement located on the source server.**

  ```
  $ dsconf set-repl-agmt-prop -h host -p port suffix-DN consumer-host:consumer-port property:value
  ```

  where property is either `repl-fractional-exclude-attr` or `repl-fractional-include-attr`.

  For example, if you want to configure a fractional agreement to exclude JPEG and TIFF photos from being replicated on the suffix dc=example,dc=com, use this command:

  ```
  $ dsconf set-repl-agmt-prop -h host2 -p 1389 dc=example,dc=com host1:1389
  repl-fractional-exclude-attr:jpegPhoto repl-fractional-exclude-attr:tiffPhoto
  ```

  To add an attribute to an existing list of attributes that should be excluded, use this command:

```
$ dsconf set-repl-agmt-prop -h host -p port suffix-DN consumer-host:consumer-port repl-fractional-exclude-attr+:attribute
```

Replication Priority

Specifying replication priority is optional. You can create replication rules to specify that certain changes, such as updating the user password, are replicated with high priority. Any changes specified in replication rules are replicated as high priority, and all other changes are replicated with normal priority.

Note – Replication priority rules only need to be created on the master server. No configuration is required for hubs and consumers.

To Configure Replication Priority

You can use DSCC to perform this task.

To create a new replication priority rule on a master, use this command:

```
$ dsconf create-repl-priority -h host -p port suffix-DN priority-name property:value
```

You can set replication priority with one or more of the following properties:

- Operation type, op-type
- BindDN, bind-dn
- BaseDN, base-dn
- Attribute type, attr

The *priority-name* is user defined.

For example, to create a replication rule specifying that user password changes are replicated with high priority, use this command:

```
$ dsconf create-repl-priority -h host2 -p 1389 dc=example,dc=com pw-rule \
attr:userPassword
```

To display current replication rules, use the dsconf list-repl-priorities -v command. When used with the -v option, this command displays additional information related to prioritized replication rules.

```
$ dsconf list-repl-priorities -h host2 -p 1389 -v
```

Initializing Replicas

After you have created a replication agreement and after both replicas have been configured, you must initialize the consumer replicated suffix before replication can begin. During initialization, you physically copy data from the supplier replicated suffix to the consumer replicated suffix.

In addition, certain error conditions or configuration changes require you to reinitialize replicas. For example, if the data in a single master replicated suffix is restored from a backup for any reason, you need to reinitialize all of the replicas that it updates.

When reinitializing, the contents of the replicated suffix are deleted on the consumer and replaced with the contents of the suffix on the master. This ensures that the replicas are synchronized and that replication updates can resume. All of the initialization methods described in this section automatically

rebuild the indexes of the consumer replica so that the consumer is ready to respond optimally to client read requests.

With multimaster replication, consumers might not need to be reinitialized if they have been updated by the other masters in the topology.

To Initialize a Replicated Suffix from a Remote (Supplier) Server

You can initialize a suffix from a remote server by using an existing replication agreement. Use this method of initializing if possible, because it is less complicated than the other methods. Use other methods only for large quantities of data that make the import too time consuming.

You can use DSCC to perform this task.

1. **Initialize your replica.**

   ```
   $ dsconf init-repl-dest -h host -p port suffix-DN destination-
   host:destination-port [destination-host:destination-port]
   ```

   where destination-host:destination-port is the host and port of the destination server that you are initializing from the remote server.

2. **(Optional) For each agreement, check that the suffix appears as initialized.**

   ```
   $ dsconf show-repl-agmt-status -h host -p port suffix-DN destination-
   host:destination-port
   ```

Replica Initialization from LDIF

To Initialize a Replicated Suffix from LDIF

This procedure outlines the general steps to use to initialize a replicated suffix from an LDIF file.

You can use DSCC to perform this task.

1. **Ensure that you have set up replication agreements.**

   You must do this before you initialize replicas.

2. **Export the original copy of the suffix data from a master replicated suffix to an LDIF file.**
   See “To Export a Replicated Suffix to LDIF.”

In a multimaster replication environment, you can use the LDIF file exported from the original master to initialize both the other masters and any consumers. In a cascading replication environment, you can use the same file to initialize both the hub replicas and their consumers.

In all cases, you must start with an LDIF file that has been exported from a configured master replica. You cannot use an arbitrary LDIF file to initialize all replicas because it does not contain replication meta-data.

3. **If you are initializing a fractional replica, filter the file to keep only the replicated attributes, then transfer that file to all of the consumer servers.**

See “Filtering an LDIF File for Fractional Replication.”

4. **Initialize your replica.**
   Do one of the following:

   - For fast initialization on a server that is offline (stopped), use the dsadm import command.

     ```
     $ dsadm import instance-path LDIF_file suffix-DN
     ```

   - To initialize a replica online from an LDIF file, use the dsconf import command.

     ```
     $ dsconf import -h host -p port LDIF_file suffix-DN
     ```

     Using dsconf import is slower than using dsadm import, but you do not need to stop your server while performing the import operation.

   For more detailed information about initializing suffixes, and for examples, see “Initializing a Suffix.” For detailed command usage, see dsadm(1M) and dsconf(1M).

5. **(Optional) For each agreement, check that the suffix appears as initialized.**

```
$ dsconf show-repl-agmt-status -h host -p port suffix-DN destination-host:destination-port
```

To Export a Replicated Suffix to LDIF

You can use DSCC to perform this task.

- Export the replicated suffix contents in an LDIF file by using one of the following commands:

  - For an offline export, type:

    ```
    $ dsadm export instance-path suffix-DN LDIF_file
    ```

  - For an online export, type:

    ```
    $ dsconf export -h host -p port suffix-DN LDIF_file
    ```

The following example will export the *entire dc=example,dc=com* replicated suffix and replication information to the file *example_replica_export.ldif*:

```
$ dsconf export -h host2 -p 1389 dc=example,dc=com \
/local/ds/ldif/example_export_replica.ldif
```

Filtering an LDIF File for Fractional Replication

Initializing a replica with fractional replication configured is transparent when using DSCC. Only the selected attributes will be sent to the consumer during the initialization.

If you have configured fractional replication, you should filter out any unused attributes before copying the exported LDIF file to the consumer servers. Directory Server provides the fildif tool for this purpose. This tool filters the given LDIF file to keep only the attributes that are allowed by the attribute set defined in your replication agreement.

