HP Client Automation Distributed Configuration Server

For the Windows ® operating system

Software Version: 8.10

Reference Guide

Document Release Date: February 2012

Software Release Date: February 2012

Legal Notices

Warranty

Restricted Rights Legend

Copyright Notice

Trademark Notices

Acknowledgements

This product includes software developed by the Apache Software Foundation (http://www.apache.org/).

This product includes cryptographic software written by Eric Young (eay@cryptsoft.com).

This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (http://www.openssl.org/).

This product includes software written by Tim Hudson (tjh@cryptsoft.com).

Documentation Updates

The title page of this document contains the following identifying information:

- Software Version number, which indicates the software version.
- Document Release Date, which changes each time the document is updated.
- Software Release Date, which indicates the release date of this version of the software.

To check for recent updates or to verify that you are using the most recent edition of a document, go to:

http://h20230.www2.hp.com/selfsolve/manuals

This site requires that you register for an HP Passport and log on. To register for an HP Passport ID, go to:

http://h20229.www2.hp.com/passport-registration.html

Or click the **New users - please register** link on the HP Passport log on page.

You will also receive updated or new editions if you subscribe to the appropriate product support service. Contact your HP sales representative for details.

Support

Visit the HP Software Support Online web site at:

http://www.hp.com/go/hpsoftwaresupport

This web site provides contact information and details about the products, services, and support that HP Software offers.

HP Software online support provides customer self-solve capabilities. It provides a fast and efficient way to access interactive technical support tools needed to manage your business. As a valued support customer, you can benefit by using the support web site to:

- Search for knowledge documents of interest
- Submit and track support cases and enhancement requests
- Download software patches
- Manage support contracts
- Look up HP support contacts
- Review information about available services
- Enter into discussions with other software customers
- Research and register for software training

Most of the support areas require that you register as an HP Passport user and sign in. Many also require a support contract. To register for an HP Passport ID, go to:

http://h20229.www2.hp.com/passport-registration.html

To find more information about access levels, go to:

http://h20230.www2.hp.com/new_access_levels.jsp

Contents

Introduction to the Distributed Configuration Server

The Distributed Configuration Server (DCS) is used to synchronize the Configuration Server Database (CSDB) *contents only* and forms a subset of the normal synchronization process of the Core to Satellites.

The DCS components are pre-configured with the Core and Satellite installation. You do not have to change these configurations or run the DCS Destination Component (`dmabatch`) to synchronize the CSDB contents. However, you can customize the configuration or troubleshoot the DCS by using this reference guide.

Overview

Distributed Configuration Server is a tool that enables an administrator to manage multiple **Configuration Server Databases (CSDBs)** in a networked environment.

An administrator can use DCS to replicate domains from one CSDB to another. This means that enterprise-wide changes can be made once, on one Configuration Server machine, then simply distributed to (replicated on) multiple down-line Configuration Servers within the enterprise. This ability offers several benefits:

- **Efficiency**
  Less time and manpower are required for making the database changes.
- **Reliability**
  Increased control over the integrity of the changes that are made.

Terminology

The following table describes the Distributed Configuration Server-specific terms that are used in this document.

Distributed Configuration Server Terminology

Term	Description
Distributed Configuration Server	The Distributed Configuration Server is an extension of the Configuration Server. It synchronizes Configuration Server Databases (CSDB) that are running on separate (DCS-enabled) machines across an enterprise.
Integration Server	The HTTP file server of Client Automation. It gets installed on a Source Configuration Server in order to facilitate multiple concurrent file transfer sessions and the creation of the container file.
Source Configuration Server (**master**)	In a DCS synchronization, the Configuration Server from which the Destination Configuration Server will receive database changes.
Destination Configuration Server (**slave**)	In a DCS synchronization, the (target) Configuration Server on which CSDB changes will be replicated. Note: This is *always* a replica of the Source database.
Synchronization	The replicating of administrator-specified domains from one Configuration Server Database (Source Configuration Server) to another (Destination Configuration Server).

Term	Description
Peer Synchronization	Synchronizing a Domain on a Destination Configuration Server from a Source Configuration Server that does not own the Domain. See Foreign-Owned Domain in this table.
Synchronization Pair	The two Configuration Servers that have been selected as the Source and Destination.
Domain Ownership	All Domains are "owned" by a Configuration Server. Domains are either **self-owned** or **foreign-owned**. See "Domain Ownership" (on page 12).
Self-Owned Domain	A Domain that is owned by the current Configuration Server. Note: In order for a Domain to be self-owned, the owning MGR_ID and current MGR_ID must be identical.
Foreign-Owned Domain	A Domain that is owned by a Configuration Server other than the current one. Note: If the owning MGR_ID and current MGR_ID are different, the Domain is foreign-owned.
Unrelated Domains	Domains that are not owned by the same Configuration Server—that is, they do not have the same owning MGR_ID.
Middle-tier Configuration Server	A middle-tier Configuration Server is not an HP product. Rather this term is used exclusively to reference any Configuration Server on which the Source and Destination components are installed so that it can support peer synchronizations.
Container File	A file, created on the Source, in which the instance data is compressed before being transferred to the Destination. This file is much faster to transfer than a large number of small files. Note: At the Commit phase, the instance-container file is used as the data source, so the files are moved directly from it to their ultimate destination. This minimizes the number of times that the data is moved and the length of time that the Configuration Server Database is hard-locked.
edmprof file	This is the text file in which the operational parameters of the Configuration Server are specified. • On Windows platforms, this file is `edmprof.dat`. Note: This guide uses this non-platform specific, generic reference.

Distributed Configuration Server

The Distributed Configuration Server product is a two-piece extension of the Configuration Server. The components—**Source** and **Destination**—function separate from, but in conjunction with, one another. Both, however, have some dependence on a Configuration Server; therefore, each must be co-located with a Configuration Server.

Note: For more information on the functionality of the DCS components, see the section, "Distributed Configuration Server Components" (on page 10)

In a **multi-tier** configuration, both components can be installed on the same machine in order to accommodate *peer synchronizations* via a *middle-tier Configuration Server*. (Peer synchronizations and middle-tier Configuration Servers are defined in the table Distributed Configuration Server Terminology.)

DCS is designed to synchronize DCS-enabled Configuration Server Databases throughout an enterprise so, although it is not essential that the CSDBs directly communicate with one another, DCS must be able to communicate with both Configuration Servers that comprise the synchronization pair (see "Two Configuration Servers: A Synchronization Pair" (on page 11)).

In a synchronization, DCS compares the control information of one CSDB with that of another, for the domains that have been selected.

Note: If SSL functionality is used during a DCS synchronization, files larger than 2 GB will fail.

IP Networking Support

HP Client Automation supports **IP version 4 (IPv4)** and **IP version 6 (IPv6)**. The Core and Satellite servers can now use either IPv4 or IPv6 for server-to-server communications. HPCA agent communications, however, are currently limited to IPv4. For more information, see the *IPv6 Networking Support* appendix in the *HP Client Automation Core and Satellite Enterprise Edition User Guide*.

Note: HP Client Automation environments that use the traditional, component-based, HPCA server installations will continue to be supported on IPv4 only.

Distributed Configuration Server Components

Inasmuch as there are two Configuration Servers involved in DCS synchronizations, the two DCS components perform different functions and must be installed separately, based on the intended role of the host Configuration Server.

- Each Configuration Server that will act as a Source must have the DCS *Source* component installed.
- Similarly, each Configuration Server that will act as a Destination must have the DCS *Destination* component installed.
- If a Configuration Server has both components of the DCS installed, it can act as Source and Destination, albeit in separate DCS operations.

With the DCS components installed on the Configuration Server machines, DCS:

- Provides the synchronization facilities to contact the Source and Destination,
- Reconciles the differences between the selected domains, and
- Provides the intermediate facilities to make identical the Source and Destination domains.

Note: The Destination is always a replica of the source.

The following section offers a more detailed look at these components and their functions.

Source vs. Destination

The Source and Destination components perform different functions during the Distributed Configuration Server synchronization. Therefore, it is important to correctly install these components in order to ensure: 1) the availability and accessibility of the appropriate Source-Destination synchronization pairs, and 2) the expected synchronization results.

Source Component

The Source component must be installed on any Configuration Server that is going to function as the master in a synchronization. This component contains the Integration Server, the product suite’s HTTP server. For a brief description of Integration Server and how it relates to DCS, see " Integration Server" (on page 11).

The Source component loads the database instances into a single repository. This repository can be directly accessed, thereby eliminating the excessive overhead of opening, storing, transferring, and writing individual files for each CSDB instance.

Integration Server

Integration Server is the Client Automation product suite’s HTTP file server. It facilitates multiple concurrent file transfer sessions (HTTP “get” requests) and the creation of the instance-container file (see Container File in table "Distributed Configuration Server Terminology").

Integration Server is not a separately licensed HP product. It integrates several independent modules—**HP Client Automation Portal (Portal)** and **HP Client Automation Proxy Server (Proxy Server)**—giving them access to all the functions and resources under its control.

Destination Component

The Destination component must be installed on any Configuration Server that is going to function as the target in DCS synchronization. This component provides direct access to the target file system.

Two Configuration Servers: A Synchronization Pair

Two Configuration Servers, one defined as the Source and the other as the Destination, comprise a Distributed Configuration Server *synchronization pair*.

Note: Cross-format synchronizations are not supported. In order to be synchronized, the Source and Destination databases must be of the same format; that is, a **UTF-8** database to a **UTF-8** database, and a **Legacy** database to a **Legacy** database.

DCS accepts one synchronization pair only, per execution. Operationally, because a synchronization can go in only one direction, this means that if two Configuration Servers (for example, MGR_001 and MGR_002) need domains from one another, two DCS executions must be done—with MGR_001 being the Source in one synchronization, and MGR_002 being the Source in the other.

Configuration Server Eligibility

In order to be eligible to participate in a DCS operation, a Configuration Server must meet the following requirements.

- In its `edmprof` file, it must be DCS-enabled. This is done by specifying:

  ```
  [MGR_STARTUP]MANAGER_TYPE=DISTRIBUTED
  ```

  or

  ```
  [MGR_STARTUP]MANAGER_TYPE=SERVER
  ```

  Note: All Configuration Servers are installed as DISTRIBUTED, so the SERVER value will have to be manually specified in the `edmprof` file. For performance-improvement information, see "Improving Performance using MANAGER_TYPE" (on page 12).

- It must have either the DCS Source or Destination component installed.

Improving Performance using MANAGER_TYPE

Typically, Destination and middle-tier Configuration Servers (see "Middle-tier Configuration Server" in table "Distributed Configuration Server Terminology") get database updates from their up-line Source Configuration Server only. Therefore, some default database administrative processes are not necessary. HP indicates that disabling these unnecessary processes will improve performance.

- If a DCS-enabled Configuration Server is going to be a Destination or middle-tier Configuration Server—getting its updates from a Source Configuration Server only and not managed via any other process—you can improve its performance by setting:

  ```
  [MGR_STARTUP]
  MANAGER_TYPE=SERVER
  ```

Domain Ownership

Configuration Server Database domains on each Configuration Server have three distinguishing characteristics: **domain name**, **owning MGR_ID**, and **current MGR_ID**. Their ownership is determined by the value of MGR_ID, and is established:

- When a CSDB is installed.
- When a domain is added to a CSDB.
  Note: For planning purposes, HP recommends maintaining unique names for CSDB domains.