This tool reads the server's configuration to determine the attribute set definition. To read the configuration file, the fildif tool must be run as root or as the user who owns the process and the files (specified by the `nsslapd-localuser` attribute). For example, the following command filters the file exported from the dc=example,dc=com suffix in the previous example:

```
$ fildif -i /local/ds1/ldif/example_master.ldif \
-o /local/ds1/ldif/filtered.ldif -b "cn=host2.example.com:1389, \
cn=replica,cn=\\"dc=example,dc=com\\",cn=mapping tree,cn=config" -p /local/ds1
```

The -i and -o options are the input and output files, respectively. The -b option is the DN of the replication agreement where fractional replication is defined. You can find this DN by using this command:

```
$ ldapsearch -h host -p port -D cn=admin,cn=Administrators,cn=config -w - \
-b              "cn=config"              "(&(objectclass=nsds5replicationagreement)
(nsDS5ReplicaPort=replica-port) \
(nsDS5ReplicaHost=replica-host))" dn
```

For example:

```
$ ldapsearch -h host2 -p 1389 -D cn=admin,cn=Administrators,cn=config -w - \
-b "cn=config" "(&(objectclass=nsds5replicationagreement) \
(nsDS5ReplicaPort=2090)(nsDS5ReplicaHost=host2))" dn
Enter bind password:
version: 1
dn: cn=host2:1389,cn=replica,cn=dc\=example\,dc\=com,cn=mapping tree,cn=config
```

For the full command-line syntax for the fildif tool, see the fildif(1) man page.

You can then use the filtered.ldif file produced by fildif to initialize the consumer in this replication agreement. Transfer the file to the consumer server and import it.

Initializing a Replicated Suffix by Using Binary Copy

A binary copy enables you to clone an entire server by using the binary backup files from one server to restore the identical directory contents onto another server. You can use a binary copy to initialize or reinitialize any server from the binary copy of a master or hub server, or a consumer from the binary copy of another consumer server.

Note – This advanced procedure interacts with the database files of Directory Server and should only be used by experienced administrators.

Certain restrictions on this feature make it practical and time efficient only for replicas with very large database files, for example, replicas containing millions of entries.

Restrictions for Using Binary Copy with Replication

Because a binary copy moves database files from one machine to another, the mechanism is subject to the following strict limitations:

- Both machines must run the same operating system, including any service packs or patches.
- Both machines must share the same processor architecture. For example, you can perform binary copy between two UltraSPARC® T1 processors but not between an UltraSPARC T1 and an AMD Opteron processor.
- Both machines must be either big endian or little endian.

- Both machines must map memory the same way. For example, you can perform binary copy between server instances on two 64-bit systems, but not between one server instance on a 32-bit system and another on a 64-bit system.
- Both machines must have the same version of Directory Server installed, including binary format (32 bits or 64 bits), service pack, and patch level.
- Both servers must have the same directory tree divided into the same suffixes. The database files for all suffixes must be copied together. Individual suffixes cannot be copied.
- Each suffix must have the same indexes configured on both servers, including VLV (virtual list view) indexes. The databases for the suffixes must have the same name.
- Each server must have the same suffixes configured as replicas.
- If fractional replication is configured, it must be configured identically on all servers.
- Attribute encryption must not be used on either server.
- The attribute value uniqueness plug-in must have the same configuration on both servers if enabled, and it must be re-configured on the new copy, as explained in the following procedures.

  These procedures describe alternate ways of performing a binary copy: a binary copy that does not require stopping the server and a binary copy that uses the minimum amount of disk space.

Making a Binary Copy for Initializing a Server

This section describes how to make a binary copy for initializing a server, and how to make a binary copy that uses minimum disk space.

To Make a Binary Copy for Initializing a Server

Use this procedure to perform a binary copy for initializing a replicated server because it uses the normal backup functionality to create a copy of the server's database files. Performing a normal backup ensures that all database files are in a coherent state without requiring you to stop the server.

This procedure has certain limitations. The backup and restore operations create copies of the database files on the same machine, thereby doubling the amount of disk space required by those files on each machine. Additionally, the actual copy operation on these files might take a significant amount time if your directory contains gigabytes of data.

For parts of this procedure, you can use DSCC to perform this task. Other parts of the procedure can only be done using the command line.

1. **Install Directory Server on the target machine for the new replicated suffix, create a new instance of the server if necessary, and configure the server according to "Restrictions for Using Binary Copy With Replication."**

2. **Create all replication agreements in your replication topology that involve this replicated suffix.**

   Include agreements from suppliers to this replica. If this replica is not a dedicated consumer, include agreements from this replica to its consumers.

3. **Select a fully configured and initialized replica of the same type as you want to initialize, either master, hub, or consumer, and perform a normal backup on it according to "Binary Backup."**

4. **Copy or transfer the files from the backup directory to a directory on the target machine by**

using the ftp command, for example.

5. **If you have initialized a new master in a multimaster replication scenario, follow the procedures in "Restoring a Master in a Multi-Master Scenario."**

To Use Binary Copy for Initializing a Server Using Minimum Disk Space

This procedure uses less disk space and takes less time because it does not make backup copies of the database files. However, it requires you to stop the server that is being cloned to order to ensure that the database files are in a coherent state.

Caution – This procedure must not be used to reinitialize a master that has already participated in a multimaster replication scenario. It can only be used to reinitialize a consumer server or to initialize a new master server. To reinitialize an existing master replica, use online initialization, import an LDIF file, or follow the procedure in "Making a Binary Copy for Initializing a Server."

For parts of this procedure, you can use DSCC to perform this task. Other parts of the procedure can only be done using the command line.

1. **Install Directory Server on the target machine for the new replicated suffix, create a new instance of the server if necessary, and configure the server according to "Restrictions for Using Binary Copy with Replication."**

2. **Create all replication agreements in your replication topology that involve this replica.**

   Include agreements from suppliers to this replica. If this replica is not a dedicated consumer, include agreements from this replica to its consumers. See "Creating and Changing Replication Agreements."

3. **Stop the target server that will be initialized or reinitialized, as described in "Starting, Stopping, and Restarting a Directory Server Instance."**

4. **Select a fully configured and initialized replica of the same type that you want to initialize, either master or hub or consumer, and stop this server as well.**

   If you are cloning a master replica in a multimaster configuration, ensure that it is fully up-to-date with all of the latest changes from the other masters before stopping it.

5. **Remove all database files from the target server, including transaction logs, change logs, and region files (__db.xxx files).**

   Unless the files have been relocated, database files and transaction logs are located in the instance-path/db directory.

6. **Copy or transfer all database files, including transaction logs and change logs, from the source replica machine to the target machine, by using the ftp command, for example.**

   Unless the files have been relocated, database files and transaction logs are located in the instance-path/db directory.

   If you are initializing a master or hub replica, you must also copy all files in the change log, which is located in instance-path/changelog by default.

7. **Restart both the source and the target servers.**

Initializing Replicas in Cascading Replication

In the case of cascading replication, always initialize replicas in the order shown in the following procedure.

To Initialize Replicas in Cascading Replication

You can use DSCC to perform this task.