A self-owned domain is a CSDB domain that is owned by the current Configuration Server. The owning MGR_ID and current MGR_ID are the same.

A foreign-owned domain is a CSDB domain that is owned by a Configuration Server other than the current one, and which is present as the result of a DCS synchronization. The owning MGR_ID and current MGR_ID are not the same.

Domain Naming Considerations

To minimize the likelihood of synchronization problems, consider the following points when creating domain names and configuring synchronizations.

- A Configuration Server cannot contain two domains with the same name.
- A Configuration Server cannot obtain one of its self-owned domains from a Configuration Server that foreign-owns the domain. For example, MGR_001 cannot receive from another Configuration Server any domain for which it (MGR_001) is listed as the owning MGR_ID.

Note: The version that is resident at the owner is always considered the current and correct copy. Its contents will always supersede and replace any changes introduced by other Configuration Servers.

One Owner vs. Multiple Owners

When planning domain ownership, it is helpful to consider whether to assign the proprietorship of all the domains to one Configuration Server, thereby centralizing control; or to disperse control by establishing domain ownership at several Configuration Servers at various, strategic points across the enterprise.

The tables in this section detail the advantages and disadvantages of each method. For additional planning considerations, see "Distributed Configuration Server Planning" (on page 15).

One Owning Configuration Server

The following table lists the benefits and drawbacks of one Configuration Server owning all the domains.

One Domain-Owning Configuration Server

Advantages	Disadvantages
Control of all applications, access rules, and users	Central control might make the database very large, depending on the organization and structure
One Configuration Server Database to backup	Central control might make the database very large, depending on the organization and structure
Data flow throughout the environment is one-way	Data flow throughout DCS is one-way
Aligns with highly centralized organizations	

Multiple Owning Configuration Servers

The following table lists the benefits and drawbacks of domain ownership being assigned to multiple Configuration Servers.

Multiple Domain-Owning Configuration Servers

Advantages	Disadvantages
Aligns readily with highly de-centralized organizations	Does not align well with highly centralized organizations
Databases are smaller and indicative of regional Source Configuration Servers	Multiple Configuration Servers must be administered and backed-up
Applications and users can be managed locally	Allows for two-way data flow, adding complexity to the DCS design
Corporate or common information can be	

Advantages	Disadvantages
managed centrally, while local information is managed locally	
Allows for two-way data flow between central and local Configuration Servers	

Note: Any Configuration Server with self-owned domains should be backed up. Foreign-owned domains can always be obtained through synchronization with the owning Configuration Server.

Domain Eligibility

The list of domains that are eligible for synchronization is dynamically compiled by Distributed Configuration Server. This list is based on the chosen synchronization pair and:

- The database control information concerning the most recent synchronization for the synchronization pair, or
- The most recent update with administrative components (such as **HP Client Automation Administrator Configuration Server Database Editor (Admin CSDB Editor)**, **HP Client Automation Administrator Packager (Packager)**, and **HP Client Automation Administrator Publisher (Publisher)**).

Only domains that have the same owner (on the Source and Destination) can be synchronized between that pair of Configuration Servers.

Selecting Domains

It is not necessary to synchronize all eligible domains between two DCS-enabled Configuration Servers. At the start of each session, an administrator can specify which of the eligible domains are to be synchronized.

Domain Eligibility Rules

The primary DCS domain synchronization eligibility rules are listed below. These apply to each domain independently. See "Log Error Messages" (on page 30).

- Synchronization cannot occur into a self-owned domain.
- There is no replication into an owning Configuration Server.
  Note: If a self-owned domain is deleted, it must be restored from a backup; it cannot be replicated from a DCS-enabled Configuration Server on which it is foreign-owned.
- Domains that are not owned by the same Configuration Server are considered *unrelated*. A domain must be owned by the same MGR_ID at the Source and Destination in order to be eligible for synchronization.
- Once a foreign-owned domain is locally updated with another Client Automation component, it cannot be used as the Source in a *peer synchronization*.
  Note: A local update occurs when a CSDB is updated by a Client Automation component (such as Admin CSDB Editor, Admin Publisher, and Admin Packager) other than Distributed Configuration Server.

- When it is possible to make such a distinction, the DCS will prevent the regression of a more current Destination by a less current peer Source. If the Destination domain has been locally updated, and the relative currency cannot be determined, the synchronization is allowed.

Configuring Distributed Configuration Server

Distributed Configuration Server functionality must be configured for two Configuration Servers.

DCS requires a communications connection between the Source and Destination Configuration Servers.

Distributed Configuration Server: Batch Mode

The command-line mode (also known as the **batch** mode) of DCS is invoked by the executable, `DMABATCH.EXE`. Once the synchronization is started, it will execute with no administrator action required. This is discussed in more detail in "Distributed Configuration Server Options and DMABATCH" (on page 20).

Synchronization Logs

When a synchronization is executed, logs and objects are created. Each subsequent run causes its predecessor’s logs to be overwritten, so that these logs and objects represent the most recent DCS synchronization.

Simultaneous Synchronizations

A Configuration Server can be simultaneously involved in multiple synchronizations in which it is the Source only. This is possible because a Source database is only being read from, whereas a Destination database is being written to.

- A Configuration Server cannot simultaneously be a Source and Destination for different synchronizations.
- A Configuration Server cannot be the Destination in multiple, simultaneous synchronizations.

Distributed Configuration Server Planning

This section offers planning considerations when Distributed Configuration Server is being implemented within a Configuration Server environment.

When to Use Distributed Configuration Server

The following is a list of situations that might arise in a software management enterprise, and in which the capabilities of DCS would prove beneficial.

- To replicate CSDB contents across an enterprise.
- When moving domains from a test environment to a production environment.
- As an alternative to local connects.

Developing a viable, functional DCS infrastructure requires knowledge of:

- The Client Automation resolution process within an environment,
- The hardware and communications configuration of an environment, and
- The Client Automation-managed information within an infrastructure.

The EDMPROF File and DCS Security

The EDMPROF File

The `edmprof` file is the text file in which the operational parameters of the Configuration Server are configured and stored. Two of its sections—**MGR_STARTUP** and **MGR_DMA**—are integral to enabling DCS and ensuring its proper operation.

Information on these `edmprof` sections, including their settings, acceptable values, and impact on DCS processing is presented in this section.

Note: For a comprehensive look at the `edmprof` file, see the *HP Client Automation Enterprise Configuration Server Reference Guide*.

Editing the EDMPROF File

The `edmprof` file can be edited in a standard text-editing application.

Caution: Be sure to review the important information in this section before editing this file. Failure to do so could adversely affect your HPCA environment.

HP recommends backing up the `edmprof` file prior to editing it.

- You must use a UTF8-aware text editor when editing the `edmprof` file.
- Be sure to select `UTF-8` as the *encoding type* when saving the file.
  - Windows users are advised to use **Notepad**.

Failure to use a recommended text-editing application and saving the file as recommended in this section could result in the file’s changes not being correctly applied to the Configuration Server. This could adversely affect your HPCA environment.

MGR_STARTUP Section

The MGR_STARTUP section dictates startup behavior for the Configuration Server. The following table describes the MGR_STARTUP settings are essential to the operation of the DCS.

MGR_STARTUP Settings and Values

Setting	Explanation
MANAGER_TYPE	In order to ensure that a Configuration Server is DCS-enabled, verify that this value is set to **DISTRIBUTED** (the default) or **SERVER**. For additional information, see "Configuration Server Eligibility" (on page 11).
MGR_TYPE	A 32-alphanumeric character (max.) Configuration Server identifier.
MGR_ID	The unique, 3-digit, hexadecimal ID for a Configuration Server. • DCS uses this value to generate object IDs in the CSDB. • Each character in this identifier can have the values 0-9 and A-F. Exception: The 256 consecutive positions from F00 through FFF are reserved for use with Client Automation.
TCP_PORT	The port on which the Configuration Server will listen.

Setting	Explanation
	This must match the port that is specified for DCS communications (`-master-port` in `dmabatch.rc`).

MGR_ID

The MGR_ID setting establishes a unique identity for each Configuration Server. All Domains in a Configuration Server Database are *owned* by a Configuration Server—identified by the value of MGR_ID.

Note: It is possible that a Domain is not owned by the Configuration Server that is hosting its database.

DCS uses the value of MGR_ID to determine which Configuration Server owns each Domain.

Domain ownership is important because in order for a Domain to be eligible for synchronization its owning MGR_ID must be the same on the Source and Destination Configuration Servers. If the owning MGR_IDs do not match, synchronization cannot occur.

Note: Although the MGR_ID must match for both Domains, it is possible that neither the Source nor the Destination is the owner. For more information, see Foreign-Owned Domain in table "Distributed Configuration Server Terminology".

MGR_DMA Section

In addition to the MGR_DMA settings that are needed to establish DCS password protection (described in "Configuration Server Security Settings" (on page 18)), there is another DCS-related setting, DMA_TIMEOUT, which is detailed in the following table.

MGR_DMA Settings and Values

Setting	Explanation of Value
SECURITY_ METHOD	Optional. If not specified, security verification is disabled. To enable native operating system security, specify **EDMSIGN**.
ADMIN_ LIST	This setting is required if a SECURITY_METHOD is specified. Specify the list of administrators (user IDs) that are allowed to use DCS on this CSDB. The format is a comma-separated (no spaces), case-sensitive list of operating system account names.
DMA_ TIMEOUT	Specify the number of seconds that DCS is to wait for non-DCS tasks to complete before applying a lock to the CSDB. If DCS times out before the task ends, it will abort. The default is **0**. • When soft-locking the CSDB, DCS must wait for all administrator tasks to end. • When hard-locking the CSDB, DCS waits for all non-DCS tasks to end.

Setting up Security

Distributed Configuration Server has an optional security feature that enables an administrator to assign password protection to one or both of the synchronization pair's Configuration Server Databases, using native operating system security.

Native Operating-System Security

This section details the assignment of password protection to the native operating system.

A special user ID and password is used to access secured CSDBs. DCS defines only one user ID and password. Therefore, all secured CSDBs that DCS might access must:

- Be defined in the `edmprof` files of their host's security system,
- Have the user ID in the ADMIN_LIST section of their `edmprof` files, and
- Have the same password for that user ID.

Note: The user ID and password values are defined in the `-userid` and `-password` options of the configuration file.

Configuration Server Security Settings

In addition to the steps outlined in "Native Operating-System Security" (on page 18), the MGR_DMA section must be added to the `edmprof` file, as described in this section.

Note: The MGR_DMA section is not included in the `edmprof` file after the Configuration Server installation because it is not needed for default operations. It can be added to the `edmprof` file in order to configure DCS as a default function of the Configuration Server.

To modify the `edmprof` file:

1. Stop the Configuration Server.
2. Open the `edmprof` file with a UTF8-aware text editor.
3. Add the section, MGR_DMA, and the settings shown below:
   ```
   [MGR_DMA]
   SECURITY_METHOD = EDMSIGNR
   ADMIN_LIST = list_of_administrators
   ```
   For a description of these settings, see table "MGR_DMA Settings and Values".
4. Save the changes in UTF8-format, close the `edmprof` file, and restart the Configuration Server.
   Note: The administrators that are specified for ADMIN_LIST must have user rights under local policy settings on the host operating system. For information on establishing operating system-specific user rights and policies, consult the operating system's product documentation.