1. **If you also have multimaster replication, ensure that one master has the complete set of data to replicate, use this master to initialize the replica on each of the other masters.**

2. **Initialize the replicas on the first-level hub replicas from their master replicas.**

3. **If you have several levels of hubs, initialize each level from the previously initialized level of hubs.**

4. **From the last level of hub replicas, initialize the replicas on the dedicated consumers.**

Indexing Replicated Suffixes

Indexes are not replicated automatically from one server instance to another. To index an attribute for all server instances holding a replicated suffix, perform one of the following actions.

- Manage all server instances holding the replicated suffix as a server group in DSCC. Add the index to one server in the group, then use the Copy Server Configuration action to copy index settings to other servers in the group.

- Manage the index on each server instance with the dsconf command.

- Use binary copy to initialize suffixes, as described in “Initializing a Replicated Suffix by Using Binary Copy.”

Incrementally Adding Many Entries to Large Replicated Suffixes

If you have a directory with a very large number of entries and you want to add a large quantity of entries, do not use ldapmodify -a because it is too time consuming. Instead, add the new entries incrementally by using the dsconf import command with an option for adding entries in a replicated topology. When you import the entries, an LDIF file is generated that contains the additions as well as replication meta-data. You then import this generated LDIF file to the other replicas. The generated LDIF file ensures that replication synchronization is constant across the replicas to which you add data.

To Add Many Entries to Large Replicated Suffixes

This procedure generates a large LDIF file. Before running the first dsconf import command, ensure that you have enough disk space available for the generated LDIF file.

You cannot use DSCC to perform this task. Use the command line, as described in this procedure.

Caution – This procedure can be used to initialize a server with a large number of entries in several passes. However, if one of the imports fails, the whole database can be lost. Be sure to backup data prior to each import.

1. **On any master replica, import the entries.**

   ```
   $ dsconf import -h host -p port -K generated-LDIF-file suffix-DN
   ```

   The -K option ensures that existing data is not removed. It also generates a file *generated-LDIF-file* that contains the new entries and information required by the replication process.

2. **On all other replicas, import the file generated in the previous step.**

   ```
   $ dsconf import -h host -p port \
   -K -f incremental-output=no generated-LDIF-file suffix-DN
   ```

   The option -f incremental-output=no specifies that an additional LDIF file will not be generated. Only one generated LDIF file is needed for this procedure.

Replication and Referential Integrity

If you are using the referential integrity plug-in with replication, you must enable it on all master servers. You do not need to enable it on hub or consumer servers.

The following limitations are associated with the use of the referential integrity plug-in in a replication environment:

- The plug-in must be enabled on all servers containing master replicas.
- You must enable the plug-in with the same configuration on every master.
- It is not useful to enable the plug-in on servers containing only hub or consumer replicas.

Maintaining Referential Integrity

Referential integrity is a plug-in mechanism that ensures that relationships between entries are maintained. Several types of attributes, such as those for group membership, contain the DN of another entry. Referential integrity can be used to ensure that when an entry is removed; all attributes that contain its DN are also removed.

For example, if a user's entry is removed from the directory and referential integrity is enabled, the server also removes the user from any groups of which the user is a member. If referential integrity is not enabled, the user must be manually removed from the group by the administrator. This is an important feature if you are integrating Directory Server with other Sun Java System products that rely on the directory for user and group management.

How Referential Integrity Works

When the referential integrity plug-in is enabled it performs integrity updates on specified attributes immediately after a delete, rename, or move operation. By default, the referential integrity plug-in is disabled.

Whenever you delete, rename, or move a user or group entry in the directory, the operation is logged to the referential integrity log file:

```
instance-path/logs/referent
```

After a specified time, known as the *update interval*, the server performs a search on all attributes for which referential integrity is enabled, and matches the entries resulting from that search with the DNs of deleted or modified entries present in the log file. If the log file shows that the entry was deleted, the corresponding attribute is deleted. If the log file shows that the entry was changed, the corresponding attribute value is modified accordingly.

When the default configuration of the referential integrity plug-in is enabled, it performs integrity updates on the member, **uniquemember**, **owner**, **seeAlso**, and **nsroledn** attributes immediately after a delete, rename, or move operation. You can, however, configure the behavior of the referential integrity plug-in to suit your own requirements.

The following behavior can be configured:

- Record referential integrity updates in a different file.
- Modify the update interval.
  If you want to reduce the impact that referential integrity updates has on your system, then increase the amount of time between updates.
- Select the attributes to which you apply referential integrity.

  If you use or define attributes containing DN values, then let the referential integrity plug-in monitor them.

Configure the Referential Integrity Plug-In

Note – All attributes in all databases that are used by the referential integrity plug-in must be indexed.
The indexes need to be created in the configuration of all the databases. When the retro changelog is enabled, the cn=changelog suffix must be indexed.

Certain limitations are associated with using the referential integrity plug-in in a replicated environment. For a list of these limitations, see “Replication and Referential Integrity.”

You can use DSCC to perform this task.

1. **Make sure that all replicas are configured and that all replication agreements are defined.**
2. **Determine the set of attributes for which you will maintain referential integrity and the update interval that you want to use on your master servers.**
3. **Enable the referential integrity plug-in on all master servers using the same set of attributes and the same update interval.**

   - To define the attributes for referential integrity, use this command:

     ```
     $ dsconf set-server-prop -h host -p port ref-integrity-attr:attribute-name \
     ref-integrity-attr:attribute-name
     ```

   - To add a referential integrity attribute to an existing list of attributes, use this command:

     ```
     $ dsconf set-server-prop -h host -p port ref-integrity-attr+:attribute-name
     ```

- To define the referential integrity update interval, use this command:

  ```
  $ dsconf set-server-prop -h host -p port ref-integrity-check-delay:duration
  ```

- To enable referential integrity, use this command:

  ```
  $ dsconf set-server-prop -h host -p port ref-integrity-enabled:on
  ```

4. **Ensure that the referential integrity plug-in is disabled on all consumer servers.**

Replication over SSL

You can configure Directory Servers involved in replication so that all replication operations occur over an SSL connection.

To Configure Replication Operations for SSL

This procedure shows example commands for setting up replication on a replication topology with two masters.

Note – This example shows a simple replication configuration, using a self-signed certificate. When setting up replication over SSL in a production environment, you will have better security if you use Certificate Authority trusted certificates instead.

Replication over SSL will fail if the supplier server certificate is an SSL server-only certificate that cannot act as a client during an SSL handshake.

While replication is secure by SSL, authentication of the replication manager is still done using a simple bind and password. You can use client-based authentication to fully secure replication, but this requires more complex settings.

You can use DSCC to perform this task.