Distributed Configuration Server Options and DMABATCH

Distributed Configuration Server Configuration Files

The Distributed Configuration Server (DCS) uses two configuration files. The `dcs.cfg` file contains settings for the DCS Source. The "dmabatch.rc" file contains settings for the DCS Destination.

dcs.cfg

This section details the `SNDBUF` parameter, which can be added to the DCS Source configuration file, `dcs.cfg`. To add this option, open `dcs.cfg` in a text editor.

- `SNDBUF` is used to change the DCS network send buffer size (the default is 32K). If you notice slow DCS download or data transfer speeds, you can set this parameter to 64K. For example:

```
dcs::init  {
 DBPATH "C:/Program Files/Hewlett-Packard/CM/ConfigurationServer/DB"

 SNDBUF  64K
}
```

- If the TCP window scaling feature is enabled with the TCP window size set to greater than 64K, `SNDBUF` can be increased accordingly (for example: 64K, 128K, or 254K). You can select the size that provides the best performance.

dmabatch.rc

The table dmabatch.rc Options describes the settings specified in the DCS Destination configuration file, `dmabatch.rc`. To edit these settings, open `dmabatch.rc` in a text editor.

- All of the options that are important to the basic operation of DCS are populated by values that were specified during the DCS Destination installation programs (see the Set by Install column in the table dmabatch.rc Options). You only need to modify an option when a non-default value is preferred.

In the table dmabatch.rc Options, the following DCS terminology is used.

- IS = Integration Server
- Master = the Source Configuration Server
- Slave = the Destination Configuration Server

dmabatch.rc Options

Option	Default Value or Required	Set by Install	Description
-master-host	Required	Y	IP name/address of the master Configuration Server
-master-id	Required	Y	3 hexadecimal-digit ID of the master Configuration Server

Option	Default Value or Required	Set by Install	Description
-master-port	Required	Y	TCP port of the master Configuration Server (usually 3464)
-master-ssl-port	444	Y	SSL port of the master Configuration Server (usually 444)
-master-timeout	3600	Y	Master Configuration Server request timeout (in seconds)
-slave-host	Required	Y	IP name/address of the slave Configuration Server (local computer name or *localhost*)
-slave-id	Required	Y	3 hexadecimal-digit ID of the slave Configuration Server
-slave-port	Required	Y	TCP port of the slave Configuration Server (usually 3464)
-slave-ssl-port	444	Y	SSL port of the slave Configuration Server (usually 444)
-slave-timeout	3600	Y	Slave Configuration Server request timeout (in seconds)
-http-port	3466	Y	-http-port 3466 Y TCP port of the IS (usually 3466)
-https-port	443	Y	SSL port of the IS (often 443; must not be the same as that of `-master-ssl-port`)
-domains	“ ” (null)	Y	Quoted list of blank-separated CSDB Domains; null list means “ALL DOMAINS”
-ssl	0	Y	Enable SSL `1`=enable SSL; `0`=disable
-userid	DMABATCH	Y	User ID to be used for Configuration Server and/or Integration Server authentication
-password_ cipher	AES	Y	TP password-encryption method. Valid values are AES and DES.
-password	<none>	Y	Password to be used for Configuration Server and/or Integration Server authentication. Note: This value can be AES-encrypted or cleartext; it must be consistent with the -password_cipher setting.
-dataless	0	Y	Skip resources `1`=skip resources; `0`=include resources
-loglvl	3	Y	Logging level `2`=terse, `3`=normal, `4`=debug, `5`=debug+

Option	Default Value or Required	Set by Install	Description
-logfile	dmabatch.log	N	Log file name (no pathing)
-logmode	w	N	Overwrite the log file `w`=overwrite log file; `a`=append the log file
-loglines	100000	Y	Number of lines to write to the log file before the log rolls over
-logerr	1	Y	Echo log to `stderr` `1`=echo; `0`=don't echo
-logpath	<none>	N	Override LOGPATH in `nvd.ini` (location of `dmabatch.log`)
-libpath	<none>	N	Override LIBPATH in `nvd.ini` (location of client objects)
-commit	1	N	Commit database updates `1`=commit; `0`=don't commit, but remain locked See also "Deferred Commit" (on page 27).
-report	0	N	Send status-reporting objects to master Configuration Server `1`=send; `0`=don't send
-dmastats-userid	DMA_ <masterid>_ <slaveid>_ <domains>	N	Value of "ZUSERID" (on page 26) in reporting object, when `-report 1`. Note: For use with the Configuration Server method ZPUTPROF.
-reset	0	N	Reset state and unlock Configuration Server on error `1`=reset state and unlock CS's on error; `0`=leave locked on error
-lock-to	FAIL	N	Action on pending hard-lock timeout: `FAIL`, `RETRY`, or `FORCE` Note: `FORCE` will kill all non-DCS tasks on slave Configuration Server.
-download	1	N	A (`1`|`0`) toggle that allows you to review the database differences before proceeding with the synchronization. For a more detailed description, see the section, "Download" (on page 23).

Option	Default Value or Required	Set by Install	Description
-max-deletes	-1	N	A “failsafe” option that can be set in order to prevent the inadvertent deletions of large parts of the database. For more information, see the section, "Maximum Deletions" (on page 23).
-unchanged-domains	1	Y	By default, the time stamps are not checked before synchronization. To improve the performance by checking the time stamps, set the value to 0 in the destination.
-resource-prefix	“ ” (NULL)	N	Set this option to “/RESOURCE” to get data directly from the Apache server on the Core and if the –dataless option is set to 0. This may also improve the performance of downloading data (resources).