1. **Create new servers and start them.**

```
$ dsadm create -p 1389 -P 1636 /local/ds1
$ dsadm create -p 2389 -P 2636 /local/ds2

$ dsadm start /local/ds1
$ dsadm start /local/ds2
```

2. **On all servers, create empty suffixes.**

```
$ dsconf create-suffix -e -i -p 1389 dc=example,dc=com
$ dsconf create-suffix -e -i -p 2389 dc=example,dc=com
```

3. **On all servers, set the multimaster password file.**

```
$ dsconf set-server-prop -e -i -h example1.server -p 1389 \
def-repl-manager-pwd-file:/local/ds1/replmanrpwd1.txt
$ dsconf set-server-prop -e -i -h example2.server -p 2389 \
def-repl-manager-pwd-file:/local/ds1/replmanrpwd2.txt
```

4. **On all servers, enable replication.**

```
$ dsconf enable-repl -h example1.server -p 1389 -e -i -d 1 master dc=example,dc=com
$ dsconf enable-repl -h example2.server -p 2389 -e -i -d 2 master dc=example,dc=com
```

5. **On all servers, view the existing default certificate.**

```
$ dsadm show-cert -F der -o certfile1 /local/ds1 defaultCert
$ dsadm show-cert -F der -o certfile2 /local/ds2 defaultCert
```

6. **On all servers, add the CA trusted certificate from all other servers.**

```
$ dsadm add-cert --ca /local/ds1 "ds2 Repl Manager Cert" certfile2
$ dsadm add-cert --ca /local/ds2 "ds1 Repl Manager Cert" certfile1
```

7. **On all master and hub (source) servers, create replication agreements with all consumer (destination) servers.**

   Note that secure LDAP ports are used for the replication agreements.

```
$ dsconf create-repl-agmt -h example1.server -p 1389 -e -i \
--auth-protocol "ssl-simple" dc=example,dc=com example2.server:2636
$ dsconf create-repl-agmt -h example2.server -p 2389 -e -i \
--auth-protocol "ssl-simple" dc=example,dc=com example1.server:1636
```

8. **For all replication agreements, configure the authentication password file to be the replication manager password file of the consumer (destination) server in the replication agreement.**

```
$ dsconf set-repl-agmt-prop -h example1.server -p 1389 -e -i \
dc=example,dc=com example2.server:2636 auth-pwd-file:/local/ds1/replmanrpwd2.txt
$ dsconf set-repl-agmt-prop -h example2.server -p 2389 -e -i \
dc=example,dc=com example1.server:1636 auth-pwd-file:/local/ds1/replmanrpwd1.txt
```

   After you have initialized the suffixes, the supplier will send all replication update messages to the consumer over SSL and will use certificates if you chose that option. Customer initialization will also use a secure connection if performed through DSCC, using an agreement configure for SSL.

9. **On all servers, restart the server in order to take configuration changes into account.**

```
$ dsadm restart /local/ds1
$ dsadm restart /local/ds2
```

10. **On one of the master servers, initialize the suffix.**

```
$ dsconf import -h example1.server -p 1389 -e -i /tmp/Example.ldif
dc=example,dc=com
```

11. **On all servers not yet initialized, initialize the servers by using a replication agreement.**

```
$ dsconf init-repl-dest -e -i -h example1.server -p 1389 \
dc=example,dc=com example1.server:2636
```

Replication over a WAN

Directory Server enables you to perform all forms of replication including multimaster replication between machines connected through a wide area network (WAN). This replication allows supplier servers to initialize and update consumers by making optimal use of the bandwidth over networks with higher latency and lower bandwidth.

Note – When deploying or troubleshooting a replication topology that replicates over a WAN, you must check network speed, latency, and packet loss. Network problems in any of these areas might cause replication delay.

In addition, replication data transfer rates will always be less than what the available physical medium allows in terms of bandwidth. If the update volume between replicas cannot physically be made to fit into the available bandwidth, tuning will not prevent your replicas from diverging under heavy update load. Replication delay and update performance are dependent on many factors, including but not limited to: modification rate, entry size, server hardware, error rates, average latency, and average bandwidth.

If you have questions about replication in your environment, contact your Sun Service Provider.

Internal parameters of the replication mechanism are optimized by default for WANs. However, if you experience slow replication due to the factors mentioned previously, you might want to empirically adjust the window size and group size parameters. You might also be able to schedule your replication to avoid peak network times, thus improving your overall network usage. Finally, Directory Server supports the compression of replication data to optimize bandwidth usage.

Configuring Network Parameters

The window and group network parameters determine how the replication mechanism groups entries to send them more efficiently over the network. These parameters affect how suppliers and consumers exchange replication update messages and acknowledgments. The parameters are configurable in every replication agreement, which allows you to tailor the replication performance according to the specific network conditions of each consumer.

Monitor the effects of any modifications that you make and adjust the parameters accordingly. You do not need to interrupt replication to modify the window size and group size parameters.

Configuring Window Size

The window size (default value 10) represents the maximum number of update messages that can be sent without immediate acknowledgment from the consumer.

It is more efficient to send many messages in quick succession instead of waiting for an acknowledgment after each message. Using the appropriate window size, you can eliminate the time replicas spend waiting for replication updates or acknowledgments to arrive.

If your consumer replica is lagging behind the supplier, increase the window size to a higher value than the default, such as 100, and check replication performance again before making further adjustments. When the replication update rate is high and the time between updates is therefore small, even replicas connected by a local area network (LAN) can benefit from a higher window size.

To Configure Window Size

You can use DSCC to perform this task.

- **Modify the window size.**

  ```
  $ dsconf set-repl-agmt-prop -h host -p port suffix-DN consumer-host:consumer-port transport-window-size:value
  ```

  For example:

```
$ dsconf set-repl-agmt-prop -h host2 -p 1389 dc=example,dc=com host1:1389 \
transport-window-size:20
```

Configuring Group Size

The group size (default value 1) represents the maximum number of data modifications that can be bundled into a single update message. If the network connection appears to be impeding replication, increase the group size to a higher value than the default, such as 10, and recheck replication performance.

When increasing the group size, make sure that the following are true:

- The window size is set significantly higher than the group size.
- The window size divided by the group size is much greater than the value for nsslapd-maxThreadsPerConn under cn=config on the consumer (typically twice as large).

  When the group size is set higher than 1, the supplier does not wait to fill a group before sending updates to the consumer.

To Configure Group Size

You can use DSCC to perform this task.

- **Modify the group size.**

  ```
  $ dsconf set-repl-agmt-prop -h host -p port suffix-DN \
  consumer-host:consumer-port transport-group-size:value
  ```

  For example:

  ```
  $ dsconf set-repl-agmt-prop -h host2 -p 1389 dc=example,dc=com host1:1389 \
  transport-group-size:10
  ```

Scheduling Replication Activity

If immediate synchronization between your replicas is not critical, you can schedule replication during periods of low network usage. Replication of data should complete significantly faster when the network is more available.