Download

- If `1`, DCS will download the resources and synchronize the target database.
- If `0`, processing will halt after the Differencing step. The resources are not downloaded and no metadata is committed. If any differences were detected, both databases will remain locked. Before deciding whether to proceed with the synchronization, check the differences in `zdreport.txt`—to see if they are what was expected.
  - Any subsequent execution of **`dmabatch`** without this option, allows the synchronization to resume using the differences that were already generated.
  - To abort this synchronization and differencing, run `dmabatch action=reset`.

Maximum Deletions

An accidental deletion on the master database could be propagated to all the slave databases. This option limits the maximum number of items that can be deleted in one synchronization at the slave database. If the limit is exceeded in the Differencing step, the process is terminated without changing anything. It’s up to the user to decide the *reasonable* maximum amount of deletions.

- A value of **0** is valid and will prevent all deletions.
- The default of **-1** means no deletion limit.

Configuration Object Equivalents

Some of the options in `dmabatch.rc` have operational equivalents in the configuration objects `ZMANAGER` and `ZMGRSYNC`. The following table lists these equivalents.

Configuration Object Equivalents of dmabatch.rc Options

Option	Set by Install	Configuration Object Equivalent
`-master-host`	Y	ZMANAGER.ZTCPADDR

Option	Set by Install	Configuration Object Equivalent
`-master-id`	Y	ZMGRSYNC.ZSRCMGID
`-master-port`	Y	ZMANAGER.ZTCPPORT
`-master-timeout`	Y	ZMANAGER.ZTIMEO
`-slave-host`	Y	ZMANAGER.ZTCPADDR
`-slave-id`	Y	ZMGRSYNC.ZDSTMGID
`-slave-port`	Y	ZMANAGER.ZTCPPORT
`-slave-timeout`	Y	ZMANAGER.ZTIMEO
`-domains`	Y	ZMGRSYNC.ZDOMAINS
`-userid`	Y	ZMGRSYNC.BATUSER
`-password`	Y	ZMGRSYNC.BATPWD
`-dataless`	Y	ZMGRSYNC.NORES
`-report`	N	ZMGRSYNC.REPORT
`-dmastats-userid`	N	ZMGRSYNC.REPTNAME
`-reset`	N	ZMGRSYNC.BATRESET
`-lock-to`	N	ZMGRSYNC.BATLOKTO

Using PUTPROF

1. In SYSTEM.PROCESS create a new instance, such as DMASTATS.
   - Specify the **Method** attribute as:
     `SYSTEM.ZMETHOD.PUTPROF_DMASTATS`
2. In SYSTEM.ZMETHOD create a new instance, such as PUTPROF_DMASTATS.
   - Specify the **Parameter** attribute as `DMASTATS`
   - Specify the **Method Name** attribute as `EDMMPPRO`

   Note: Each execution of DCS might generate several reporting objects at various points in the processing (see "DMASTATS" (on page 24)). Each of these reporting objects will overwrite the previous one.

DMASTATS

The following table defines the fields of the DMASTATS object.

DMASTATS Fields Defined

Field	Definition
BATCHDAT	Date of this report

Field	Definition
BATCHTIM	Time of this report
BATSTDAT	Date of correlated starting (id=1) report
BATSTTIM	Time of correlated starting (id=1) report
BATCHRC	Character return code (if REPORTID > 1) **Note**: See table "REPORTID Values Defined" for REPORTID values. See table "BATCHRC and BATCHMSG values" for detailed BATCHRC information.
BATCHMSG	Completion message (if REPORTID > 1) **Note**: See "REPORTID Values Defined" for REPORTID values.
BATARGS	DMABATCH command line
BATLKSTA	Result of `DMABATCH ACTION=LOCKSTATUS`: **U** (Unlocked); **S** (Soft-locked); **X** (Exclusive Soft-locked); **H** (Hard-locked)
DMASTATE	**0** = Initial; **1** = Compared; **2** = Downloaded; **3** = Committed
ZSRCMGID	`-master-id` (Source MGR_ID)
ZDSTMGID	`-slave-id` (Destination MGR_ID)
REPORTID	Identifies which DCS processing point sent the report. **Note**: See table "REPORTID values defined" for detailed REPORTID information.
SCOPE	N/A
ZDOMAINS	List of domains
ZUSERID	User name for use with PUTPROF method (see "ZUSERID" (on page 26))

The following table identifies which DMASTATS.REPORTID processing point sent the report.

REPORTID Values Defined

REPORTID	Definition
1	Starting
2	Differencing completed, differences found
3	Differencing completed, no differences found
4	Staging completed
5	Commit completed
6	Ending

Every DMABATCH execution sends REPORTIDs 1 and 6. In addition, synchronizations might send REPORTIDs 3 or 2, 4, and 5 for intermediate status.

The following table lists the Configuration Server BATCHRC and corresponding BATCHMSG responses to DCS requests.

BATCHRC and BATCHMSG Values

BATCHRC	BATCHMSG
000	No differences found. / Successfully Completed
001	dmabatch internal structural error
016	Execution failed (see BATCHMSG for details).
101	Invalid cmdline keyword or keyword combination; master & slave IDs same.
103	Configuration Server ID/host/port not specified (see BATCHMSG for details).
108	No eligible domains.

ZUSERID

If `-dmatstats-userid` was specified in the configuration file (`dmabatch.rc`), ZUSERID uses that value. This name can be:

- A 32-character (maximum) alphanumeric name.
  If it is longer than 32 characters, it will be truncated.
- US national characters, such as **@**, **$**, **#**, and _ are allowed.

If `-dmatstats-userid` was not specified in the configuration file (`dmabatch.rc`), ZUSERID is generated based on one of the following.

- If a synchronization operation:
  DMA_*src-id*_*dest-id*_DOMS_*domains*

  where *domains* is an underscore-separated list of Domains in this synchronization, or ALL if ZDOMAINS=*. For example,

  ```
  DMA_100_203_DOMS_SOFTWARE_POLICY
  ```

- If a special stand-alone operation (such as ACTION=LOCK):
  DMA_<*target_id*>_<*action*>

  ```
  DMABATCH ACTION=LOCKSTATUS MGRID=123
  ```

  generates

  ```
  DMA_123_LOCKSTATUS
  ```

  Note: *target_id* can be independent of *src_id* and *dst_id*.

DMABATCH Command-line Options

Reset

Normally, if synchronization fails during the Staging phase (for example, due to a lost connection), it is left in a state that ensures that it can subsequently be restarted from the point of failure. This entails leaving both Configuration Servers locked. If leaving both Configuration Servers locked, pending a restart, is not acceptable, this option allows the session to be reset to the initial state.

- To manually reset a failed session, specify:
  DMABATCH ACTION=RESET

This action causes an immediate unlocking of both Configuration Servers; staged resources will immediately be freed. The trade-off is that "ability to restart" is sacrificed, which can be a problem if staging failed near the end of a long process.

Deferred Commit

DCS offers the ability to defer committing the database updates on the Destination Configuration Server to a time when it is less busy. To do this, use the DMABATCH command, COMMIT, as shown below.

- To defer the “commit” to a time when the Destination Configuration Server is less busy, specify:
  `DMABATCH COMMIT=NO`

  All DCS processing will be halted after the Staging phase.

Note: This command is equivalent to –`commit` 0 in `dmabatch.rc`.

If Staging is successful, `BATCHRC=000` and the following message will be returned,

```
ZMGRSYNC.BATCHMSG="Commit bypassed by COMMIT=NO"
```

At this point the:

- Source Configuration Server will be unlocked,
- Destination Configuration Server will be soft-locked.

To commit the database updates to the Destination Configuration Server, the Commit phase must subsequently be done without `COMMIT=NO`.

DMABATCH Scripting Commands

The functions that are described in this section are DMABATCH command-line arguments, intended for use in a script that executes DMABATCH and handles error conditions. These functions are called with the command:

```
DMABATCH ACTION=
```

Note: If no value is specified for **`ACTION`** (as seen above), or if `ACTION=SYNC`, a normal synchronization is done. Any other **`ACTION`** value does the indicated action only, with no synchronization.

DMABATCH Line Commands

This section details the use and functionality of the DMABATCH commands.

DMABATCH ACTION=QUIESCE [CSID=id][QTYPE=TASK|TRANS]

This command puts the Destination in a "quiescent" state, thereby increasing the chance of later obtaining a hard-lock.

- If `CSID=` is omitted, the default is the Destination’s ID.
- `QTYPE=TASK`: prevents any new, non-DCS HPCA agent tasks from starting.
- `QTYPE=TRANS`: same as TASK, but also, for currently running non-DCS HPCA agent tasks, the HPCA agent connection terminates when the HPCA agent sends the next transaction.
- This action would likely be scripted to run before a synchronization, thereby decreasing the likelihood of later having to stop any tasks.

DMABATCH ACTION=RESUME [CSID=id]

This command ends a "quiescent" state on the Destination.

- If `CSID=` is omitted, the default is the Destination’s ID.

DMABATCH ACTION=KILLTASKS [CSID=id]

This command should be used if QUIESCE was not sufficient to clear out other tasks in time for DCS to enter the Commit phase. It terminates all non-DCS HPCA agent tasks on the Destination, allowing a DCS run to obtain a hard-lock and commit the changes.

- If `CSID=` is omitted, the default is the Destination’s ID.
- Use KILLTASKS in a script after a synchronization terminates with `BATCHRC=003` (hard-lock timeout).

DMABATCH ACTION=RESET

This command will reset an incomplete synchronization session to its initial state, cause an immediate unlock of both Configuration Servers, and release staged resources (if any). Any subsequent synchronization of the defined synchronization pair will start from scratch.

- Use RESET in a script after a synchronization fails (BATCHRC not = 000) if it is determined that the synchronization cannot be resumed in a timely manner and the Configuration Servers cannot be left locked.

DMABATCH ACTION=SOFTLOCK [CSID=id]
DMABATCH ACTION=UNLOCK [CSID=id]
DMABATCH ACTION=HARDLOCK [CSID=id]

These commands are intended for use with multiple synchronizations from a shared Source database.

- Soft-locking the Source guarantees that resource data that is generated for Source Domains will be retained in cache instead of being recalculated for each synchronization, thereby increasing performance.
- If `CSID=` is omitted, the default is the Destination’s ID.

- If the HARDLOCK command is run, the specified Configuration Server will be hard-locked, thereby preventing HPCA agents from connecting. Also, any HPCA agent tasks that are running on the Configuration Server are killed.
  The specified Configuration Server must currently be unlocked. The default is the Destination Configuration Server.

  Note: This operation is intended for use in environments in which multiple Configuration Servers share a database—a practice that HP *advises against*.

CSID Value Considerations

- If `CSID=` is omitted, the default is the Destination Configuration Server's ID.
- If `CSID ≠` the Source or Destination Configuration Server (master or slave), the -port and -address of the Configuration Server must be specified.

Results of DMABATCH

The results of a DMABATCH synchronization are found in the *batch-message* variable, BATCHMSG, and the associated *batch return-code* variable, BATCHRC (see table "BATCHRC and BATCHMSG values"), in the ZMGRSYNC object.