You can schedule replication to start and end at a certain time of day, on a daily or weekly basis. You can do this independently for every consumer through its replication agreement. The new schedule will take effect immediately, causing the next replication of data for the corresponding consumer to be delayed until first allowed by the schedule.

To Schedule Replication Activity

You can use DSCC to perform this task.

- **Modify the replication schedule.**

  ```
  $ dsconf set-repl-agmt-prop -h host -p port suffix-DN \
  host:port repl-schedule:value
  ```

  For example, if you want to set replication to occur between 2:00 and 4:00 every night, type:

  ```
  $ dsconf set-repl-agmt-prop -h host2 -p 1389 dc=example,dc=com host1:1389 \
  ```

```
repl-schedule:"0200-0400 0123456"
```

where 0123456 indicate the days of the week, with 0 representing Sunday, 1 representing Monday, and so on.

Configuring Replication Compression

To reduce the bandwidth used by replication, you may configure replication to compress the data that is sent when updating consumers. The replication mechanism uses the Zlib compression library. Both supplier and consumer must be running on a Solaris or Linux platform to enable compression.

You should empirically test and select the compression level that gives you best results for your expected replication usage in your WAN environment. Do not set this parameter in a LAN where there is wide network bandwidth because the compression and decompression computations will slow down replication.

To Configure Replication Compression

You cannot use DSCC to perform this task. Use the command line, as described in this procedure.

- **Configure replication compression on the replication agreement entry in the master server.**

  ```
  $ dsconf set-repl-agmt-prop -h host -p port suffix-DN \
  consumer-host:consumer-port transport-compression:level
  ```

  where level can be high, medium, low, or none.

  For example, to use the fastest compression when sending replication updates to the consumer on host1:1389, type:

  ```
  $ dsconf set-repl-agmt-prop -h host2 -p 1389 dc=example,dc=com host1:1389 \
  transport-compression:high
  ```

Modifying the Replication Topology

This section explains these aspects of managing an existing replication topology:

- Changing the Replication Manager
- Managing Replication Agreements
- Promoting or Demoting Replicas
- Disabling a Replicated Suffix
- Keeping Replicated Suffixes Synchronized

Changing the Replication Manager

You can edit a replication agreement to change the replication manager identity that is used to bind to the consumer server. To avoid any interruption of the replication, you should define the new replication manager entry or certificate entry on the consumer before modifying the replication agreement. However, if replication is interrupted due to a bind failure, the replication mechanism will automatically send all the necessary updates when you correct the error, within the limits of the replication recovery settings. For the procedure, see “Using a Non-Default Replication Manager.”

Managing Replication Agreements

You can disable, enable, or delete a replication agreement.

Disabling a Replication Agreement

When a replication agreement is disabled, the master stops sending updates to the designated consumer. Replication to that server is stopped, but all settings in the agreement are preserved.

You may resume replication by re-enabling the agreement at a later time. See “Enabling a Replication Agreement” for information about resuming the replication mechanism after an interruption.

To Disable a Replication Agreement

You can use DSCC to perform this task. For information, refer to DSCC online help.

Disable a replication agreement.

```
$ dsconf disable-repl-agmt -h host -p port suffix-DN consumer-host:consumer-port
```

For example:

```
$ dsconf disable-repl-agmt -h host2 -p 1389 dc=example,dc=com host1:1389
```

Enabling a Replication Agreement

Enabling a replication agreement resumes replication with the designated consumer. However, if replication has been interrupted longer than the replication recovery settings allow and the consumer was not updated by another supplier, you must reinitialize the consumer. The replication recovery settings are the maximum size and age of this supplier’s change log and the purge delay of the consumer (see “To Perform Advanced Consumer Configuration”).

When the interruption is short and replication can recover, the master will update the consumer automatically when the agreement is re-enabled.

To Enable a Replication Agreement

You can use DSCC to perform this task.

Enable a replication agreement.

```
$ dsconf -h host -p port enable-repl-agmt suffix-DN consumer-host:consumer-port
```

For example:

```
$ dsconf -h host2 -p 1389 enable-repl-agmt dc=example,dc=com host1:1389
```

Deleting a Replication Agreement

Deleting a replication agreement stops the replication to the corresponding consumer and removes all configuration information about the agreement. If you want to resume replication at a later date, disable the agreement instead, as described in “Disabling a Replication Agreement.”

To Delete a Replication Agreement

You can use DSCC to perform this task.

- **Delete a replication agreement.**

```
$ dsconf delete-repl-agmt -h host -p port suffix-DN consumer-host:consumer-port
```

For example:

```
$ dsconf delete-repl-agmt -h host2 -p 1389 dc=example,dc=com host1:1389
```

Promoting or Demoting Replicas

Promoting or demoting a replica changes its role in the replication topology. Dedicated consumers can be promoted to hubs, and hubs can be promoted to masters. Masters can be demoted to hubs, and hubs can also be demoted to dedicated consumers. However, masters cannot be demoted directly to consumers, just as consumers cannot be promoted directly to masters.

The allowed promotions and demotions within the multimaster replication mechanism make the topology very flexible. A site that was formerly served by a consumer replica might grow and require a hub with several replicas to handle the load. If the load includes many modifications to the replica contents, the hub can become a master to allow faster local changes that can then be replicated to other masters at other sites.

When promoting or demoting replicas, be aware of the following:

- If you promote a consumer, it becomes a hub. If you promote a hub, it becomes a master. You cannot promote a server directly from consumer to master. You must first promote the consumer to a hub, then promote the hub to a master. Likewise, when demoting a master to a consumer, you must demote the master to a hub before demoting from a hub to a consumer.

- When demoting a master to a hub, the replica will become read-only and be configured for sending referrals to the remaining masters. The new hub will retain all of its consumers, whether hubs or dedicated consumers.

- Demoting a single master to a hub will create a topology without a master replica. Directory Server will allow you to do this under the assumption that you will define a new master. However, it is better to add a new master as a multimaster and allow it to be initialized before demoting the other master.

- Before demoting a hub to a consumer, you must disable or delete all replication agreements to and from the hub. If you do not do this, the demote operation will fail with this error: LDAP_OPERATIONS_ERROR “Unable to demote a hub to a read-only replica if some agreements are enabled”.

  If the hub’s consumers were not updated by other hubs or masters, they will no longer be updated. You should create new agreements on the remaining hubs or masters to update these consumers.

- When promoting a consumer to a hub, its change log is enabled, and you may define new agreements with consumers.

- When promoting a hub to a master, the replica will accept modification requests, and you may define new agreements with other masters, hubs, or dedicated consumers.

To Promote or Demote a Replica

You can use DSCC to perform this task.

- **Promote or demote a replica by using one of the commands:**

  ```
  $ dsconf promote-repl -h host -p port role suffix-DN
  $ dsconf demote-repl -h host -p port role suffix-DN
  ```

  where role is master, hub or consumer.

Disabling a Replicated Suffix

Disabling a replicated suffix removes it from the replication topology. It will no longer be updated or send updates, depending on its role as a master, hub, or consumer. Disabling a suffix on a supplier server deletes all replication agreements, and they will have to be re-created if the replica is enabled again.

To Disable a Replicated Suffix

You can use DSCC to perform this task.

- **Disable a replicated suffix.**

  ```
  $ dsconf disable-repl -h host -p port suffix-DN
  ```

  For example:

  ```
  $ dsconf disable-repl -h host2 -p 1389 dc=example,dc=com
  ```

Keeping Replicated Suffixes Synchronized

After you stop a Directory Server involved in replication for regular maintenance, when it comes back online, you need to ensure that it gets updated through replication immediately. In the case of a master in a multimaster environment, the directory information needs to be updated by another master in the multimaster set. In other cases, after a hub server or a dedicated consumer server is taken offline for maintenance, when they come back online, they need to be updated by the master server.

This section describes the replication retry algorithm and explains how to force replication updates to occur without waiting for the next retry.

Note – The procedures described in this section can be used only when replication is already set up and consumers have been initialized.

Replication Retry Algorithm

When a source replica is unsuccessful in replicating to a destination, it retries periodically in incremental time intervals. The retry intervals depend on the error type.

Note that even if you have configured replication agreements to always keep the source replica and the destination replica synchronized, this is not sufficient to immediately update a replica that has been offline for over five minutes.

To Force Replication Updates

If replication has stopped, you can force replication updates to the destination suffixes.

You cannot use DSCC to perform this task. Use the command line, as described in this procedure.

- **On the source server, restart replication updates to the destination server.**

  ```
  $ dsconf update-repl-dest-now -h host -p port suffix-DN destination-
  host:destination-port
  ```

  For example:

  ```
  $ dsconf update-repl-dest-now -h host2 -p 1389 dc=example,dc=com host1:1389
  ```

Moving a Master Replica to a New Machine

In some situations, it might be necessary to move a master replica to a different machine. If you do not need to use the same host name and port number, use dsconf change-repl-dest to change the host name and port number of the remote replica. For more information, see "To Change the Destination of a Replication Agreement."

If you need to retain the same host name and port number, you must remove the master from the existing topology, and then re-add the master to the topology.

It is much easier to use DSCC to perform these tasks, because DSCC takes care of any impacted replication agreements. If you use DSCC, however, you cannot specify the same replica ID that the master originally had in the topology. To use the same replica ID, you must use the command line to perform these tasks, as follows.

To Remove a Master From an Existing Replication Topology

Make sure that all changes from the master have already been replicated.

1. **If you can, back up the master using binary copy so that you do not lose any changes.**

2. **Demote the master replica to a hub replica.**

   See "Promoting or Demoting Replicas."

3. **Wait for the hub to start replicating to other servers.**
   When the hub opens a replication session to the other servers in the topology, it remains in the RUV but is no longer used in referrals.

4. **Stop the hub.**
   See "Starting, Stopping, and Restarting a Directory Server Instance."

5. **Remove the hub from the topology.**
   See "Disabling a Replicated Suffix."

To Add a Master to an Existing Replication Topology

1. **Add the master replica, using the same replica ID.**
   See "Enabling Replication on a Master Replica."

2. **Recreate the replication agreements from that master to the other replicas in the topology.**

3. **Initialize the new master.**

a. **If you were able to back up the master, initialize the master from this backup.**
b. If you were not able to back up the master (in the event of a machine crash), initialize the master from another master in the topology.

Replication With Releases Prior to Directory Server 6.2

This section provides information about how to configure replication with releases of Directory Server prior to 6.2.

Replicating Between Directory Server 6.2 and Directory Server 5.1 or 5.2

Directory Server 5.1, 5.2 and 6.2 are compatible with regard to replication configuration, with the following exceptions:

- Replication priority is not supported in releases prior to Directory Server 6.2. If you configure replication priority on a 6.2 master replica, the replication priority will be transferred to consumers running Directory Server 6.2, but not to any consumers running a previous version of Directory Server.

- An unlimited number of masters is not supported on replication topologies that contain Directory Server 5.1 or 5.2 masters. Although Directory Server 6.2 supports an unlimited number of masters in a replication topology, this number is limited to four if your replication topology includes any Directory Server 5.2 master servers. Directory Server 5.1 does not support multimaster replication.

Using the Retro Change Log

The retro change log is used by LDAP clients to ascertain the history of changes made to the Directory Server data. The retro change log is stored in a separate database to the Directory Server change log, under the suffix cn=changelog.

A retro change log can be enabled on a standalone server or on each server in a replication topology. When the retro change log is enabled on a server, by default updates to all suffixes on that server are logged. The retro change log can be configured to log updates to specified suffixes only.

For information about using the retro change log in a replicated topology and about restrictions on using the retro change log, see “Replication and the Retro Change Log Plug-In” in *Sun Java System Directory Server Enterprise Edition 6.2 Reference.*

For information about the attributes of an entry in the retro change log, see the changeLogEntry(5dsoc) man page.

For more information about modifying the retro change log, see the dsconf(1M) man page.

This section explains various ways that you can use the retro change log.

To Enable the Retro Change Log

To use the retro change log, you must enable it.

You cannot use DSCC to perform this task. Use the command line, as described in this procedure.

1. **Modify the retro change log configuration entry:**

   ```
   $ dsconf set-server-prop -h host -p port retro-cl-enabled:on
   ```

2. **Restart the server.**

   For information, see “Starting, Stopping, and Restarting a Directory Server Instance.”

To Configure the Retro Change Log to Record Updates for Specified Suffixes

When the retro change log is enabled on a server, by default it records updates to all suffixes on the server. This procedure describes how to configure the retro change log to record updates to specified suffixes only.

You cannot use DSCC to perform this task. Use the command line, as described in this procedure.

1. **Modify the retro change log configuration entry:**

   ```
   $ dsconf set-server-prop -h host -p port retro-cl-suffix-dn:suffix-DN
   ```

   For example, to log changes only on the cn=Contractors,dc=example,dc=com suffix and the ou=People,dc=example,dc=com suffix, use this command:

   ```
   $ dsconf set-server-prop -h host2 -p 1389 \
   retro-cl-suffix-dn:"cn=Contractors,dc=example,dc=com" \
   retro-cl-suffix-dn:"ou=People,dc=example,dc=com"
   ```

   To add a suffix to an existing list of specified suffixes, use this command:

   ```
   $ dsconf set-server-prop -h host -p port retro-cl-suffix-dn+:suffix-DN
   ```

2. **Restart the server.**

   For information, see “Starting, Stopping, and Restarting a Directory Server Instance.”

To Configure the Retro Change Log to Record Attributes of a Deleted Entry

This procedure describes how to configure the retro change log to record specified attributes of an entry when that entry is deleted.

You cannot use DSCC to perform this task. Use the command line, as described in this procedure.

1. **Specify the attributes that must be recorded:**

   ```
   $ dsconf set-server-prop -h host -p port retro-cl-deleted-entry-attr: \
   attribute1 attribute2
   ```

   For example, to set the retro change log to record the UID attributes of deleted entries, use this command:

```
$ dsconf set-server-prop -h host -p port retro-cl-deleted-entry-attr:uid
```

To add an attribute to an existing list of specified attributes, use this command:

```
$ dsconf set-server-prop -h host -p port retro-cl-deleted-entry-attr+:attribute
```

2. **Restart the server.**

   For information, see “Starting, Stopping, and Restarting a Directory Server Instance.”

To Trim the Retro Change Log

The entries in the retro change log can be removed automatically after a specified period of time. To configure the period of time after which entries are deleted automatically, make sure that the retro change log is enabled, then set the nsslapd-changelogmaxage configuration attribute in the cn=Retro Changelog Plugin, cn=plugins, cn=config entry.

You cannot use DSCC to perform this task. Use the command line, as described in this procedure.

1. **Check that the retro change log is enabled.**

   ```
   $ dsconf get-server-prop -h host -p port retro-cl-enabled
   ```

2. **If the retro change log is not enabled, enable it.**

   ```
   $ dsconf set-server-prop -h host -p port retro-cl-enabled:on
   ```

3. **Set the maximum age for changes logged.**

   ```
   $ dsconf set-server-prop -h host -p port retro-cl-max-age:duration
   ```
   where duration can be either undefined (no age limit) or one of the following:

   - s for seconds
   - m for minutes
   - h for hours
   - d for days
   - w for weeks

   For example, to set the retro change log maximum age to two days, type:

   ```
   $ dsconf set-server-prop -h host 2 -p 1389 retro-cl-max-age:2d
   ```

   The retro change log will be trimmed at the next operation on the change log.

Accessing Control and the Retro Change Log

The retro change log supports search operations. It is optimized for searches that include filters of this form:

(&(changeNumber>=X)(changeNumber<=Y))

As a general rule, do not perform add or modify operations on the retro change log entries. You can delete entries to trim the size of the log. The only time that you need to perform a modify operation on the retro change log is to modify the default access control policy.

When the retro change log is created, by default, the following access control rules apply:

- Read, search, and compare rights are granted to all authenticated users (userdn=anyone, not to be confused with anonymous access where userdn=all) to the retro change log top entry cn=changelog.

- Write and delete access are not granted, except implicitly to the Directory Manager.

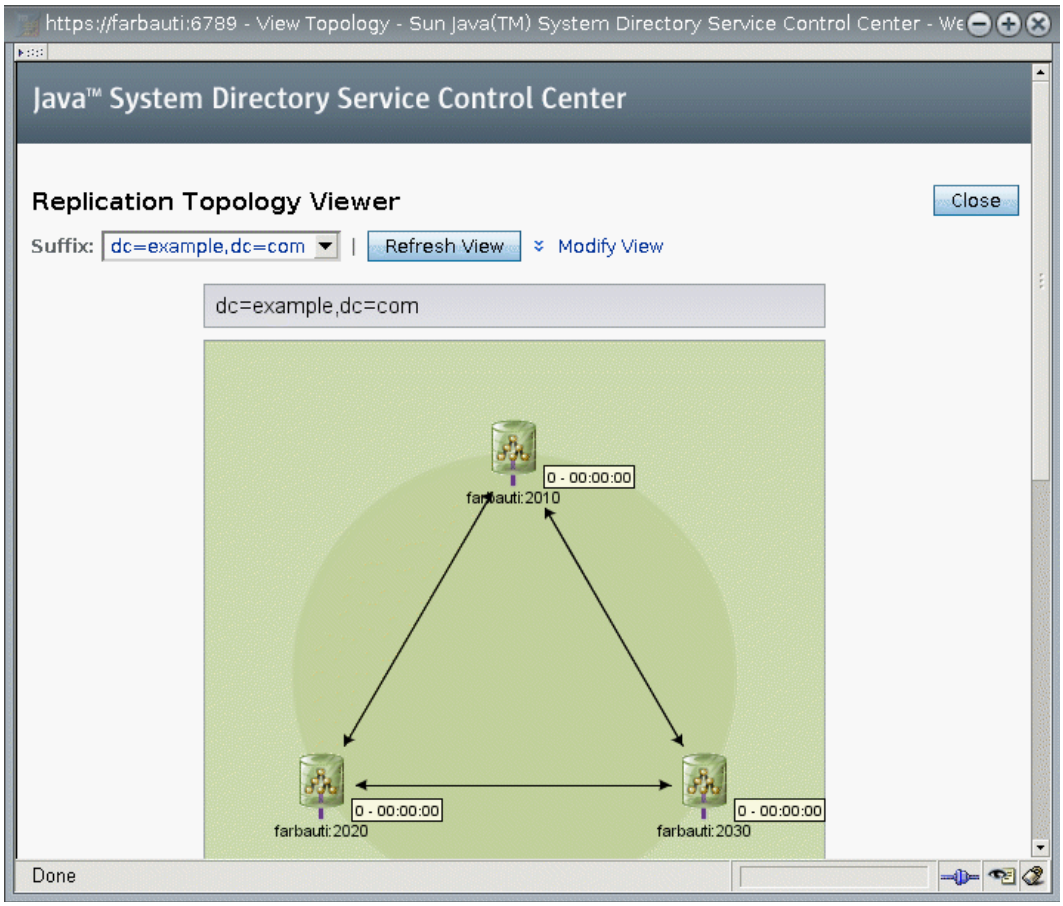
  Do not grant read access to anonymous users because the retro change log entries can contain modifications to sensitive information such as passwords. You may want to further restrict access to the retro change log contents if authenticated users should not be allowed to view its contents.

Getting Replication Status

You can get replication status by using DSCC or by using command-line tools.

Getting Replication Status in DSCC

You can graphically view replication, including replication agreements and replication delay, by using the Suffix tab. For more information, see the DSCC online help. In addition, you can use DSCC to view your replication topology, as shown in following figure.



Getting Replication Status by Using the Command Line

If you are unable to use DSCC, use command-line tools to obtain information about your replication deployment.

The man pages provide full command-line syntax and usage examples for these tools.

- repldisc - “Discovers” and constructs a table of all known servers in a replication deployment. See the repldisc(1) man page.
- insync - Indicates the synchronization state between a supplier and one or more consumer replicas. See the insync(1) man page.
- entrycmp - Compares the same entry in two or more replicas. See the entrycmp(1) man page.

Solving Common Replication Conflicts

Multimaster replication uses a loose consistency replication model. This means that the same entries may be modified simultaneously on different servers. When updates are sent between the two servers, any conflicting changes must be resolved. Most resolution occurs automatically. For example, the timestamp associated with the change on each server is resolved by the most recent change taking precedence. However, some change conflicts require manual intervention to reach a resolution.

This section covers the following topics:

- Solving Replication Conflicts by Using DSCC
- Solving Replication Conflicts by Using the Command Line
- Solving Naming Conflicts
- Solving Orphan Entry Conflicts
- Solving Potential Interoperability Problems

Solving Replication Conflicts by Using DSCC

The easiest way to resolve a replication conflict is by using DSCC. See the DSCC online help for information.

Solving Replication Conflicts by Using the Command Line

You can solve replication conflicts by using the command line. Entries that have a change conflict that cannot be resolved automatically by the replication process contain the operational attribute nsds5ReplConflict as a conflict marker.

To find entries with conflicts, periodically search for entries that contain this attribute. For example, you could use the following ldapsearch command to find entries with conflicts:

```
$ ldapsearch -h host2 -p 1389 -D cn=admin,cn=Administrators,cn=config \
-w - -b "dc=example,dc=com" "(nsds5ReplConflict=*)"
```

Note that the nsds5ReplConflict attribute is indexed by default.

Solving Naming Conflicts

Entries with identical DNs may be created on separate masters if they are created before the servers replicate the changes to each other. Upon replication, the conflict resolution mechanism will automatically rename the second entry created.

An entry with a DN naming conflict is renamed by including its unique identifier, provided by the operational attribute nsuniqueid, in its DN.

For example, if the entry uid=bjensen,ou=People,dc=example,dc=com is created simultaneously on two masters, both will have the following two entries after replication:

- uid=bjensen,ou=People,dc=example,dc=com
- nsuniqueid=66446001-1dd211b2-66225011-2ee211db+uid=bjensen,dc=example,dc=com

The second entry must be given a useful DN. You can delete the conflicting entry and add it again with a non-conflicting name. However, renaming the entry ensures that its contents have not changed. The renaming procedure depends on whether the naming attribute is single-valued or multivalued. See the following procedures.

To Rename a Conflicting Entry That has a Multivalued Naming Attribute

You can use DSCC to perform this task. For information, refer to DSCC online help.

1. **Rename the entry while keeping the old RDN value, for example:**

```
$ ldapmodify -h host2 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn: nsuniqueid=66446001-1dd211b2-66225011-2ee211db+uid=bjensen,dc=example,dc=com
changetype: modrdn
newrdn: uid=bj66446001
deleteoldrdn: 0
^D
```

   You cannot delete the old RDN value in this step because it also contains the nsuniqueid operational attribute, which cannot be deleted.

2. **Remove the old RDN value of the naming attribute and the conflict marker attribute, for example:**

```
$ ldapmodify -h host2 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn: uid=bj66446001,dc=example,dc=com
changetype: modify
delete: uid
uid: bjensen
-
delete: nsds5ReplConflict
^D
```

To Rename a Conflicting Entry With a Single-Valued Naming Attribute

When the naming attribute in a duplicate entry is single-valued, for example dc (domain component), you cannot simply rename the entry to another value of the same attribute. Instead, you must give the entry a temporary name. You can use DSCC to perform this task. For information, refer to DSCC online help.

1. **Rename the entry by using a different naming attribute, and keep the old RDN, for example:**

```
$ ldapmodify -h host2 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn: nsuniqueid=66446001-1dd211b2-66225011-2ee211db+dc=HR,dc=example,dc=com
changetype: modrdn
newrdn: o=TempHREntry
```

```
deleteoldrdn: 0
^D
```

You cannot delete the old RDN value in this step because it also contains the nsuniqueid operational attribute, which cannot be deleted.

2. **Change the desired naming attribute to a unique value and remove the conflict marker attribute, for example:**

```
$ ldapmodify -h host2 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn: o=TempHREntry,dc=example,dc=com
changetype: modify
replace: dc
dc: NewHR
delete: nsds5ReplConflict
^D
```

3. **Rename the entry back to the intended naming attribute, for example:**

```
$ ldapmodify -h host2 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn: dc=NewHR,dc=example,dc=com
changetype: modrdn
newrdn: dc=HR
deleteoldrdn: 1
^D
```

By setting the value of the deleteoldrdn attribute to 1, you delete the temporary attribute-value pair o=TempHREntry. If you want to keep this attribute, set the value of the deleteoldrdn attribute to 0.

Solving Orphan Entry Conflicts

When a delete operation is replicated, and the consumer server finds that the entry to be deleted has child entries, the conflict resolution procedure creates a glue entry to avoid having orphaned entries in the directory.

In the same way, when an add operation is replicated, and the consumer server cannot find the parent entry, the conflict resolution procedure creates a glue entry representing the parent so that the new entry is not an orphan entry.

Glue entries are temporary entries that include the object classes glue and extensibleObject. Glue entries can be created in various ways:

- If the conflict resolution procedure finds a deleted entry with a matching unique identifier, the glue entry is a resurrection of that entry. It also includes the glue object class and the nsds5ReplConflict attribute.

  In such cases, you can either modify the glue entry to remove the glue object class and the nsds5ReplConflict attribute to keep the entry as a normal entry, or delete the glue entry and its child entries.

- The server creates a minimal entry with the glue and extensibleObject object classes. In such cases, you must either modify the entry to turn it into a meaningful entry or delete the entry and all of its child entries.

Solving Potential Interoperability Problems

For interoperability with applications that rely on attribute uniqueness, such as a mail server, you might need to restrict access to the entries that contain the nsds5ReplConflict attribute. If you do not restrict access to these entries, the applications that require only one attribute will pick up both the original entry and the conflict resolution entry that contains the nsds5ReplConflict and operations will fail.

To restrict access, you need to modify the default ACI that grants anonymous read access using the following command:

```
$ ldapmodify -h host2 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn: dc=example,dc=com
changetype: modify
delete: aci
aci: (target ="ldap:///dc=example,dc=com")
(targetattr !="userPassword"
(version 3.0;acl "Anonymous read-search access";
allow (read, search, compare)(userdn = "ldap:///anyone");)
-
add: aci
aci: (target="ldap:///dc=example,dc=com")
(targetattr!="userPassword")
(targetfilter="(!(nsds5ReplConflict=*))")(version 3.0;acl
"Anonymous read-search access";allow (read, search, compare)
(userdn="ldap:///anyone");)
^D
```

The new ACI will keep entries that contain the nsds5ReplConflict attribute from being returned in search results.