- After the Differencing step, a BATCHRC of `000` indicates that no domain differences were found.
  The associated BATCHMSG message is "`No differences found`."
- During the Staging and Commit phases of DCS, the `000` return code indicates that the phase was successful.
  The corresponding BATCHMSG is "`Successfully Completed`."

Troubleshooting the Distributed Configuration Server

Note: If your environment uses Core and Satellite servers, first read the *HP Client Automation Enterprise Edition Getting Started and Concepts Guide* as the troubleshooting information in that guide may override the information in this guide.

Logs to Obtain

For Distributed Configuration Server problems, `dmabatch.log` and the Destination Configuration Server log are needed. Although, it is unlikely that the Source Configuration Server log will be helpful, do not discard it or allow it to be overwritten before contacting HP Technical Support.

Log Error Messages

The five log messages that are shown below correspond to the domain eligibility rules. Depending on the circumstances of the failed synchronization, these messages might be found in `dmabatch.log`.

- The “last synchronization” date refers to the local time on the Destination Configuration Server.

Domain Eligibility Error Messages

Message
`Skipping Domain [`*`domain`*`] because owners at source <`*`src ID`*`> and destination <`*`dest ID`*`> do not match`
`Skipping Domain [`*`domain`*`] because cannot synchronize to owner`
`Skipping Domain [`*`domain`*`] because owners at source <`*`src ID`*`> and destination <`*`dest ID`*`> do not match`
`Skipping Domain [`*`domain`*`] because non-authoritative replica: updated (<`*`date`*`> <`*`time`*`>) since last synchronization (<`*`date`*`> <`*`time`*`>)`
`Skipping Domain [`*`domain`*`] because possible DB regression – destination replica (<`*`date`*`> <`*`time`*`) more recent than source (<`*`date`*`> <`*`time`*`>)`

Log and Object Locations

The default locations for the DCS objects and logs are `IDMLIB` and `IDMLOG`, respectively.

Note: The values of these variables can be overridden by the `-libpath` and `-logpath` options in the file, `dmabatch.rc`.

Activating Tracing for DCS

In the MGR_TRACE section of the `edmprof` file, add the following setting and value:

```
DMA=YES
```

Note: The DMA acronym represents *Distributed Manager Adapter*, the original name of the Distributed Configuration Server.

Distributed Configuration Server Objects and Files

Distributed Configuration Server Objects

Obtain the following objects from the default Distributed Configuration Server directory. If DCS is running on a desktop with either an HPCA agent or HPCA Administrator, look in `IDMLIB`.

- ZMANAGER contains the properties of all the Configuration Servers that have been defined to DCS. TP parameters (including TP trace level) are defined here, per Configuration Server.
- ZMGRSYNC contains information about the synchronization pair, including any applicable password information. This object is refreshed when: 1) another synchronization pair is defined and 2) when there is a domain change. The object retains information of saved DCS sessions for subsequent recall.
  The non-TP trace level is determined by ZMGRSYNC.ZTRACEL.

  There are two ZMGRSYNC variables that relate to DMABATCH—BATCHMSG and BATCHRC. These are described in the table "BATCHRC and BATCHMSG values".

  Note: Settings in the `dmabatch.rc` file will always supersede settings in the ZMANAGER and ZMGRSYNC objects.

Distributed Configuration Server Files

This section defines the DCS `.MK`, `.DAT`, and `.IDX` files. Each of these files is preceded by a domain name, as in:

domain`.dat`

.MK

These are *metakit* database files that contain, in a compact and searchable format, a domain's metadata; its instances and class definitions.

- `.MK` files are built on all platforms, on the Source and Destination Configuration Servers.

.DAT

These files cache a domain's *small resource* files. This is a performance feature that minimizes CSDB file operations.

- `.DAT` files are built on Solaris and Windows platforms only; on Source Configuration Servers only.

Note: The amounts of free disk space and the space used by metakits and resource caches, before and after each domain analysis operation, are shown in `dmabatch.log`.

.IDX

These are *index* files for the corresponding .DAT files.

- `.IDX` files are built on Solaris and Windows platforms only; on Source Configuration Servers only.

The EDMAMS Utilities

EDM Access Method Services (EDMAMS) is a set of utilities that can be used to create, delete, copy, change, and list CSDB objects.

Note: For more information on the EDMAMS verbs, see the *HP Client Automation Enterprise Configuration Server Reference Guide*.

The EDMAMS verb "UPDATE_MGRIDS" can be used to update the MGR_ID, Configuration Server name, owning MGR_ID, and owning Configuration Server name.

UPDATE_MGRIDS

- If **DOMAIN** is omitted, all domains are updated.
- All keywords are optional; however, at least one keyword other than **DOMAIN** must be specified.

ZEDMAMS VERB=UPDATE_MGRIDS(,FILE=*file name*)(,DOMAIN=*domain name*)(,CLASS=class name)(,MNAME=*local Configuration Server name*)(,MID=*local Configuration Server ID*)(,MMNAME=*owning Configuration Server name*)(,MMID=*owning Configuration Server ID*)

Domain Eligibility

- *What if no domains are eligible for synchronization?*
  If no domains are presented as eligible for synchronization, make sure that both Configuration Servers were installed as DCS-enabled Configuration Servers. It might be necessary to use the "UPDATE_MGRIDS" verb to rectify this issue.

  Note: For more information about the MANAGER_TYPE setting of the MGR_STARTUP section of the `edmprof` file, see the *HP Client Automation Enterprise Configuration Server Reference Guid*e.

We appreciate your feedback!

If an email client is configured on this system, by default an email window opens when you click here.

If no email client is available, copy the information below to a new message in a web mail client, and then send this message to docfeedback@hp.com.

Product name and version: HP Client Automation Distributed Configuration Server, 8.10

Document title: Reference Guide

Feedback: