



Informatica® Data Replication
9.8.0

Command Line Interface for the Server Manager

Informatica Data Replication Command Line Interface for the Server Manager

9.8.0

December 2018

© Copyright Informatica LLC 2013, 2020

This software and documentation are provided only under a separate license agreement containing restrictions on use and disclosure. No part of this document may be reproduced or transmitted in any form, by any means (electronic, photocopying, recording or otherwise) without prior consent of Informatica LLC.

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation is subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License.

Informatica and the Informatica logo are trademarks or registered trademarks of Informatica LLC in the United States and many jurisdictions throughout the world. A current list of Informatica trademarks is available on the web at <https://www.informatica.com/trademarks.html>. Other company and product names may be trade names or trademarks of their respective owners.

Portions of this software and/or documentation are subject to copyright held by third parties. Required third party notices are included with the product.

The information in this documentation is subject to change without notice. If you find any problems in this documentation, report them to us at infa_documentation@informatica.com.

Informatica products are warranted according to the terms and conditions of the agreements under which they are provided. INFORMATICA PROVIDES THE INFORMATION IN THIS DOCUMENT "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING WITHOUT ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND ANY WARRANTY OR CONDITION OF NON-INFRINGEMENT.

Publication Date: 2020-09-18

Table of Contents

Preface	6
Informatica Resources.	6
Informatica Network.	6
Informatica Knowledge Base.	6
Informatica Documentation.	6
Informatica Product Availability Matrices.	7
Informatica Velocity.	7
Informatica Marketplace.	7
Informatica Global Customer Support.	7
Chapter 1: Overview of the Server Manager Command Line Interface	8
Overview.	8
idrcmd Application.	9
Interactive Mode.	9
Starting the idrcmd Application in Interactive Mode.	9
Non-interactive Mode.	10
Starting the idrcmd Application in Non-interactive Mode.	10
Sample Script for Starting the idrcmd Application in Non-Interactive Mode.	10
Users and Privileges.	11
Chapter 2: Usage Scenario: Using Server Manager CLI Commands to Run Replications	12
Prerequisite Tasks for Running Replication Jobs from the Server Manager CLI.	12
Using the Server Manager CLI to Run Replications.	13
Running a Scheduled Replication Job in Interactive Mode.	13
Running a Scheduled Replication Job in Non-interactive Mode with Scripts.	14
Chapter 3: Command Reference	15
Overview.	15
Connecting to a Server Manager.	16
connect.	16
disconnect.	17
ping.	18
Managing Replication Configurations and the Replication Environment.	19
create config.	19
delete config.	21
download config.	22
export config.	22
export scn.	23
get scn.	24

import config.	25
import scn.	27
set scn.	28
show configs.	29
show processed logs.	29
update config.	30
Managing Server Manager Instances.	31
add server.	31
delete server.	31
show servers.	32
show version.	32
Configuring Data Replication from One Source to Multiple Targets.	33
add target.	33
get target list.	34
rename schema.	34
set conflict logging path.	35
set log table schema.	36
set replication type.	36
Managing Replication Tasks.	37
abort task.	38
delete task.	38
get task detail.	39
restore task.	39
show task history.	40
show task list.	41
show tasks.	41
start task.	41
stop task.	42
Managing Replication Schedules.	43
abort schedule.	43
create schedule.	44
delete schedule.	46
edit schedule.	46
show running schedules.	48
show schedule info.	48
show schedules.	49
start schedule.	49
stop schedule.	50
Viewing Execution Logs for Replication Tasks.	50
show logs.	50
start log watcher.	51
Managing Connections.	52

add connection.	52
delete connection.	55
edit connection.	56
show connection info.	59
show connections.	60
Managing Open Transactions.	60
force transaction.	60
show open transactions.	61
Deploying Configurations Locally or to Another Replication Environment.	63
Deployment Considerations.	64
deploy config.	65
redeploy config.	66
Deployment Properties File.	67
Managing Database Supplemental Logging.	71
suplog add.	72
suplog cancel.	74
suplog open.	74
suplog refresh.	74
suplog remove.	75
suplog run.	77
suplog save.	78
suplog show.	78
Managing Change Data Capture for Microsoft SQL Server Sources.	80
get msdb list.	80
get msinstance list.	81
get mstable list.	81
set replication mode.	82
Managing User Notifications.	83
add email subscriber.	83
add snmp subscriber.	84
configure smtp.	86
configure snmp.	87
Auxiliary Commands.	88
exit.	88
help.	88
Index.	90

Preface

Refer to the *Informatica® Data Replication Command Line Interface for the Server Manager* for information about the commands that you can run to manage Data Replication configurations, schedules, and tasks from the command line.

This guide is intended for application programmers who want to integrate Data Replication with third-party applications.

Informatica Resources

Informatica provides you with a range of product resources through the Informatica Network and other online portals. Use the resources to get the most from your Informatica products and solutions and to learn from other Informatica users and subject matter experts.

Informatica Network

The Informatica Network is the gateway to many resources, including the Informatica Knowledge Base and Informatica Global Customer Support. To enter the Informatica Network, visit <https://network.informatica.com>.

As an Informatica Network member, you have the following options:

- Search the Knowledge Base for product resources.
- View product availability information.
- Create and review your support cases.
- Find your local Informatica User Group Network and collaborate with your peers.

Informatica Knowledge Base

Use the Informatica Knowledge Base to find product resources such as how-to articles, best practices, video tutorials, and answers to frequently asked questions.

To search the Knowledge Base, visit <https://search.informatica.com>. If you have questions, comments, or ideas about the Knowledge Base, contact the Informatica Knowledge Base team at KB_Feedback@informatica.com.

Informatica Documentation

Use the Informatica Documentation Portal to explore an extensive library of documentation for current and recent product releases. To explore the Documentation Portal, visit <https://docs.informatica.com>.

If you have questions, comments, or ideas about the product documentation, contact the Informatica Documentation team at infa_documentation@informatica.com.

Informatica Product Availability Matrices

Product Availability Matrices (PAMs) indicate the versions of the operating systems, databases, and types of data sources and targets that a product release supports. You can browse the Informatica PAMs at <https://network.informatica.com/community/informatica-network/product-availability-matrices>.

Informatica Velocity

Informatica Velocity is a collection of tips and best practices developed by Informatica Professional Services and based on real-world experiences from hundreds of data management projects. Informatica Velocity represents the collective knowledge of Informatica consultants who work with organizations around the world to plan, develop, deploy, and maintain successful data management solutions.

You can find Informatica Velocity resources at <http://velocity.informatica.com>. If you have questions, comments, or ideas about Informatica Velocity, contact Informatica Professional Services at ips@informatica.com.

Informatica Marketplace

The Informatica Marketplace is a forum where you can find solutions that extend and enhance your Informatica implementations. Leverage any of the hundreds of solutions from Informatica developers and partners on the Marketplace to improve your productivity and speed up time to implementation on your projects. You can find the Informatica Marketplace at <https://marketplace.informatica.com>.

Informatica Global Customer Support

You can contact a Global Support Center by telephone or through the Informatica Network.

To find your local Informatica Global Customer Support telephone number, visit the Informatica website at the following link:

<https://www.informatica.com/services-and-training/customer-success-services/contact-us.html>.

To find online support resources on the Informatica Network, visit <https://network.informatica.com> and select the eSupport option.

CHAPTER 1

Overview of the Server Manager Command Line Interface

This chapter includes the following topics:

- [Overview, 8](#)
- [idrcmd Application, 9](#)
- [Interactive Mode, 9](#)
- [Non-interactive Mode, 10](#)
- [Users and Privileges, 11](#)

Overview

You can use the Server Manager Command Line Interface (CLI) commands to manage, run, and monitor replication tasks and schedules in a Data Replication distributed environment with the Server Manager component. With the CLI, you enter commands from command prompts and windows or other text-based consoles such as shells.

Use the Server Manager CLI commands to perform the following tasks:

- Connect to, disconnect from, or ping a Server Manager
- Manage replication configurations and Sync Point values
- View database logs that have been processed for a configuration
- List Server Manager instances and manage subservers
- Configure data replication from one source to multiple targets
- Manage replication tasks and schedules
- View the execution logs for replication tasks
- View connection information
- Manage open transactions
- Deploy configurations locally or to another replication environment
- Manage database supplemental logging for all source types
- Configure and manage user notifications
- Get help information about the Server Manager CLI commands
- Exit the Server Manager CLI

idrcmd Application

Data Replication uses the idrcmd application to execute Server Manager commands. You can run the idrcmd application on the source system, on the target system, or on a standalone system that can access the Server Manager system.

To start the idrcmd application, run one of the following scripts that Data Replication provides in the top-level installation directory:

- On Windows, run idrcmd.bat.
- On Linux and UNIX, run idrcmd.sh.

You can run the idrcmd application in interactive or non-interactive mode:

- In interactive mode, you invoke the idrcmd application once to open an IDRCMD prompt from which you enter commands in text form.
- In non-interactive mode, you invoke the idrcmd application each time you want to run a single command. You enter a command as a command line parameter for the idrcmd application.

Interactive Mode

Use interactive mode if you want to manage replication jobs from a system that does not have a graphical environment.

To use interactive mode, start the idrcmd application to open an IDRCMD prompt. Then enter Server Manager commands at the prompt. The idrcmd application executes the commands.

After you enter the connect command to connect to the Server Manager, the Server Manager CLI establishes a Server Manager session and stores the session state in the idrcmd application memory. All subsequent commands are executed in the context of this session. After you disconnect from the Server Manager or quit from the IDRCMD prompt, the Server Manager CLI closes the Server Manager session.

Starting the idrcmd Application in Interactive Mode

To use the Server Manager Command Line Interface in interactive mode, start the idrcmd application without command-line arguments.

1. Use the cd command to change to the Data Replication installation directory.

```
cd DataReplication_installation
```

2. Run the idrcmd application.

- On Windows, run the following script:

```
idrcmd.bat
```

- On Linux and UNIX, run the following script:

```
./idrcmd.sh
```

3. Enter Server Manager commands at the command prompt:

```
command [parameters]
```

Non-interactive Mode

Use non-interactive mode to use scripts to integrate Server Manager functionality with third-party software or to automate the execution of replication jobs in a distributed environment.

To use non-interactive mode, invoke the `idrcmd` application with a single Server Manager command as the application argument. Consider the following requirements for running the `idrcmd` application in non-interactive mode:

- On Windows, to invoke the `idrcmd` application from a batch file, use the call statement.
- On Linux and UNIX, to invoke the `idrcmd` application within the existing shell, use the source command.

After you enter the connect command to connect the Server Manager, the Server Manager CLI establishes a Server Manager session and stores the session state in the `DataReplication_installation/state.dat` file. All subsequent invocations of the `idrcmd` application are executed in the context of this session. After you disconnect from the Server Manager, the Server Manager CLI closes the session and removes information from the session state file.

Starting the `idrcmd` Application in Non-interactive Mode

To use the Server Manager Command Line Interface in non-interactive mode, append a single command as an argument to the `idrcmd` application.

1. Use the `cd` command to change to the Data Replication installation directory.

```
cd DataReplication_installation
```

2. Run the `idrcmd` application with a single command as the argument.

- On Windows, run the following script:

```
idrcmd.bat command
```

- On Linux and UNIX, run the following script:

```
source ./idrcmd command
```

Sample Script for Starting the `idrcmd` Application in Non-Interactive Mode

Use this sample script as a basis for starting the `idrcmd` application in non-interactive mode.

Copy and customize the sample script for your Server Manager instance. Replace variables with the values that reflect your environment.

Create a text file that contains the Server Manager CLI commands that you want to run and name the text file using the following format: `file_name.txt`. The following example text file includes Server Manager CLI commands that edit several configuration schedules:

```
connect host port login=user_name passwd=  
edit schedule schedule_name1 enable=y et=end_time  
edit schedule schedule_name2 enable=y et=end_time  
edit schedule schedule_name3 enable=y et=end_time  
edit schedule schedule_name4 enable=y et=end_time  
disconnect  
exit
```

Use the following command to start the `idrcmd` application using the `idrcmd.sh` utility and to run the Server Manager CLI commands in the text file:

```
./idrcmd.sh < file_name.txt
```

Users and Privileges

Data Replication provides two types of user accounts: the idradmin user account and regular replication user accounts. Initially, you must connect to the Server Manager as the idradmin user with a blank password. After you connect to the Server Manager Main server the first time, you can change the password for the idradmin user and create other user accounts.

Data Replication uses an ownership mechanism to determine user privileges. Ownership pertains to the following replication objects:

- Configurations
- Schedules
- Tasks
- Connections
- Environment variable lists
- Subscribers

When you create a replication object, you specify an owner for it.

Note: Object ownership completely determines user permissions. Neither the idradmin user nor a regular replication user can customize user permissions in the Data Replication Console.

The idradmin user has all rights and permissions. The idradmin user can modify all of the replication objects and start or stop all of the replication tasks and schedules. The idradmin user can also perform the following administrative tasks:

- Create and manage user accounts.
- Add subservers to a replication server group.
- Edit the Server Manager settings.
- View execution logs of replication tasks.
- View database connections.
- Manage open transactions.
- Deploy configurations.
- Manage database supplemental logging.
- Manage user notifications.

If you connect to a Main server as a regular replication user, you can perform only the following tasks:

- Modify only the replication objects for which you are designated as the owner.
- Start and stop the replication tasks and schedules for which you are designated as the owner.
- Run replication jobs only for the configurations for which you are designated as the owner.
- View execution logs of replication tasks for which you are designated as the owner.
- View database connections for the configurations for which you are designated as the owner.
- Manage open transactions for the configurations for which you are designated as the owner.
- Deploy the configurations for which you are designated as the owner.
- Manage database supplemental logging for the configurations for which you are designated as the owner.

CHAPTER 2

Usage Scenario: Using Server Manager CLI Commands to Run Replications

This chapter includes the following topics:

- [Prerequisite Tasks for Running Replication Jobs from the Server Manager CLI, 12](#)
- [Using the Server Manager CLI to Run Replications, 13](#)

Prerequisite Tasks for Running Replication Jobs from the Server Manager CLI

Before you can use the Server Manager Command Line Interface (CLI) to run replication jobs, you must create a replication configuration by using the Data Replication Console. Also, define all of the Server Manager subservers by using the Data Replication Console.

Optionally, you can create a replication schedule for the configuration in the Data Replication Console if you do not want to create it later from the Server Manager CLI.

1. If you use Server Manager subservers to run replication tasks, define all of the subservers on the **Server Manager** tab > **Servers** view in the Data Replication Console.
2. Create a .db replication configuration in the Data Replication Console and save it to the SQLite database on the Server Manager Main server. When you connect to the Server Manager Main server later from the idrcmd application, use the show configs command to verify that the configuration exists on the server.

For more information about using the Data Replication Console, see the *Informatica Data Replication User Guide*.

3. Optionally, create a schedule for the configuration on the **Server Manager** tab > **Schedules** view in the Data Replication Console.

Alternatively, use the Server Manager CLI to create a schedule.

Using the Server Manager CLI to Run Replications

You must perform several steps to run replication jobs from the command line. The specific steps depend on whether you use the CLI in interactive mode or non-interactive mode.

Running a Scheduled Replication Job in Interactive Mode

Use this procedure to run data replication jobs from the Server Manager Command Line Interface in interactive mode.

Before you begin, create the replication configuration in the Data Replication Console. For more information, see the *Data Replication User Guide*.

1. Start the idradadmin application.

For example, on Windows, run the following commands:

```
cd DataReplication_installation
idrcmd.bat
```

The idrcmd application starts and displays the IDRCMD prompt.

2. At the prompt, enter the ping command to ensure that the Server Manager Main server is available.

For example, enter the following command:

```
ping 192.168.1.11 8088 protocol=http
```

The idrcmd application displays the status of the Main server.

3. If the Server Manager Main server is available, enter the connect command to connect to it.

For example, enter the following command:

```
connect 192.168.1.11 8088 login=demo passwd=demo protocol=http
```

4. To use a replication configuration that you previously created, perform one of the following actions:

- If you created a .db configuration file and saved it to the SQLite database on the Server Manager Main server, enter the show configs command to verify that the configuration exists in the database.
- If you created an .xml configuration file, use the import config command to import it for Server Manager use.

5. If you did not previously create a schedule for the replication configuration, use the create schedule command to create the schedule.

For example, enter the following command to create a continuous schedule:

```
create schedule Ora2Ora_schedule continuous Ora2Ora_config owner=idradadmin
sd=01.04.2013 st=10:00:0
0 ed 01.04.2014 et=23:59:00 days=all lst=7
```

6. To start the replication schedule, enter the start schedule command.

For example, enter the following command:

```
start schedule Ora2Ora_schedule
```

To view the status of the replication tasks in the schedule, use the show schedule info command.

7. To stop the replication schedule, enter the stop schedule command.

For example, enter the following command:

```
stop schedule Ora2Ora_schedule
```

8. After the replication schedule stops, disconnect from the Server Manager Main server to close the session.

Either enter the disconnect command or close the console.

Running a Scheduled Replication Job in Non-interactive Mode with Scripts

You can invoke Server Manager CLI commands for starting and stopping replication schedules from scripts and then call the scripts from third-party software. The scripts call the `idrcmd` application in non-interactive mode.

Before you begin, create the replication configuration in the Data Replication Console. For more information, see the *Data Replication User Guide*. Also, create the replication schedule for the configuration in the Data Replication Console or by using the `create schedule` command for the Server Manager CLI.

1. In the *DataReplication_installation* directory, create a script file for starting a replication job:

- On Windows, create a batch file.
- On Linux and UNIX, create a shell script.

In the script, include the commands for connecting to the Main server and disconnecting from the Main server.

For example, create a script that connects to the Server Manager Main server, starts a schedule, and then disconnects the CLI session.

- On Windows, create the following batch file:

```
call idrcmd.bat connect 127.0.0.1 8088 login=user_name passwd=user_password
call idrcmd.bat start schedule schedule_name
call idrcmd.bat disconnect
```

- On Linux or UNIX, create the following shell script:

```
#!/bin/sh
source ./idrcmd.sh connect 127.0.0.1 8088 login=user_name passwd=user_password
./idrcmd.sh start schedule schedule_name
./idrcmd.sh disconnect
```

2. In the *DataReplication_installation* directory, create a script file for stopping a replication job.

In the script, include commands for connecting to the Main server and disconnecting from the Main server.

For example, create a script that connects to the Server Manager Main server, stops the schedule, and then disconnects the CLI session:

- On Windows, create the following batch file:

```
call idrcmd.bat connect 127.0.0.1 8088 login=user_name passwd=user_password
call idrcmd.bat stop schedule schedule_name
call idrcmd.bat disconnect
```

- On Linux or UNIX, create the following shell script:

```
#!/bin/sh
source ./idrcmd.sh connect 127.0.0.1 8088 login=user_name passwd=user_password
./idrcmd.sh stop schedule schedule_name
./idrcmd.sh disconnect
```

3. Call the script files for starting and stopping replication schedules from your third-party software.

The replication jobs are then invoked when your third-party software runs.

Note: On Linux and UNIX, use the `source` command to run the scripts.

CHAPTER 3

Command Reference

This chapter includes the following topics:

- [Overview, 15](#)
- [Connecting to a Server Manager, 16](#)
- [Managing Replication Configurations and the Replication Environment, 19](#)
- [Managing Server Manager Instances, 31](#)
- [Configuring Data Replication from One Source to Multiple Targets, 33](#)
- [Managing Replication Tasks, 37](#)
- [Managing Replication Schedules, 43](#)
- [Viewing Execution Logs for Replication Tasks, 50](#)
- [Managing Connections, 52](#)
- [Managing Open Transactions, 60](#)
- [Deploying Configurations Locally or to Another Replication Environment, 63](#)
- [Managing Database Supplemental Logging, 71](#)
- [Managing Change Data Capture for Microsoft SQL Server Sources, 80](#)
- [Managing User Notifications, 83](#)
- [Auxiliary Commands, 88](#)

Overview

The Server Manager commands are arranged by general task. For each task, the commands are listed alphabetically. The command descriptions include the command syntax, parameters, and examples.

The syntax uses the following conventions:

- Square brackets [] indicate an optional parameter.
- Curly brackets { } Indicate that you must enter at least one of the items within the brackets.
- A vertical bar (|), also called a *pipe*, between items indicates that you enter only one of the items.

All Server Manager commands are case-sensitive.

Connecting to a Server Manager

Use the Server Manager commands to control Server Manager connections.

Commands are available to perform the following tasks:

- Test if a Server Manager is running at the specified IP address.
- Connect to a Server Manager Main server.
- Disconnect from a Server Manager Main server.
- Clear information about all sessions that were established between the Server Manager Command Line Interface and the Server Managers Main server instances.

connect

Connects to the Server Manager Main server on a specific host.

Important: You must connect to the Server Manager Main server before you can run any other commands.

After you connect to the Server Manager Main server, the idrcmd application stores the session information in the following manner:

- In interactive mode, the idrcmd application stores the session information in the application memory.
- In non-interactive mode, the idrcmd application stores the session information in the state.dat file.

Syntax

```
connect host port login=user_name passwd=user_password [protocol={http|https}]  
[pathToCert=path_to_certificate] [pathToTruststore=path_to_java_truststore]  
[storepass=password_to_java_truststore]
```

Parameters

The following table describes the command parameters:

Parameter	Description
host	The host name or IP address of the system with the Server Manager Main server instance.
port	The port number of the Server Manager Main server instance. Valid values are integers from 1 through 65535.
login	A user name that has the authority to connect to the specified Server Manager Main server instance.
passwd	A valid password for the specified login user. Note: If a password is not required for the login user, enter <code>passwd=</code> , without any spaces or characters after the equals sign.
protocol	The type of protocol that the Server Manager Main Server uses for connections. Valid values are: <ul style="list-style-type: none">• HTTP• HTTPS Default value: HTTP
pathToCert	If the protocol parameter is set to HTTPS, the full path to the cert.pem certificate file.

Parameter	Description
pathToTruststore	If the protocol parameter is set to HTTPS, the full path to the TrustStore file that Java uses to store trusted certificates.
storepass	If the protocol parameter is set to HTTPS, the password for the Java TrustStore.

Example

The following command establishes a connection to the Main server that runs at the specified IP address and port for the idradadmin user with a blank password:

```
connect 192.168.1.11 8088 login=idradadmin passwd=
```

The following command establishes an initial HTTPS connection to the Main server that runs at the specified IP address and port for the demo user:

```
connect 192.168.1.12 8090 login=demo passwd=demo protocol=https
pathToCert=C:\mycert.pem pathToTruststore=C:\Program Files\Java\jre1.8.0_40\lib\security\cacerts
storepass=changeit
```

The following considerations pertain to HTTPS connections to the Main server:

- To add a certificate for the first time, enter the pathToCert, pathToTruststore, and storepass parameters.
- If the certificate is already in the default Java TrustStore directory, you do not need to specify the pathToCert, pathToTruststore, or storepass parameters. You can use the default location of the Java TrustStore file, which is either *JAVA_HOME*/jre/lib/security/cacerts for the JDK or *JAVA_HOME*/lib/security/cacerts for the JRE.
- If the certificate is already in a Java TrustStore directory that is different from the default directory, specify the pathToTruststore parameter the first time you connect to the Main server.

For later connections to the same Main server, you do not need to specify these parameters again.

disconnect

Disconnects from the Server Manager Main server to which you are currently connected.

You can connect to only one Server Manager instance at a time. If you are connected to a Server Manager instance and want to connect to another Server Manager instance, first disconnect from the current Server Manager instance.

After you disconnect from a Server Manager instance, the idrcmd application deletes the session information in the following manner:

- In interactive mode, the idrcmd application deletes the session information from the application memory.
- In non-interactive mode, the idrcmd application deletes the session information from the state.dat file.

Syntax

```
disconnect
```

The following short form is acceptable:

```
dc
```

ping

Tests the availability of a Server Manager Main server or a subserver on a system with a specific host name or IP address. Also verifies user credentials for a Server Manager Main server if you specify the login user name and password in the command.

Syntax

```
ping host or IP address port [protocol={http|https}] [login=user_name]
[passwd=user_password] [pathToCert=path_to_certificate]
[pathToTruststore=path_to_java_truststore] [storepass=password_to_java_truststore]
```

Parameters

The following table describes the command parameters:

Parameter	Description
host	The host name or IP address of the Server Manager Main server or subserver instance.
port	The port number of the Server Manager Main server or subserver instance on the specified host. Valid values are integers from 1 through 65535.
protocol	The type of protocol that the Server Manager Main Server or subserver uses. Valid values are: <ul style="list-style-type: none">• HTTP• HTTPS Default value: HTTP
login	A user name that has the authority to connect to the specified Server Manager Main server instance. Note: You cannot specify a user name for connecting to a subserver.
passwd	A valid password for the specified login user on the Server Manager Main server. Note: You cannot specify a password for connecting to a subserver.
pathToCert	If the protocol parameter is set to HTTPS, the full path to the cert.pem certificate file.
pathToTruststore	If the protocol parameter is set to HTTPS, the full path to the TrustStore file that Java uses to store trusted certificates.
storepass	If the protocol parameter is set to HTTPS, the password for the Java TrustStore.

Examples

The following command pings a Server Manager instance that runs at the specified IP address and port:

```
ping 192.168.1.11 8088
```

The following command pings a Main server that runs at the specified IP address and port under the idradadmin user with no login password:

```
ping 192.168.1.11 8088 login=idradadmin passwd=
```

Note: If a password is not required for the login user, enter `passwd=`, without any spaces or characters after the equals sign.

The following command pings a Server Manager instance that uses an HTTPS connection, runs at the specified IP address and port under the demo user, and also adds a certificate to the Java TrustStore:

```
ping 192.168.1.12 8090 protocol=https
pathToCert=C:\mycert.pem pathToTruststore=C:\Program Files\Java\jre1.8.0_40\lib\security\cacerts
storepass=changeit
```

The following command pings a Server Manager instance that uses an HTTPS connection, runs at the specified IP address and port under the demo user, and that already contains a certificate in the Java TrustStore:

```
ping 192.168.1.12 8090 protocol=https
pathToTruststore=C:\Program Files\Java\jre1.8.0_40\lib\security\cacerts
```

Managing Replication Configurations and the Replication Environment

Use the Server Manager Command Line Interface to work with replication configurations.

Commands are available to perform the following tasks:

- Import an XML configuration that you created on the Server Manager CLI system into the SQLite database that the Server Manager Main server uses.
- Delete a configuration from the SQLite database that the Server Manager uses.
- Download a configuration in XML format from a remote Server Manager instance to a local computer.
- Export a configuration in XML format and SQLite database format.
- Export the Sync Point values for the mapped target tables in a replication configuration in CSV format.
- View the Sync Point values for the mapped target tables in a replication configuration.
- Import an XML configuration file or configuration SQLite database file to the Server Manager Main server to which you are connected.
- Import the Sync Point values for the mapped target tables in a replication configuration in CSV format.
- Set the Sync Point values for the mapped target tables in a replication configuration.
- View available configurations.
- View a list of processed logs for a configuration.
- Update a configuration in the SQLite database from an XML configuration.

create config

Creates a replication configuration on the Server Manager Main server based on an XML configuration file that is on the system where the Server Manager CLI runs. You must have previously created the XML configuration file. After you run this command, you can work with the configuration from the Data Replication Console or Server Manager CLI.

Syntax

You can enter the command with or without parameters. If you run the command without any parameters, the Server Manager CLI prompts you to enter them in interactive mode. If you specify one or more optional parameters in the command, the Server Manager CLI uses these parameter values to update the configuration in non-interactive mode and preserves the values that were last set for the other optional parameters.

To enter the command with its parameters, use the following syntax:

```
create config path_to_XML_configuration_file name=configuration_name owner=user_name
src=source_server dest=target_server srcenv=source_environment_variables_list
destenv=target_environment_variables_list [srcconnection=source_connection]
[destconnection=target_connection]
```

The following short form is acceptable:

```
cc path_to_XML_configuration_file name=configuration_name owner=user_name
src=server_name dest=server_name srcenv=source_environment_variables_list
destenv=target_environment_variables_list [srcconnection=source_connection]
[destconnection=target_connection]
```

Parameters

The following table describes the command parameters:

Parameter	Description
path_to_XML_configuration_file	The full path to the XML configuration file on the local Server Manager CLI system. Note: If the path includes spaces or special characters, enclose the path value with single quotation marks.
name	The configuration name in the SQLite database that the Server Manager uses to store the configuration information. Note: Configuration names are not case-sensitive. They can contain the digits 0-9, Latin letters A-Z and a-z, and the underscore (_) character. The Server Manager CLI truncates configuration names that are longer than 100 characters.
owner	The name of the Server Manager user to assign as the owner for the configuration. To set the current connected user as the configuration owner, enter this parameter without a value, as follows: owner=
src	The name of the source Server Manager instance. To set the Main Server Manager as the source Server Manager, enter this parameter without a value, as follows: src=
dest	The name of the target Server Manager instance. To set the Main Server Manager as the target Server Manager, enter this parameter without a value, as follows: dest=
srcenv	The name of the environment variables list for the source Server Manager. To use the system environment variables, enter this parameter without a value, as follows: srcenv=
destenv	The name of the environment variables list for the target Server Manager. To use the system environment variables, enter this parameter without a value, as follows: destenv=

Parameter	Description
srcconnection	Optional. The name of the source database connection that you want to import into the SQLite database for the configuration. The specified connection must be defined on the source Server Manager instance. This parameter value overrides the default source connection from the imported XML configuration file. If you omit this parameter and the configuration file does not contain source connection information, the Server Manager CLI returns an error.
destconnection	Optional. The name of the target database connection that you want to import into the SQLite database for the configuration. The specified connection must be defined on the target Server Manager instance. This parameter value overrides the default target connection from the imported XML configuration file. If you omit this parameter and the configuration file does not contain target connection information, the Server Manager CLI returns an error.

Examples

The following command imports the Ora2Ora_local.xml configuration file from the C:\configs directory on the local system as the Ora2Ora configuration. The Ora2Ora configuration is owned by idradmin and uses the Main server as both the source and target Server Manager instance. The Main server uses system environment variables.

```
create config C:\configs\Ora2Ora_local.xml name=Ora2Ora owner=idradmin src=Main
dest=Main srcenv= destenv=
```

The following command displays a series of prompts to enter the parameters:

```
cc
```

delete config

Deletes a replication configuration from the SQLite database that the Server Manager uses.

Important: Before you delete a replication configuration, delete all schedules that use the configuration.

Syntax

```
delete config configuration_name
```

The following short form is acceptable:

```
dconf configuration_name
```

Parameters

The following table describes the command parameter:

Parameter	Description
<i>configuration_name</i>	The name of the configuration to delete.

Example

The following command deletes the Ora2Ora_config configuration:

```
delete config Ora2Ora_config
```

download config

Downloads a replication configuration from the Server Manager Main server and saves the downloaded configuration as an XML configuration file to the local computer where you run this command.

Syntax

Enter the command with both positional parameters:

```
download config configuration_name path_to_XML_configuration_file
```

The following short form is acceptable:

```
dlc configuration_name path_to_XML_configuration_file
```

Parameters

The following table describes the command parameters:

Parameter	Description
<i>path_to_XML_configuration_file</i>	The full path to the downloaded XML configuration file on the local computer. If you do not specify a path, the Server Manager saves the configuration to the current directory. Note: If the path includes spaces or special characters, enclose the path value with single quotation marks.
<i>configuration_name</i>	The name of the replication configuration on the remote Server Manager instance that you want to download.

Example

The following command downloads the Ora2Ora configuration and saves it as the Ora2Ora.xml file in the C:\configs directory.

```
download config Ora2Ora C:\configs\
```

export config

Exports a replication configuration to an XML configuration file and SQLite database file in a specific directory on the Server Manager Main server system.

For example, you can use this command for the following purposes:

- Export configuration files for use in other Data Replication environments and Server Manager systems.
- Create a backup of a configuration in another directory.
- Retain internal replication task processing information that is stored in the configuration SQLite database before importing the configuration back into Data Replication as a copy with a new name.

Syntax

```
export config configuration_name [dest_path_and_directory_for_configuration_file]
```

The following short form is acceptable:

```
ec configuration_name [dest_path_and_directory_for_configuration_file]
```

Parameters

The following table describes the command parameters:

Parameter	Description
<i>configuration_name</i>	The name of the replication configuration to export. Tip: Use the show configs command to list the replication configuration names.
<i>dest_path_and_directory_for_configuration_file</i>	The path and directory to which you are exporting the configuration in XML format and in SQLite database format. You can specify an absolute path or a path that is relative to the <i>DataReplication_installation</i> directory. If you do not specify a path, the Server Manager exports the configuration file to the <i>DataReplication_installation</i> directory. Note: If the path includes spaces or special characters, enclose the path value with single quotation marks.

Examples

The following command exports the configuration named Ora2Ora to the export directory:

```
export config Ora2Ora C:\export
```

The following command exports the Ora2Ora configuration to the *DataReplication_installation* directory:

```
ec Ora2Ora
```

export scn

Exports the Sync Point values for mapped target tables in a replication configuration to a .csv file. You can then override the Sync Point values in the .csv file and import the updated file by using the import scn command.

For DB2 and Microsoft SQL Server sources, the Sync Point value is an LSN. For MySQL sources, the Sync Point value is a log coordinate that represents the binary log file index and offset position. For Oracle sources, the Sync Point value is an SCN.

Syntax

```
export scn configuration_name [file_name] [-l]
```

The following short form is acceptable:

```
escn configuration_name [file_name] [-l]
```

Parameters

The following table describes the command parameters:

Parameter	Description
<i>configuration_name</i>	The name of the configuration for which to export Sync Point values to a .csv file.
<i>file_name</i>	The name of the .csv file to which the comma-separated list of Sync Point values is exported. This command generates this file.
-l	Indicates that the .csv file is located on the computer that runs the Data Replication Console. Omit this option if the file is located on the Server Manager Main server system.

Example

The following example command exports the Sync Point values for the Ora2Ora_config configuration to the Ora2Ora.csv file that is located on the Data Replication Console system:

```
import scn Ora2Ora_config "C:\Ora2Ora.csv" -l
```

Sync Point .CSV File

The .csv file to which you export Sync Point values for mapped target tables has a specific format.

File Format

The file uses the following format:

```
#tgt name: target_connection_name  
#id: DbConnMapID  
object_ID, Sync_Point # "source_schema"."source_table"
```

The following table describes the variables in this file format:

Variable	Description
<i>target_connection_name</i>	The name of the target connection. For configurations that have multiple targets, this file contains a section with Sync Point values for each target connection.
<i>DbConnMapID</i>	The internal unique identifier for the combination of the configuration and target connection.
<i>object_ID</i>	The unique identifier of the source table.
<i>Sync_Point</i>	The Sync Point value of the source table that is identified by the object ID. The Applier uses the Sync Point value to start applying changes to the mapped target table from a particular point in the transaction log.
<i>source_schema</i>	The name of the source schema.
<i>source_table</i>	The name of the source table.

Sample File

The following sample file contains Sync Point values for a configuration that has two targets:

```
#tgt name: ora_tgt_1  
#id: 6  
90222, 169192565862 # "SRC"."TABLE1"  
90229, 169192565863 # "SRC"."TABLE2"  
#tgt name: ora_tgt_2  
#id: 7  
90222, 169192565904 # "SRC"."TABLE1"  
90229, 169192565905 # "SRC"."TABLE2"
```

get scn

Lists the Sync Point values for the mapped target tables in a replication configuration. Use this command with the set scn command to update the Sync Point values in a configuration based on another configuration that maps the same tables.

For DB2 and Microsoft SQL Server sources, the Sync Point value is an LSN. For MySQL sources, the Sync Point value is a log coordinate that represents the binary log file index and offset position. For Oracle sources, the Sync Point value is an SCN.

The output from the get scn command includes a row for each mapped target table. Each row contains an object ID, SCN/log coordinate/LSN value, low SCN/log coordinate/LSN value, and table name.

Use this output to create input for the set scn command.

Syntax

```
get scn configuration_name [connection_name]
```

The following short form is acceptable:

```
gscn configuration_name [connection_name]
```

Parameters

The following table describes the command parameters:

Parameter	Description
configuration_name	The name of the configuration for which to list Sync Point values.
connection_name	The name of the connection to the target database that contains the tables for which to get Sync Point values for the Applier. Include this parameter if you use connections that are defined on the Server Manager tab > Connections view to connect to the target.

Examples

The following example command lists the Sync Point values in the Ora2Ora_config configuration:

```
get scn Ora2Ora_config
```

The following example command lists the Sync Point values for a target that uses the Ora_tgt2 connection in the Ora2Ora_config configuration:

```
gscn Ora2Ora_config Ora_tgt2
```

Sample Output

```
id      scn      scn_low  name
75110   26584040  0        TABLE1
76710   26584040  0        TABLE2
```

import config

Imports information from an XML configuration file or configuration SQLite database file to the Server Manager Main server to which you are connected. The file can be from the local replication environment or from another replication environment. Use this command to create another configuration instance locally or to migrate a configuration to another Main server environment.

For example, you can import a configuration that you created in a test environment for use in the production environment.

Syntax

You can enter the command with or without parameters. If you run the command without any parameters, the Server Manager CLI prompts you to enter them in interactive mode. If you specify one or more optional parameters in the command, the Server Manager CLI uses these parameter values to update the configuration in non-interactive mode and preserves the values that were last set for the other optional parameters.

To enter the command with its parameters, use the following syntax:

```
import config path_to_configuration_file name=configuration_name owner=user_name
src=source_server_name dest=target_server_name srcenv=source_environment_variables_list
```

```
destenv=target_environment_variables_list [srcconnection=source_connection]
[destconnection=target_connection] [needclean=need_clean_option]
```

The following short form is acceptable:

```
ic path_to_configuration_file name=configuration_name owner=user_name src=server_name
dest=server_name srcenv=source_environment_variables_list
destenv=target_environment_variables_list [srcconnection=source_connection]
[destconnection=target_connection] [needclean=need_clean_option]
```

Parameters

The following table describes the command parameters:

Parameter	Description
<i>path_to_configuration_file</i>	The full path to the XML configuration file or configuration SQLite database file to import. Note: If the path includes spaces or special characters, enclose the path value with single quotation marks.
name	The name of the configuration to import. Note: Configuration names are not case-sensitive. They can contain the digits 0-9, Latin letters A-Z and a-z, and the underscore (_) character. The Server Manager CLI truncates configuration names that are longer than 100 characters.
owner	The name of the Server Manager user to assign as the owner for the configuration. To set the current connected user as the configuration owner, enter this parameter without a value, as follows: owner=
src	The name of the source Server Manager instance. To set the Main Server Manager as the source Server Manager, enter this parameter without a value, as follows: src=
dest	The name of the target Server Manager instance. To set the Main Server Manager as the target Server Manager, enter this parameter without a value, as follows: dest=
srcenv	The name of the environment variables list for the source Server Manager. To use the system environment variables, enter this parameter without a value, as follows: srcenv=
destenv	The name of the environment variables list for the target Server Manager. To use the system environment variables, enter this parameter without a value, as follows: destenv=

Parameter	Description
srcconnection	Optional. The name of the source database connection that you want to import into the SQLite database for the configuration. The specified connection must be defined on the source Server Manager instance. This parameter value overrides the default source connection from the imported XML configuration file or configuration SQLite database file. If you omit this parameter and the configuration file does not contain source connection information, the Server Manager CLI returns an error.
destconnection	Optional. The name of the target database connection that you want to import into the SQLite database for the configuration. The specified connection must be defined on the target Server Manager instance. This parameter value overrides the default target connection from the imported XML configuration file or configuration SQLite database file. If you omit this parameter and the configuration file does not contain target connection information, the Server Manager CLI returns an error.
needclean	Optional. Indicates whether the Server Manager cleans the configuration file before importing it into the SQLite database. Valid values are: <ul style="list-style-type: none"> • true • false

Examples

The following command imports the Ora2Ora_local.xml configuration file from the C:\configs directory on the Main server system as the Ora2Ora configuration. The Ora2Ora configuration is owned by idradadmin and uses the Main server as both the source and target Server Manager instance. The Main server uses system environment variables.

```
import config C:\configs\Ora2Ora_local.xml name=Ora2Ora owner=idradadmin src=Main
dest=Main srcenv= destenv=
```

The following command displays a series of prompts to enter the parameters:

```
ic
```

import scn

Imports Sync Point values for mapped target tables from a .csv file into a configuration. The Sync Point values are then available to the Applier for target apply processing. You must have previously exported the Sync Point values for one or more target tables to a .csv file from the Data Replication Console or with the export scn command from the Server Manager CLI.

For DB2 and Microsoft SQL Server sources, the Sync Point value is an LSN. For MySQL sources, the Sync Point value is a log coordinate that represents the binary log file index and offset position. For Oracle sources, the Sync Point value is an SCN.

Important: Before you import Sync Point values, review the .csv file that contains these values. If you do not want to import Sync Point values for some tables, delete the lines that correspond to the tables from the .csv file.

Syntax

```
import scn configuration_name [file_name] [-l]
```

The following short form is acceptable:

```
iscn configuration_name [file_name] [-l]
```

Parameters

The following table describes the command parameters:

Parameter	Description
<i>configuration_name</i>	The name of the configuration to which to import Sync Point values for mapped target tables.
<i>file_name</i>	The name of the .csv file that contains the comma-separated list of Sync Point values to import. This file was generated by the export scn command.
-l	Indicates that the .csv file is located on the computer that runs the Data Replication Console. Omit this option if the file is located on the Server Manager Main server system.

Example

The following example command imports the Sync Point values from the Ora2Ora.csv file that is located on the Data Replication Console system into the Ora2Ora_config configuration:

```
import scn Ora2Ora_config "C:\Ora2Ora.csv" -l
```

set scn

Sets the Sync Point values for the mapped target tables in a replication configuration.

For DB2 and Microsoft SQL Server sources, the Sync Point value is an LSN. For MySQL sources, the Sync Point value is a log coordinate that represents the binary log file index and offset position. For Oracle sources, the Sync Point value is an SCN.

Syntax

```
set scn configuration_name [connection_name] SyncPoint_value_list
```

The following short form is acceptable:

```
sscn configuration_name [connection_name] SyncPoint_value_list
```

Parameters

The following table describes the command parameters:

Parameter	Description
<i>configuration_name</i>	The name of the configuration for which to set the Sync Point values.
<i>connection_name</i>	The name of the connection for the target database that contains the tables for which to set the Sync Point values for the Applier. Include this parameter if you use connections that are defined on the Server Manager tab > Connections view to connect to the target.
<i>SyncPoint_value_list</i>	The list of the Sync Point values for the mapped target tables that the get scn command generated. For example, for Oracle, the list has the following format: 'objectID_1 scn_1 low_scn_1' 'objectID_2 scn_2 low_scn_2'...

Examples

The following example command sets the Sync Point values in the Ora2Ora_config configuration:

```
set scn Ora2Ora_config '74599 26572854 0' '74597 26572854 0' '74586 26572852 0'
```

The following example command sets the Sync Point values for a target that uses the Ora_tgt2 connection in the Ora2Ora_config configuration:

```
sscn Ora2Ora_config Ora_tgt2 '74599 26572854 0' '74597 26572854 0' '74586 26572852 0'
```

show configs

For each Server Manager user, displays a list of replication configurations that are available to the Server Manager.

Syntax

```
show configs
```

The following short form is acceptable:

```
cl
```

show processed logs

Displays a list of the database logs that the Extractor processed for a configuration.

Note: For Microsoft SQL Server sources, the Server Manager CLI displays only backup logs. For Oracle sources, the Server Manager CLI displays both online and archived redo logs.

Syntax

Enter the command with the configuration name:

```
show processed logs configuration_name  
spl configuration_name
```

Parameters

The following table describes the command parameter:

Parameter	Description
<i>configuration_name</i>	The name of the replication configuration for which you want to list the processed logs.

Examples

The following example command displays the list of processed logs for the configuration named oracle2oracle:

```
show processed logs oracle2oracle
```

Sample Output

```
ID Name Thread ID Start time End time  
18 01_MF_1_18_9N4YJJ4F_.ARC 1 Apr 24, 2014 12:24:04 PM Apr 24, 2014 12:28:00 PM  
19 01_MF_1_19_9N5SR9BJ_.ARC 1 Apr 24, 2014 12:24:04 PM Apr 24, 2014 12:28:00 PM  
20 01_MF_1_20_9N5VFXPH_.ARC 1 Apr 24, 2014 12:24:04 PM Apr 24, 2014 12:28:00 PM  
21 01_MF_1_21_9N6HMB5V_.ARC 1 Apr 24, 2014 12:24:04 PM Apr 24, 2014 12:28:00 PM  
22 01_MF_1_22_9N7DQMD7_.ARC 1 Apr 24, 2014 12:24:04 PM Apr 24, 2014 12:28:00 PM  
23 01_MF_1_23_9N7JJ34K_.ARC 1 Apr 24, 2014 12:24:04 PM Apr 24, 2014 12:28:00 PM
```

The following table describes the properties of processed logs:

Property	Description
ID	The sequence number of the processed log file.
Name	The name of the processed log file.
Thread ID	For Oracle sources, the unique identifier of a redo thread. For other sources, the unique identifier of the database.
Start time	The date and time when the Extractor opens an intermediate file that contains records from the processed log file.
End time	The date and time when the Extractor closes an intermediate file that contains records from the processed log file.

update config

Updates a configuration on the Server Manager Main server with information from an XML configuration file.

Syntax

Enter the command with both positional parameters:

```
update config path_to_XML_configuration_file configuration_name
```

The following short form is acceptable:

```
uc path_to_XML_configuration_file configuration_name
```

Parameters

The following table describes the command parameters:

Parameter	Description
<i>path_to_XML_configuration_file</i>	The full path to the XML configuration file from which to import information for updating the configuration in the SQLite database. Note: If the path includes spaces or special characters, enclose the path value with single quotation marks.
<i>configuration_name</i>	The name of the configuration in the SQLite database to update.

Examples

The following command uses the Ora2Ora_local.xml configuration file in the C:\configs_directory to update the Ora2Ora configuration on the Main server:

```
update config C:\configs\Ora2Ora_local.xml Ora2Ora
```

Managing Server Manager Instances

Use the Server Manager Command Line Interface to manage the servers in a replication server group.

Commands are available to perform the following actions:

- Add a Server Manager subserver to a replication server group.
- Delete a Server Manager subserver from a replication server group.
- View available Server Manager instances.
- View the version of the Main server instance and the Server Manager CLI build.

add server

Adds a subserver to the replication server group of the Main server to which you are connected.

Syntax

```
add server [-s] server_name host port
```

The following short form is acceptable:

```
asrv [-s] server_name host port
```

Parameters

The following table describes the command parameters:

Parameter	Description
-s	Indicates which type of protocol the subserver uses. Include this parameter if the subserver uses HTTPS. By default, the subserver uses HTTP.
server_name	The name of the Server Manager subserver to add to the replication server group. Note: Subserver names can contain only the digits 0-9, Latin letters A-Z and a-z, and the underscore (_) character. The Server Manager CLI truncates subserver names that are longer than 100 characters.
host	The host name or IP address of the system where the subserver runs.
port	The port number of the subserver instance. Valid values are integers from 1 through 65535.

Examples

The following command adds a subserver named subserver1 to the replication server group. The subserver runs on the host DemoPC and port 8088 and uses HTTP.

```
add server subserver1 DemoPC 8088
```

delete server

Deletes a subserver from the replication server group of the Main server to which you are connected.

Important: Before you delete a subserver, perform the following prerequisite tasks for each replication configuration that is connected to the subserver that you want to delete:

- On the **Server Manager** tab > **Schedules** view, stop all running tasks. For more information, see the "Scheduling Replication Tasks" chapter of the *Informatica Data Replication User Guide*.

- On the **Server Manager** tab > **Configs** view, change to another source or target subserver. For more information, see the "Scheduling Replication Tasks" chapter of the *Informatica Data Replication User Guide*.
- On the **Server Manager** tab > **Tasks** view, if you created user-defined tasks that run on the subserver that you want to delete, change the subserver on which user-defined tasks run. For more information, see the "Managing Replication Configurations" chapter of the *Informatica Data Replication User Guide*.

Syntax

```
delete server server_name [-f]
```

The following short form is acceptable:

```
delsrv server_name [-f]
```

Parameters

The following table describes the command parameters:

Parameter	Description
server_name	The name of the Server Manager subserver to delete from the replication server group.
-f	Forces the deletion of a subserver that is not available from the Main server.

Examples

The following command deletes a subserver named subserver1 from a replication server group.

```
delete server subserver1
```

show servers

Displays a list of Server Manager instances that includes the Main server to which you are connected and all of the subservers that are associated with this Main server.

Syntax

```
show servers
```

The following short form is acceptable:

```
shsrv
```

show version

Shows the version of the Main server instance to which you are connected and the build number of the Server Manager Command Line Interface.

Syntax

```
show version
```

Sample Output

The following output displays when you enter the show version command and you are not connected to the Server Manager Main server:

```
Server Manager CLI Version 9.6.3 Build 4435661
```


The following output displays when you enter the show version command and you are connected to the Server Manager Main server:

```
Server Manager Version 9.6.3 Build 4435774 Build date Feb 12 2016 15:54:50 - protocol
version 9630
Server Manager CLI Version 9.6.3 Build 4435661
```

Configuring Data Replication from One Source to Multiple Targets

Use the Server Manager Command Line Interface to configure data replication from one source to multiple targets.

Commands are available to perform the following actions:

- Add a secondary target to the configuration.
- View targets for a particular replication configuration.
- Customize Applier settings for a secondary target, including the target schema name, audit log table schema, apply mode, and the path to the log file that the Applier uses to log conflict resolutions.

add target

Adds a secondary target to the replication configuration.

Syntax

```
add target configuration_name Server_Manager_name connection_name
[recovery=recovery_table_name]
```

The following short form is acceptable:

```
atgt configuration_name Server_Manager_name connection_name
[recovery=recovery_table_name]
```

Parameters

The following table describes the command parameters:

Parameter	Description
configuration_name	The name of the configuration for which to add a secondary target.
Server_Manager_name	The name of the Server Manager instance to use for the secondary target. You can specify either the Main server or a subserver.
connection_name	The name of a database connection that you previously defined in the Data Replication Console. The target uses the specified connection to connect to the target database. You must specify a connection for the secondary target that is different from the connection of the primary target.

Parameter	Description
recovery_table_name	A fully qualified name for the recovery table that the Server Manager CLI generates on the secondary target. If you do not specify a recovery table name, the Server Manager CLI does not generate the recovery table on the secondary target. In this case, the Applier looks for the recovery table with the name you specified in the configuration for the primary target. The Applier might end with an error if a table with this name does not exist on the secondary target.

Examples

The following command adds a secondary target to the Ora2Ora_config configuration. The target uses the ora_tgt connection to connect to the target database and uses the Main server to run replication tasks. The Server Manager CLI generates the recovery table named DEMO_TGT.IDR_RECOVERY for the secondary target.

```
add target Ora2Ora_config Main ora_tgt recovery=DEMO_TGT.IDR_RECOVERY
```

get target list

Lists all of the targets that a replication configuration defines, including the primary target and all of the secondary targets.

Syntax

```
get target list configuration_name
```

The following short form is acceptable:

```
gtl configuration_name
```

Parameters

The following table describes the command parameter:

Parameter	Description
configuration_name	The name of the configuration for which to list targets.

Examples

The following command lists all of the targets that the Ora2Ora_config configuration defines:

```
get target list Ora2Ora_config
```

rename schema

Overrides the name of the schema that includes the mapped target tables for a secondary target. Use this command if you want to use a schema name for a secondary target that is different from the schema name of the primary target.

Note: For Microsoft SQL Server targets, the schema name is equivalent to the SQL Server database name.

For example, if you map tables in the schema *Schema1* on a primary target and map similar tables in the schema *Schema2* on a secondary target, you can override *Schema1* with *Schema2* for the secondary target.

Syntax

```
rename schema configuration_name connection_name name_on_primary_target
name_on_secondary_target
```

The following short form is acceptable:

```
rs configuration_name connection_name name_on_primary_target name_on_secondary_target
```

Parameters

The following table describes the command parameters:

Parameter	Description
configuration_name	The name of the configuration that includes the secondary target for which you want to override the schema or database name.
connection_name	The name of the database connection for the secondary target.
name_on_primary_target	The schema name on the primary target that you want to override on the secondary target.
name_on_secondary_target	The schema name that replaces the <i>name_on_primary_target</i> value on the secondary target. Note: For databases that use case-sensitive database object names, enter the schema name in the same case that the database uses. For Oracle unquoted identifiers, use uppercase.

Example

The following command overrides the schema name *Schema1* that is defined on the primary target with the schema name *Schema2* that is defined on the secondary target. The secondary target is defined in the Ora2Ora_config configuration and uses the ora_tgt connection.

```
rename schema Ora2Ora_config ora_tgt Schema1 Schema2
```

set conflict logging path

Overrides the path to the log file that the Applier uses for conflict resolution on a secondary target.

Syntax

```
set conflict logging path configuration_name connection_name path_to_log
```

The following short form is acceptable:

```
sclp configuration_name connection_name path_to_log
```

Parameters

The following table describes the command parameters:

Parameter	Description
configuration_name	The name of the configuration that includes the secondary target for which you want to override the path to the log file for conflict resolution.
connection_name	The name of the database connection for the secondary target.
path_to_log	The path to the log file that the Applier uses for conflict resolution on the secondary target. Note: If the path includes spaces or special characters, enclose the path value with single quotation marks.

Example

The following command specifies the override path of /home/demo/logs for the secondary target that is defined in the Ora2Ora_config configuration and that uses the ora_tgt connection:

```
set conflict logging path Ora2Ora_config ora_tgt /home/demo/logs
```

set log table schema

For the secondary targets that use Merge Apply mode, specifies the name of the schema to use for audit log tables. If you do not specify the audit log schema for a secondary target, Data Replication uses the schema that you specified for the primary target on the **Runtime Settings** tab > **Calculated Columns** view in the Data Replication Console.

Syntax

```
set log table schema configuration_name connection_name schema_name
```

The following short form is acceptable:

```
slts configuration_name connection_name schema_name
```

Parameters

The following table describes the command parameters:

Parameter	Description
configuration_name	The name of the configuration that includes the secondary target for which you want to set the schema name for audit log tables.
connection_name	The name of the database connection for the secondary target.
schema_name	The name of the schema to use for the audit log tables on the secondary target.

Examples

The following command specifies the DEMO_TGT audit log table schema for the secondary target that is defined in the Ora2Ora_config configuration and that uses the ora_tgt connection:

```
set log table schema Ora2Ora_config ora_tgt DEMO_TGT
```

set replication type

Overrides the apply mode for a secondary target.

For example, if you use SQL Apply mode on a primary target, you can override this mode with Audit Apply mode on a secondary target.

Syntax

```
set replication type configuration_name connection_name apply_mode
```

The following short form is acceptable:

```
srt configuration_name connection_name apply_mode
```

Parameters

The following table describes the command parameters:

Parameter	Description
configuration_name	The name of the configuration that includes the secondary target for which you want to override the apply mode.
connection_name	The name of the database connection for the secondary target.
apply_mode	The apply mode to use for the secondary target. Valid values are: <ul style="list-style-type: none">• audit. Use Audit Apply mode.• merge. Use Merge Apply mode.• sql. Use SQL Apply mode.

Example

The following command sets the Audit Apply mode for the secondary target that is defined in the Ora2Ora_config configuration and that uses the ora_tgt connection:

```
set replication type Ora2Ora_config ora_tgt audit
```

Managing Replication Tasks

Use the Server Manager Command Line Interface to work with replication tasks such as the Applier, Copy File, External, Extractor, Initial Sync, MS SQL Server Backup, and Send File tasks.

Commands are available to perform the following actions:

- Abort a replication task.
- Delete a replication task.
- Show detailed information about a replication task.
- Start a replication task that you previously stopped or canceled.
- Show information about replication schedules and unscheduled tasks that are running and that have previously run.
- Show a list of the tasks that are defined for the Server Manager.
- Show replication tasks that are in a particular schedule.
- Start a replication task.
- Stop a replication task.

abort task

Forces a running replication task to end. Use this command in urgent situations where you cannot wait for the task to complete normally.

Syntax

Enter the command with either the task run ID or the task name and configuration name:

```
abort task run_id
abort task task_name configuration_name
```

The following short forms are acceptable:

```
at run_id
at task_name configuration_name
```

Parameters

The following table describes the command parameters:

Parameter	Description
<i>run_id</i>	The unique execution identifier of the task.
<i>task_name</i>	The name of the task.
<i>configuration_name</i>	The name of the configuration that the running task is using.

Example

The following command forces a task that has the run ID of 61 to end:

```
abort task 61
```

delete task

Deletes a replication task from the SQLite database that the Server Manager uses.

Syntax

```
delete task task_name
```

The following short form is acceptable:

```
dt task_name
```

Parameters

The following table describes the command parameter:

Parameter	Description
<i>task_name</i>	The name of the task to delete.

Example

The following command deletes the `extract_custom` task:

```
delete task extract_custom
```

get task detail

Displays detailed information about a replication task.

This information includes the task type, task status, the Server Manager server that is designated to run the task, the command that runs the task, the database connection that the task uses, and other information.

Syntax

Enter the command with the task run ID:

```
get task detail run_id
```

The following short form is acceptable:

```
gtd run_id
```

Parameters

The following table describes the command parameter:

Parameter	Description
<i>run_id</i>	The unique execution identifier of the task.

Example

The following command gets information for the task that has the run ID of 77:

```
get task detail 77
```

Sample Output

```
task:          'ora2ora_cont_apply'  
run_id:       '77'  
task_type:    'applier'  
status:       'running'  
server:       'Main'  
start_date:   '2013-01-09 11:01:51'  
end_date:     '<none>'  
command:      'dbsync_apply_oracle.exe config=./configs\ora2ora.db SM_RUN_ID=77  
CONFIG_LOADER=./configs\ora2ora_2\ora2ora_loader.db INT_FILES DIR=./output\  
REMOVE_PROCESSED_FILES=n CONN_ID=2 REALTIME_MODE=y JOURNAL_MODE=WAL UDP_SERVER_PORT=8087  
SQLITE_PAGE_SIZE=16384 SQLITE_CACHE_SIZE=16384' connection: 'ora_src'  
run_dir:      '<none>'  
return_code:  '0'
```

restore task

Starts a replication task that you previously stopped or canceled in a running schedule.

Important: You cannot use the restore task command to start a task in a schedule that is not running.

Syntax

Enter the command with the task run ID:

```
restore task run_id
```

The following short form is acceptable:

```
rt run_id
```

Parameters

The following table describes the command parameter:

Parameter	Description
<i>run_id</i>	The unique execution identifier of the task.

Example

The following example command starts the task that has the run ID of 37:

```
restore task 37
```

show task history

Displays information about replication schedules and unscheduled tasks that are running and that have previously run. This information includes the schedule status, configuration associated with the schedule, task status and return code, and date and time when each schedule and task started.

Syntax

```
show task history [number_of_chain_ids offset] [-r|-s|-f]
```

The following short form is acceptable:

```
sth [number_of_chain_ids offset] [-r|-s|-f]
```

Parameters

The following table describes the command parameters:

Parameter	Description
<i>number_of_chain_ids</i>	The maximum number of chain IDs to display. Default value: 15
<i>offset</i>	The sequential number of the chain ID that the command displays as the first of the schedule instances. Default value: 0
[-r -s -f]	The status of the schedule instances and tasks for which to display information. Valid values are: <ul style="list-style-type: none">• -r for running.• -s for success.• -f for failed. If you omit this parameter, the command displays information for all schedule instances or tasks.

Example

The following command displays information about running schedules and tasks:

```
show task history
```

Sample Output

```
0)chain_id:'#40' schedule_name:'ora2ora_ondemand' status:'running' start_on:'01/10/2013' start_at:'09:32:18' config:'ora2ora2'  
0)run_id: '#89' | task_name: 'ora2ora_ondemand_extract' | status: 'running' | return_code: '0' | block_type: 'no block'  
1)run_id: '#88' | task_name: 'ora2ora_ondemand_sendFile' | status: 'sleeping' | return_code: '0' | block_type: 'blocked by
```



```

dependency'
2)run_id: '#87' | task_name: 'ora2ora_ondemand_apply' | status: 'sleeping' | return_code: '0' | block_type: 'blocked by
dependency'
1)chain_id: '#38' schedule_name: 'ora2ora_admin' status: 'running' start_on: '01/09/2013' start_at: '16:19:19' config: 'ora2ora'
0)run_id: '#83' | task_name: 'ora2ora_admin_extract' | status: 'sleeping' | return_code: '0' | block_type: 'blocked by task'
1)run_id: '#82' | task_name: 'ora2ora_admin_sendFile' | status: 'sleeping' | return_code: '0' | block_type: 'blocked by
dependency'
2)run_id: '#81' | task_name: 'ora2ora_admin_apply' | status: 'sleeping' | return_code: '0' | block_type: 'blocked by
dependency'

```

show task list

Lists all of the replication tasks that are defined for the Server Manager.

Syntax

```
show task list
```

The following short form is acceptable:

```
stl
```

show tasks

Displays replication tasks that are included in a specific schedule.

Syntax

```
show tasks schedule_name
```

The following short form is acceptable:

```
t1 schedule_name
```

Parameters

The following table describes the command parameter:

Parameter	Description
<i>schedule_name</i>	The name of the schedule for which to show tasks.

Examples

Both of the following example commands list the tasks that are in the Ora2Ora_schedule schedule:

```
show tasks Ora2Ora_schedule
t1 Ora2Ora_schedule
```

Sample Output

```

Schedule: 'Ora2Ora_schedule' status: running
run_id: #6 'Ora2Ora_schedule_extract' status: running
run_id: #7 'Ora2Ora_schedule_sendFile' status: running
run_id: #8 'Ora2Ora_schedule_apply' status: running

```

start task

Starts a replication task.

Syntax

```
start task task_name [-d] [-l] [args="task_parameters"] [configuration_name]
[connection_list]
```

The following short form is acceptable:

```
st task_name [-d] [-l] [args="task_parameters"] [configuration_name] [connection_list]
```

Parameters

The following table describes the command parameters:

Parameter	Description
<i>task_name</i>	The name of the replication task.
-d	Runs the task with its dependencies.
-l	Runs the task as a foreground process.
args="task_parameters"	One or more optional parameters for the task. If you enter multiple task parameters, use a space to separate them from one another. Enclose all of the task parameters in double quotation (") marks. See Appendix D in the <i>Informatica Data Replication User Guide</i> for a list of optional InitialSync, Extractor, and Applier command line parameters.
<i>configuration_name</i>	The name of the configuration with which to run the task. Include this parameter if the task requires a configuration.
<i>connection_list</i>	For configurations that have multiple targets, a space-separated list of database connections that correspond to the targets for which to run the task.

Examples

The following example command starts the extract task that uses the Ora2Ora_config configuration as a foreground process, skips integrity checking for the redo logs, and captures change data for transactions that have a default name:

```
start task extract -l args="SKIP_CONTINUITY_CHECK=y SKIP_DEFAULT_TX=n" Ora2Ora_config
```

The following example command starts the apply task that uses the Ora2Ora_config configuration in the background for the target that uses the Ora_tgt connection:

```
st apply Ora2Ora_config Ora_tgt
```

stop task

Stops a running replication task. Before stopping, the task completes normal processing.

Syntax

Enter the command with either the task run ID or the task name and configuration name:

```
stop task run_id  
stop task task_name configuration_name
```

The following short forms are acceptable:

```
stopt run_id  
stopt task_name configuration_name
```

Parameters

The following table describes the command parameters:

Parameter	Description
<i>run_id</i>	The unique execution identifier of the task.
<i>task_name</i>	The name of the task.
<i>configuration_name</i>	The name of the configuration that the running task is using.

Examples

The following example command stops the task that has the run ID of 37:

```
stop task 37
```

The following example command stops the extract task that uses the Ora2Ora_config configuration:

```
stopt extract Ora2Ora_config
```

Managing Replication Schedules

Use the Server Manager Command Line Interface to work with replication schedules.

Commands are available to perform the following tasks:

- Abort a schedule.
- Create a schedule.
- Delete a schedule.
- Edit a schedule.
- Show a list of the schedules that are running.
- Show detailed information about a schedule.
- Show a list of replication schedules.
- Start a schedule.
- Stop a schedule.

abort schedule

Forces a replication schedule to stop without first waiting for the scheduled tasks to complete processing normally. Use this command in urgent situations only.

Syntax

Enter the command with either the schedule name or schedule chain ID:

```
abort schedule schedule_name [-l]  
abort schedule chain_id=chain_id [-l]
```

The following short forms are acceptable:

```
as schedule_name [-l]
as chain_id=chain_id [-l]
```

Parameters

The following table describes the command parameters:

Parameter	Description
<i>schedule_name</i>	The name of the schedule to abort.
chain_id	The identification number of the schedule to abort. Tip: To see a list of existing schedules with their names and chain IDs, run the show schedules command.
-l	Wait for the schedule to stop before terminating the command. If you do not add this parameter, the command terminates immediately after sending the request to abort the schedule to the Server Manager.

Example

The following command forces the schedule named Ora20ra_continuous to stop:

```
abort schedule Ora20ra_continuous
```

create schedule

Creates a replication schedule.

Syntax

You can enter the command with or without parameters. If you run the command without any parameters, the Server Manager CLI prompts you to enter them in interactive mode. If you specify one or more optional parameters in the command, the Server Manager CLI uses these parameter values to update the schedule in non-interactive mode and preserves the values that were last set for the other optional parameters.

To enter the command with its parameters, use the following syntax:

```
create schedule schedule_name wizard_type configuration_name [owner=schedule_owner
sd=start_date st=start_time ed=end_date et=end_time days=week_days lst=log_storage_time
int=interval -s]
```

The following short form, which contains the required parameters only, is acceptable:

```
cs schedule_name wizard_type configuration_name
```

Parameters

The following table describes the command parameters:

Parameter	Description	Default
<i>schedule_name</i>	A unique name for the schedule. Note: Schedule names can contain only the digits 0-9, Latin letters A-Z and a-z, and the underscore (_) character. The Server Manager CLI truncates schedule names that are longer than 100 characters.	No default value.
<i>wizard_type</i>	The type of schedule execution scheme. Valid values are: - sequential or seq - parallel or pri - continuous or con	No default value.
<i>configuration_name</i>	The name of the replication configuration to use for the schedule.	No default value.
<i>owner</i>	The name of the user who is the owner of the schedule.	The user who is connected to the Server Manager Main server and invokes the create schedule command.
<i>sd</i>	The date on which to start the schedule, in the format that corresponds to the local system settings.	The current date.
<i>st</i>	The time at which to start the schedule, in the following format: <code>hh:mm:ss</code>	The current time.
<i>ed</i>	The date on which to stop the schedule, in the format that corresponds to the local system settings.	One year after the current date.
<i>et</i>	The time at which to stop the schedule, in the following format: <code>hh:mm:ss</code>	23:59:00
<i>days</i>	A comma-, semicolon-, or space-separated list of the days of the week on which to run the schedule. Valid values are: - mon - Represents Monday. - tue - Represents Tuesday. - wed - Represents Wednesday. - thu - Represents Thursday. - fri - Represents Friday. - sat - Represents Saturday. - sun - Represents Sunday. - all - Represents all days of the week. Important: Enclose a space-separated list with single quotation marks.	all
<i>lst</i>	The number of days to retain the schedule logs.	30 days
<i>int</i>	A repeat interval, in seconds, for running the schedule. Important: Omit this parameter for continuous replication schedules.	3600 seconds (60 minutes)
<i>-s</i>	Runs the schedule immediately after it is created.	No default value.

Examples

The following example command displays a series of prompts for entering the parameters in interactive mode and then creating the schedule:

```
cs
```

The following example command specifies all of the parameters inline and creates the continuous replication schedule that has the name `Ora2Ora_schedule` and uses the configuration `Ora2Ora_config`:

```
create schedule Ora2Ora_schedule continuous Ora2Ora_config owner=idradadmin sd=01.04.2013
st=10:00:0
0 ed 01.04.2014 et=23:59:00 days=all lst=7
```

delete schedule

Deletes a replication schedule.

Syntax

```
delete schedule schedule_name [-d] [-f]
```

The following short form is acceptable:

```
ds schedule_name [-d] [-f]
```

Parameters

The following table describes the command parameters:

Parameter	Description
<i>schedule_name</i>	The name of the schedule to delete.
-d	Delete the tasks that are included in the schedule.
-f	Stop the schedule before deleting it.

Examples

The following example command deletes the `Ora2Ora_daily` schedule and all of its tasks:

```
delete schedule Ora2Ora_daily -d
```

The following example stops the `Ora2Ora_daily` schedule and then deletes it:

```
ds Ora2Ora_daily -f
```

edit schedule

Edits parameters for a specific replication schedule.

Syntax

Enter the command with the required `schedule_name` parameter and any of the optional parameters, or enter the command with the required parameter only. The following command syntax includes all of the parameters:

```
edit schedule schedule_name [name=new_schedule_name] [type=schedule_type]
[owner=schedule_owner] [cfg=configuration_name] [enable=run_now] [sd=start_date]
[st=start_time] [ed=end_date] [et=end_time] [days=week_days] [lst=log_storage_time]
[int=interval]
```

The following short form, which contains the required parameter only, is acceptable:

```
es schedule_name
```

If you run the command without any parameters, the Server Manager CLI prompts you to enter them in interactive mode. If you specify one or more optional parameters in the command, the Server Manager CLI uses these parameter values to update the schedule in non-interactive mode and preserves the values that were last set for the other optional parameters.

Parameters

The following table describes the command parameters:

Parameter	Description
<code>schedule_name</code>	The name of the schedule that you want to edit.
<code>name</code>	A new name for the schedule. Note: Schedule names can contain only the digits 0-9, Latin letters A-Z and a-z, and the underscore (_) character. The Server Manager CLI truncates schedule names that are longer than 100 characters.
<code>type</code>	The type of schedule. Valid values are: <ul style="list-style-type: none"> • on demand or ond • periodically or prd • continuous or con
<code>owner</code>	The name of the user who is the owner of the schedule. The default value is the user who is connected to the Server Manager Main server and invokes the edit schedule command.
<code>cfg</code>	The name of the replication configuration to use for the schedule.
<code>enable</code>	Indicates whether to run the schedule immediately after it is edited. Valid values are: <ul style="list-style-type: none"> • y. Run the schedule after it is edited. • n. Run the schedule at the specified start date and time.
<code>sd</code>	The date on which to start the schedule, in the format that corresponds to the local system settings.
<code>st</code>	The time at which to start the schedule, in the following format: hh:mm:ss
<code>ed</code>	The date on which to stop the schedule, in the format that corresponds to the local system settings.
<code>et</code>	The time at which to stop the schedule, in the following format: hh:mm:ss

Parameter	Description
days	<p>A comma-, semicolon-, or space-separated list of the days of the week on which to run the schedule. Valid values are:</p> <ul style="list-style-type: none"> • mon - Represents Monday. • tue - Represents Tuesday. • wed - Represents Wednesday. • thu - Represents Thursday. • fri - Represents Friday. • sat - Represents Saturday. • sun - Represents Sunday. • all - Represents all days of the week. <p>Important: Enclose a space-separated list in single quotation marks.</p>
lst	The number of days to retain schedule logs.
int	<p>The repeat interval, in seconds, for running the schedule.</p> <p>Important: Omit this parameter for continuous replication schedules.</p>

Examples

The following example command displays a series of prompts for editing the parameters in interactive mode for the ora2ora schedule:

```
es ora2ora
```

The following example command specifies the changed parameters for the ora2ora_sequential schedule:

```
edit schedule ora2ora_sequential name=ora2ora_continuous type=con owner=idradmin
cfg=ora2ora enable=n sd=04.01.2013 st=10:00:00 ed=04.01.2014 et=23:59:00 days=all lst=7
```

show running schedules

Displays a list of replication schedules that are running.

Syntax

```
show running schedules
```

The following short form is acceptable:

```
srs
```

show schedule info

Displays detailed information about a specific replication schedule.

This information includes the schedule name, schedule type, schedule status, the replication configuration that the schedule uses, the owner of the configuration, and other information.

Syntax

```
show schedule info schedule_name
```

The following short form is acceptable:

```
ssi schedule_name
```


Parameters

The following table describes the command parameter:

Parameter	Description
<i>schedule_name</i>	The name of the schedule for which to display detailed information.

Example

The following example command displays detailed information for the Ora2Ora_continuous schedule:

```
ssi Ora2Ora_continuous
```

Sample Output

```
name           'Ora2Ora_continuous'  
type           'continuous'  
enabled        'true'  
status         'running'  
config         'Test'  
owner          'idradmin'  
starts on     'Feb 27, 2013 10:00:00 AM'  
stops on      'Mar 30, 2013 10:00:00 AM'  
run every     'continuous'  
days         'fri; sat; tue; mon; wed; sun; thu;'  
store logs for '1 days'  
tasks:  
  Ora2Ora_apply: sleeping run_id:#59 [with dependency]  
last chain_id:#21
```

show schedules

Displays a list of replication schedules that includes the schedule statuses.

Syntax

```
show schedules
```

The following short form is acceptable:

```
s1
```

start schedule

Starts a specific replication schedule.

Syntax

```
start schedule schedule_name
```

The following short form is acceptable:

```
ss schedule_name
```

Parameters

The following table describes the command parameter:

Parameter	Description
<i>schedule_name</i>	The name of the schedule to start.

stop schedule

Stops a specific replication schedule after waiting for the scheduled tasks to complete processing normally.

Syntax

Enter the command with either the schedule name or the replication chain ID:

```
stop schedule schedule_name [-l]
stop schedule chain_id=chain_id [-l]
```

The following short forms are acceptable:

```
stopsch schedule_name [-l]
stopsch chain_id=chain_id [-l]
```

Parameters

The following table describes the command parameters:

Parameter	Description
<i>schedule_name</i>	The name of the schedule to stop.
chain_id	The chain ID of the schedule to stop.
-l	Wait for the schedule to stop before terminating the command. If you do not add this parameter, the command terminates immediately after sending the request to stop the schedule to the Server Manager.

Viewing Execution Logs for Replication Tasks

Use the Server Manager Command Line Interface to view execution logs for replication tasks.

You can view a standard execution log, an additional execution log, or an execution log that is automatically refreshed.

show logs

Displays an execution log for a specific replication task.

Each replication task generates logs of the following types:

- **Standard log.** Contains the task output. A replication task might generate several standard log files with the following file name format:

```
output<sequence_number>.log[.gz|.zip]
```

- **Additional log.** Contains the contents of the Server Manager logs for the task execution, including command-line, error, and debug information. A replication task generates a single additional log with the following file name format:

```
additional_output.log[.gz|.zip]
```

Note: Depending on the log compression setting of a Server Manager instance, when a task finishes processing a log file, it compresses the log file and adds the extension .gz or .zip.

Syntax

Enter the command with the task run ID parameter and optionally with either the log file name or sequence number:

```
show logs run_id [file_name]  
show logs run_id [sequence_number]
```

If you do not specify the optional *file_name* or *sequence_number* parameter, the Server Manager CLI lists the execution logs for the specified task in interactive mode. You can then enter the sequence number of the log file to view.

Parameters

The following table describes the command parameters:

Parameter	Description
<i>run_id</i>	The unique execution identifier of the replication task. Tip: To list tasks with their unique run IDs, run the show task history command.
<i>file_name</i>	The file name of the execution log that you want to view. Note: You can omit the file extension.
<i>sequence_number</i>	The sequence number of the log file for the specified task. Note: An additional log file always has the <i>sequence_number</i> value of 0. The first file of a standard log always has the <i>sequence_number</i> value of 1.

Examples

The following example command displays the first standard log file named output1 for the task that has the run ID of 25.

```
show logs 25 output1
```

The following example command displays the log file that has the sequence number of 1 for the task that has the run ID of 25.

```
show logs 25 1
```

start log watcher

Displays an execution log for a specific replication task, which is automatically refreshed each time the task logs a new message.

Syntax

Enter the command with the task run ID and optionally with the -a and -i parameters:

```
start log watcher run_id [-a] [-i]
```

The following short form is acceptable:

```
slw run_id [-a] [-i]
```

To quit from the log watcher, press ESC.

Parameters

The following table describes the command parameters:

Parameter	Description
<i>run_id</i>	The unique execution identifier of the replication task.
<i>-a</i>	Display the additional log instead of the standard log.
<i>-i</i>	Display the entire log. If you do not add this parameter, the start log watcher command displays only new log messages.

Managing Connections

Use the Server Manager Command Line Interface to work with connections to sources and targets.

Commands are available to perform the following tasks:

- Create a connection to a source or target.
- Delete a connection.
- Edit a connection.
- List connections that are defined in the Server Manager.
- View connection information.

add connection

Creates a connection to a source or target.

Syntax

To enter the command and its parameters, use the following syntax:

```
add connection name=connection_name type=database_type host=host port=port login=login
passwd=password [owner=owner] [instance=instance/database]
[ccs=custom_connection_string] [jdbc=jdbc_custom_url] [path=flat_file_path]
[https=true|false] [asm_host=asm_host] [asm_port=asm_port] [asm_user=asm_user]
[asm_passwd=asm_password] [asm_instance=asm_instance]
[asm_ccs=asm_custom_connection_string] [asm_jdbc=asm_jdbc_custom_url]
[kafka_path=kafka_path] [kafka_topic=kafka_topic] [kafka_format=avro]
[kafka_structure=avroFlatSchemaFormatV1|avroNestedSchemaFormatV2] [kafka_encoding=binary|
json] [version=version] [-c] [-d] [-r] [-s] [-a]
```

The following short form is acceptable:

```
ac name=connection_name type=database_type host=host port=port login=login
passwd=password [owner=owner] [instance=instance/database]
[ccs=custom_connection_string] [jdbc=jdbc_connection_string] [path=flat_file_path]
[https=true|false] [asm_host=asm_host] [asm_port=asm_port] [asm_user=asm_user]
[asm_passwd=asm_password] [asm_instance=asm_instance]
[asm_ccs=asm_custom_connection_string] [asm_jdbc=asm_jdbc_connection_string]
[kafka_path=kafka_path] [kafka_topic=kafka_topic] [kafka_format=avro]
[kafka_structure=avroFlatSchemaFormatV1|avroNestedSchemaFormatV2] [kafka_encoding=binary|
json] [version=version] [-c] [-d] [-r] [-s] [-a]
```

Parameters

The following table describes the command parameters:

Parameter	Description
name	A name for the source or target connection that you are adding. Connection names must be unique. The names can contain the digits 0-9, Latin letters A-Z and a-z, and the underscore (_) character. The Data Replication Console truncates connection names that are longer than 120 characters.
type	A source or target type. Valid values are: <ul style="list-style-type: none">• amazon redshift for Amazon Redshift targets.• db2 for DB2 for Linux, UNIX, and Windows databases.• flat file for Flat File targets.• greenplum for Greenplum appliances.• kafka for Apache Kafka targets.• ms sql server for Microsoft SQL Server databases.• mysql for MySQL databases.• netezza for Netezza appliances.• oracle for Oracle databases.• postgresql for PostgreSQL databases.• teradata for Teradata appliances.• vertica for Vertica appliances.
host	Host name or IP address of the machine where the source or target runs.
port	Port number that Data Replication uses to connect to the source or target. Valid values are integers from 1 through 65535.
login	A user name that has the authority to connect to the specified source or target.
passwd	A valid password for the specified user.
owner	The Server Manager user to be used as the connection owner.
instance	Depending on the source or target type, enter one of the following types of values: <ul style="list-style-type: none">• An Amazon Redshift, DB2, Greenplum, Netezza, PostgreSQL, or Vertica database name.• An Oracle instance name that is defined in an ORACLE_SID environment variable.
css	A custom connection string for the ODBC driver that the Extractor uses to access the source or target. If you use an ODBC Data Source Name (DSN) to connect to the source database, provide the DSN connection string. On Windows, if you installed multiple ODBC driver versions on one system, use this setting to specify a name for the driver that you want to use.
jdbc	A custom URL for the JDBC driver that the Server Manager CLI uses to access the source or target.
path	For flat file targets, a directory in the distributed file system to which the Applier loads flat files.

Parameter	Description
https	For Amazon Redshift and Greenplum targets, specifies whether to use the HTTPS protocol to connect to the target. Valid values are: <ul style="list-style-type: none"> • true • false Default value: false
asm_host	The host name or IP address of the system with the Oracle ASM instance.
asm_port	The port number that is used to connect to the Oracle ASM instance. Valid values are integers from 1 through 65535. Default value: 1521.
asm_user	The user name of the default Oracle user that was created at Oracle installation, or another user that you define. This user must have SYSDBA privileges.
asm_passwd	A clear text password for the Oracle ASM user that is specified in the asm-user parameter.
asm_instance	An Oracle ASM instance name that is defined in the ORACLE_SID environment variable.
asm_ccs	A custom connection string for the ODBC driver that the Extractor uses to access the Oracle ASM database.
asm_jdbc	A custom URL for the JDBC driver that the Server Manager uses to access the specified Oracle ASM database.
kafka_path	For Apache Kafka targets, the path to the directory that contains the Kafka producer properties file.
kafka_topic	For Apache Kafka targets, the Kafka topic or topics to which Data Replication publishes messages that contain change data by using this connection. Enter one of the following values: <ul style="list-style-type: none"> • The name of the topic for all of the mapped tables in all of the configurations that use this connection. • The value USE_TABLE_NAME, which causes Data Replication to publish messages to a separate topic for each mapped table in each configuration that uses this connection. By default, Kafka automatically creates a topic when Data Replication publishes a message to the topic but the topic does not yet exist. If you do not want Kafka to automatically create topics, set the auto.create.topics.enable property to false in the Kafka server.properties file. With this setting, you must first create the topic or topics to which Data Replication publishes messages, before issuing this command. Otherwise, Data Replication issues an error.
kafka_format	For Apache Kafka targets, the message format. The only valid value is avro.
kafka_structure	For Apache Kafka targets, a message structure for the avro format. Valid values are avroFlatSchemaFormatV1 and avroNestedSchemaFormatV2.
kafka_encoding	For Apache Kafka targets, the message encoding type for the avro format. Valid values are binary and json.
version	For Cloudera and Hortonworks targets, the version number.
-c	Indicates whether to use an ODBC custom connection string or JDBC custom URL to connect to the source or target. If you specify this parameter, you must also specify the css parameter, jdbc parameter, or both.
-d	For Oracle sources and targets, indicates that you connect to the database without using the Oracle instance. The connection string takes the form of /.

Parameter	Description
-r	For Oracle sources and targets, indicates that you capture data only from Oracle archived redo logs in a RAC environment.
-s	For Oracle sources and targets, indicates whether to use the Oracle SID instead of the SERVICE_NAME to build the connection string.
-a	For Oracle ASM sources and targets, indicates whether to use the Oracle SID instead of the SERVICE_NAME to build the connection string.

Examples

The following command creates the ora_src connection:

```
ac name=ora_src type=oracle host=10.51.40.78 port=1521 login=DATAREP_USER
passwd=DATAREP_USER_PASS owner=idradadmin instance=orcl
```

The following command creates the asm_connection connection:

```
add connection name=asm_connection type=oracle host=10.51.40.26 port=1521 login='sys as
sysdba' passwd=84218421 instance=orcl asmHost=10.51.40.26 asmPort=1521 asmUser=sys
asmPasswd=84218421 asmInstance=+asm
```

delete connection

Deletes a connection to a source or target. The connection must not be assigned to any configurations.

Syntax

```
delete connection connection_name
```

The following short form is acceptable:

```
dcn connection_name
```

Parameters

The following table describes the command parameter:

Parameter	Description
<i>connection_name</i>	The name of the source or target connection to delete.

Example

The following example command deletes the oracle_source connection:

```
dcn oracle_source
```

edit connection

Edits a connection to a source or target. The updated connection information will be used for all configurations to which the connection is assigned.

Syntax

To enter the command and its parameters, use the following syntax:

```
edit connection connection_name [name=new_name] [type=database_type] [host=host]  
[port=port] [login=login] [passwd=password] [owner=owner] [instance=instance/database]  
[ccs=custom_connection_string] [jdbc=jdbc_connection_string] [path=flat_file_path]  
[https=true|false] [asm_host=asm_host] [asm_port=asm_port] [asm_user=asm_user]  
[asm_passwd=asm_password] [asm_instance=asm_instance]  
[asm_ccs=asm_custom_connection_string] [asm_jdbc=asm_jdbc_connection_string]  
[kafka_path=kafka_path] [kafka_topic=kafka_topic] [kafka_format=avro]  
[kafka_structure=avroFlatSchemaFormatV1|avroNestedSchemaFormatV2] [kafka_encoding=binary|  
json] [version=version] [-c] [-d] [-r] [-s] [-a] [-f] [-u]
```

The following short form is acceptable:

```
ecn connection_name [name=new_name] [host=host] [port=port] [login=login]  
[passwd=password] [owner=owner] [instance=instance/database]  
[ccs=custom_connection_string] [jdbc=jdbc_connection_string] [path=flat_file_path]  
[https=true|false] [asm_host=asm_host] [asm_port=asm_port] [asm_user=asm_user]  
[asm_passwd=asm_password] [asm_instance=asm_instance]  
[asm_ccs=asm_custom_connection_string] [asm_jdbc=asm_jdbc_connection_string]  
[kafka_path=kafka_path] [kafka_topic=kafka_topic] [kafka_format=avro]  
[kafka_structure=avroFlatSchemaFormatV1|avroNestedSchemaFormatV2] [kafka_encoding=binary|  
json] [version=version] [-c] [-d] [-r] [-s] [-a] [-f] [-u]
```

Parameters

The following table describes the command parameters:

Parameter	Description
<i>connection_name</i>	The name of the connection that you want to edit.
name	<p>A new name for the source or target connection.</p> <p>Connection names must be unique. The names can contain the digits 0-9, Latin letters A-Z and a-z, and the underscore (_) character. The Data Replication Console truncates connection names that are longer than 120 characters.</p> <p>If you are renaming a connection that is used in existing configurations, the Server Manager CLI prompts you for confirmation. If you confirm the new connection name, the name is changed in all of the configurations that used the previous name for the connection.</p>
type	<p>A source or target type. Valid values are:</p> <ul style="list-style-type: none">• amazon redshift for Amazon Redshift targets.• db2 for DB2 for Linux, UNIX, and Windows databases.• flat file for Flat File targets.• greenplum for Greenplum appliances.• kafka for Apache Kafka targets.• ms sql server for Microsoft SQL Server databases.• mysql for MySQL databases.• netezza for Netezza appliances.• oracle for Oracle databases.• postgresql for PostgreSQL databases.• teradata for Teradata appliances.• vertica for Vertica appliances.

Parameter	Description
host	Host name or IP address of the machine where the source or target runs.
port	Port number that Data Replication uses to connect to the source or target. Valid values are integers from 1 through 65535.
login	A user name that has the authority to connect to the specified source or target.
passwd	A valid password for the specified user.
owner	A Server Manager user to be used as the connection owner.
instance	Depending on the source or target type, enter one of the following types of values: <ul style="list-style-type: none"> • An Amazon Redshift, DB2, Greenplum, Netezza, PostgreSQL, or Vertica database name. • An Oracle instance name that is defined in an ORACLE_SID environment variable.
css	A connection string for the ODBC driver that the Extractor uses to access the source or target. If you use an ODBC Data Source Name (DSN) to connect to the source database, provide the DSN connection string. On Windows, if you installed multiple ODBC driver versions on one system, use this setting to specify a name for the driver that you want to use.
jdbc	A custom URL for the JDBC driver that the Server Manager CLI uses to access the source or target.
path	For flat file targets, a directory in the distributed file system to which the Applier loads flat files.
https	For Amazon Redshift and Greenplum targets, specifies whether to use the HTTPS protocol to connect to the target. Valid values are: <ul style="list-style-type: none"> • true • false Default value: false
asm_host	The host name or IP address of the system with the Oracle ASM instance.
asm_port	The port number that is used to connect to the Oracle ASM instance. Valid values are integers from 1 through 65535. Default value: 1521.
asm_user	The user name of the default Oracle user that was created at Oracle installation, or another user that you define. This user must have SYSDBA privileges.
asm_passwd	A clear text password for the Oracle ASM user that is specified in the asm_user parameter.
asm_instance	An Oracle ASM instance name that is defined in the ORACLE_SID environment variable.
asm_ccs	A custom connection string for the ODBC driver that the Extractor uses to access the Oracle ASM database.
asm_jdbc	A custom URL for the JDBC driver that the Server Manager uses to access the specified Oracle ASM database.
kafka_path	For Apache Kafka targets, the path to the directory that contains the Kafka producer properties file.

Parameter	Description
kafka_topic	<p>For Apache Kafka targets, the Kafka topic or topics to which Data Replication publishes messages that contain change data by using this connection. Enter one of the following values:</p> <ul style="list-style-type: none"> • The name of the topic for all of the mapped tables in all of the configurations that use this connection. • The value USE_TABLE_NAME, which causes Data Replication to publish messages to a separate topic for each mapped table in each configuration that uses this connection. <p>By default, Kafka automatically creates a topic when Data Replication publishes a message to the topic but the topic does not yet exist. If you do not want Kafka to automatically create topics, set the auto.create.topics.enable property to false in the Kafka server.properties file. With this setting, you must first create the topic or topics to which Data Replication publishes messages, before issuing this command. Otherwise, Data Replication issues an error.</p>
kafka_format	For Apache Kafka targets, the message format. The only valid value is avro.
kafka_structure	For Apache Kafka targets, a message structure for the avro format. Valid values are avroFlatSchemaFormatV1 and avroNestedSchemaFormatV2.
kafka_encoding	For Apache Kafka targets, the message encoding type for the avro format. Valid values are binary and json.
version	For Cloudera and Hortonworks targets, the version number.
-c	Indicates whether to use an ODBC custom connection string or JDBC custom URL to connect to the source or target. If you specify this parameter, you must also specify the css parameter, jdbc parameter, or both.
-d	For Oracle sources and targets, indicates that you connect to the database without using the Oracle instance. The connection string takes the form of /.
-r	For Oracle sources and targets, indicates that you capture data only from Oracle archived redo logs in a RAC environment.
-s	For Oracle sources and targets, indicates whether to use the Oracle SID instead of the SERVICE_NAME to build the connection string.
-a	For Oracle ASM sources and targets, indicates whether to use the Oracle SID instead of the SERVICE_NAME to build the connection string.
-f	When the Server Manager CLI runs in non-interactive mode, forces the connection edits to be committed to the configurations that use the connection unless you changed the source or target type specified in the type parameter.
-u	If you changed the instance parameter value, updates the database object IDs for all configurations that use the connection.

Examples

The following command changes the owner of the ora_src connection to user321:

```
ecm ora_src owner=user321
```

show connection info

Displays detailed information about a connection that is defined in the Server Manager.

Syntax

```
show connection info connection_name
```

The following short form is acceptable:

```
sci connection_name
```

Parameters

The following table describes the command parameters:

Parameter	Description
<i>connection_name</i>	The name of the connection for which to display detailed information.

Example

The following example command displays detailed information for the oracle_source connection:

```
sci oracle_source
```

Sample Output

```
General Info
Name                oracle_source
Vendor              Oracle_
Host                10.51.40.26
Port                1521
Login               src
Instance            orcl
Uses custom URL     true
JDBC custom URL
Custom Connection String
Uses direct connection      false
Supports RAC                false
Uses SID instead of SERVICE_NAME  false
Enable ASM                   true
ASM instance host name       10.51.40.26
ASM instance port            1521
ASM sys username              sys
ASM instance                  +asm
Uses ASM SID instead of SERVICE_NAME  false
ASM JDBC custom URL           false
ASM Custom Connection String    false
Owner                          idradmin

ASM properties
ShowConnectionInfo.host        10.51.40.26
ShowConnectionInfo.port        1521
ShowConnectionInfo.login        sys
ShowConnectionInfo.instance     +asm
ShowConnectionInfo.custom_jdbc_url
ShowConnectionInfo.custom_conn_string
ShowConnectionInfo.uses_sid     false

Number of configurations, using this connection
as a source                      1
as a primary target               0
as a secondary target             0
```

show connections

Displays a list of source and target connections that are defined in the Server Manager.

Syntax

```
show connections
```

The following short form is acceptable:

```
sc
```

Managing Open Transactions

Use the Server Manager Command Line Interface to view open transactions on the source and to commit or roll back open transactions on the target.

Commands are available to perform the following tasks:

- Show the open transactions on the source database.
- Mark an open transaction as a transaction to be committed or rolled back on the target database.

force transaction

Marks an open transaction for commit or rollback processing on the target.

Important: The Extractor does not process any change records that are written to a source transaction after the commit or rollback record.

Syntax

```
force transaction configuration_name transaction_XID "internal_SCN" {commit|rollback}
```

Parameters

The following table describes the command parameters:

Parameter	Description
<i>configuration_name</i>	The name of the configuration for which to commit or roll back an open transaction.
<i>transaction_XID</i>	The unique identifier of the open transaction.
<i>internal_SCN</i>	The SCN or LSN value for the BEGIN TRANSACTION operation in the database format.
{commit rollback}	Whether to commit or roll back the open transaction on the target. Valid values are: <ul style="list-style-type: none">• commit or 0. Commit the open transaction.• rollback or 1. Roll back the open transaction.

Example

The following example command marks an open transaction for a commit on the target for the Oracle2Oracle configuration:

```
force transaction oracle2oracle 0x0009.007.0001b473 "13948547824 0" commit
```

show open transactions

Displays a list of open transactions.

Syntax

```
show open transactions configuration_name
```

Parameters

The following table describes the command parameter:

Parameter	Description
<i>configuration_name</i>	The name of the configuration for which to display open transactions.

Example

The following example command lists open transactions for the Oracle2Oracle configuration:

```
show open transactions Oracle2Oracle
```

Sample Output

The following sample output is for a configuration with an Oracle source:

```
Transaction #0
Transaction Name: 'qwerty'
XID: '0x0009.007.0001b473'
Begin Transaction SCN: '0x0003.3f65f2f0'
Loopback Transaction: 'false'
Records in Transaction: '9'
Redo Thread ID: '1'
System Table Transaction: '0'
Begin Transaction Timestamp: 'Apr 1, 2014 2:13:43 PM'
Undo Block Address (UBA): '0x00c00100.0001b473.0007'
Session ID: '14'
Session Serial Number: '11279'
Internal SCN: '13948547824 0'
Internal XID: '2533304855278707'
Transaction size (in bytes): '1584'
Database Transaction Status: 'ACTIVE'
Database User Name: 'SRC'
System User: 'abobkov'
Host: 'KZW178313'
```

The following table describes the properties of open transactions that can appear in the output, depending on the source type and whether the transaction is marked for commit or rollback processing:

Property	Description
Transaction	A transaction counter that the Server Manager CLI generates to list open transactions. The counter uses the format <i>#number</i> . The <i>number</i> value begins at 0 and is incremented by 1 for each subsequent open transaction for the specified configuration.
Transaction Name	The name of the open transaction.

Property	Description
XID	The transaction ID of the open transaction. Note: For Oracle sources, the Server Manager CLI might list multiple open transactions with the same XID value if Oracle reuses transaction IDs. When listing these transactions, the command first lists the transactions for unmapped tables that do not have a Begin Transaction Timestamp and then lists the other transactions in chronological order, beginning with the transaction that has the earliest Begin Transaction Timestamp .
Begin Transaction LSN or Begin Transaction SCN (for Oracle sources)	The SCN or LSN value for the BEGIN TRANSACTION operation.
Loopback Transaction	Indicates whether the Extractor skips the transaction based on the transaction name. The Extractor skips transactions if their names are specified in the SKIP_TX command line parameter. If the SKIP_DEFAULT_TX parameter is set to Y, the Extractor also skips transactions that have the default name of DbSyncTransaction. Valid values are: - true . The Extractor skips the transaction. - false . The Extractor does not skip the transaction.
Records in Transaction	The number of records in the transaction that are available in intermediate files.
Database ID or Redo Thread ID (for Oracle sources)	For Oracle sources, the unique identifier of a redo thread. For other sources, the unique identifier of a database.
DDL Transaction or System Table Transaction (for Oracle sources)	Indicates whether the transaction affects database system tables. A transaction that affects database system tables usually includes DDL operations. The database system tables that Data Replication monitors are specified in the following parameters in the <i>DataReplication_installation/uiconf/default.cfg</i> file: - db2_fixed_source_tables_names - mssql_fixed_source_tables_names - oracle_fixed_source_tables_names Valid values are: - 0 . The transaction does not affect system tables. - 1 . The transaction affects system tables.
Begin Transaction Timestamp	The date and time of the first redo record for the mapped table. This value displays only if the transaction includes changes to a mapped table in the configuration. If the transaction includes changes to an unmapped source table, the Begin Transaction Timestamp property does not contain a value.
Undo Block Address (UBA)	For Oracle sources, specifies the Undo Block Address.
Session ID	For Oracle sources, specifies the identifier of the session that is associated with the open transaction.
Session Serial Number	For Oracle sources, specifies the serial number of the session that is associated with the open transaction.
Internal LSN or (for Oracle) Internal SCN	The LSN or SCN value for the BEGIN TRANSACTION operation in the database format.
Internal XID	The transaction ID in the database format.
Transaction size	The amount of data, in bytes, in the transaction records in the intermediate files.

Property	Description
Database Transaction Status	The status of the transaction that is reported by the source database. If the source database does not contain information about the transaction, Data Replication sets the transaction status to NOT ACTIVE.
Database User Name	The name of the database user who opened the transaction.
System User	The name of the system user who opened the transaction.
Host	The IP address or host name of the system with the database client that opened the transaction.
Application ID	For DB2 sources, the ID of the database client that opened the transaction.

Deploying Configurations Locally or to Another Replication Environment

Use the Server Manager CLI to deploy configurations or configuration changes for which you are the owner. You can perform deployments to the local Server Manager Main server or remotely to another Main server. If you are the idradmin user, you can deploy any replication configuration.

Use the deploy config command to deploy an entire configuration to a local or remote Server Manager Main server. The deploy config command is equivalent to the **Full** deployment type in the Data Replication Console. You must perform a full deployment the first time you deploy a configuration.

After the initial deployment, you can use the redeploy config command to deploy configuration changes. The redeploy config command is equivalent to the **Partial** deployment type in the Data Replication Console.

In the following common scenarios, perform a full deployment:

- Deploy a configuration from a test environment to a remote production environment.
- Deploy a configuration to the local Server Manager Main server to quickly generate multiple configuration files based on the original configuration.
- Deploy a configuration to the local Server Manager Main server to configure a replication job that replicates data from multiple sources to a single target.

The Data Replication documentation uses the following terms to refer to the configurations and environments that are involved in the deployment process:

- **Destination environment.** The environment to which you deploy a configuration or configuration changes. The deployment destination can be a remote production environment or a local environment that uses the same Server Manager Main server.
- **New configuration.** The copy of the original configuration in the destination environment.
- **Original configuration.** The configuration that you deploy from the original environment.
- **Original environment.** The environment from which you deploy the original configuration.

The Server Manager CLI uses the Replication Configuration CLI to update the replication configuration in the destination environment immediately after the deployment. The Replication Configuration CLI updates source and target schemas and object IDs. By default, the Replication Configuration CLI also creates virtual indexes in the configuration for the source tables that do not have primary keys.

Deployment Considerations

Review the following guidelines that pertain to using the Server Manager CLI to deploy replication configurations:

- Before you deploy a configuration or configuration changes to a destination environment, manually apply any DDL changes to the destination databases that are necessary to make the databases match.
- Ensure that all components in both environments use the same version of Data Replication. If one environment uses an earlier Data Replication version, upgrade that Data Replication installation.
- In the *DataReplication_installation/uiconf/default.cfg* file on the computer where you use the Server Manager CLI, set the `use_default_values_during_saving` parameter to true:

```
use_default_values_during_saving=true
```

The Replication Configuration CLI requires this parameter setting to automatically use default values and stay in noninteractive mode when saving a configuration.

- If the source tables that you mapped in the configuration do not have primary keys, the Replication Configuration CLI creates a virtual index for these tables, by default, to properly process Updates and Deletes. The index includes all of the table columns. Informatica recommends the default setting of 3 to process the commands for deploying configurations. Alternatively, you can disable the replication of Update and Delete operations by specifying the following parameter in the *DataReplication_installation/uiconf/default.cfg* file:

```
pk_warning_option parameter=2
```

With this parameter set to 2, Data Replication replicates only Insert operations.

Note: If you set the parameter to 4, Data Replication ignores the warning that is issued when it encounters source tables in a configuration that do not have primary keys. Informatica does not recommend this setting because Update and Delete operations might be replicated incorrectly for the tables that do not have a primary key or unique index definition.

- You can replace the schema names from the original environment during the initial configuration deployment. However, do not define schema replacement rules in the deployment properties file that replace a single schema name with multiple schema names or that replace multiple schema names with a single schema name. When defining schema replacement rules, ensure that each schema in the original environment corresponds to a single schema in the destination environment.
- The deployment process uses the login credentials of the `prodSmUser` user that are specified in the `deploy.properties` file to connect to the Server Manager in the destination environment. For a remote deployment, these login credentials include the password that is specified in the `passwd` parameter of the `deploy config` or `redeploy config` command. These login credentials cannot be in use by any running Data Replication application in the destination environment during the deployment. Before you deploy a configuration or configuration changes to a destination environment, ensure that no Data Replication applications are running under these login credentials. Also, ensure that the login credentials remain inactive in the destination environment throughout the deployment process.
- If you make DDL changes to tables or columns in the destination database after an initial deployment of a configuration, the DDL changes will be lost when you redeploy the configuration unless you perform one of the following actions:
 - Make the same DDL changes to the database in the original environment before you redeploy the configuration.
 - In the `redeploy config` command, use the `-r` or `-e` parameter, or both parameters. If you use the `-m` or `-a` parameter in the `redeploy config` command, the DDL changes to the destination databases will be lost.

deploy config

Deploys a replication configuration to a local or remote Server Manager Main server. The deploy config command is equivalent to the **Full** deployment type in the Data Replication Console.

Syntax

Enter the command with the required *configuration_name* and *path_to_deploy_properties_file* parameters and any of the optional parameters. The following command syntax includes all of the parameters:

```
deploy config configuration_name path_to_deploy_properties_file  
[path_to_save_updated_deploy_properties_file] [passwd=password] [-l]
```

Parameters

The following table describes the command parameters:

Parameter	Description
<i>configuration_name</i>	The name of the replication configuration to deploy.
<i>path_to_deploy_properties_file</i>	An absolute path to the directory from which Data Replication uploads the deployment properties file that contains deployment information for the selected configuration, including the required <i>prodSrcConnection</i> and <i>prodTgtConnection</i> parameters. Note: To create the initial deployment properties file, use the default <i>deploy.properties</i> file in the <i>DataReplication_installation/doc/samples/deploy</i> directory. Enter details for the selected configuration and then save the file at this path. For more information, see "Deployment Properties File" on page 67 .
<i>path_to_save_updated_deploy_properties_file</i>	An absolute path to the directory to which Data Replication writes a copy of the updated deployment properties file after the deployment. The properties file name has the following format: <i>new_configuration_name_deploy.properties</i> .
<i>passwd</i>	A valid password for the login user that is specified in the deployment properties file. Do not specify a password for a local deployment. Optionally, specify a password for a remote deployment. If you do not specify a password in the command for a remote deployment, the Server Manager CLI prompts you to enter the password in interactive mode. If you include the <i>deploy config</i> command name in a script, specify the password before the deployment.
-l	Include this parameter to perform a local deployment. Omit this parameter to perform a remote deployment.

Example

The following example command deploys the *ora2ora* configuration to a remote Server Manager Main server by using the deployment properties in the *ora2ora_deploy.properties* file:

```
deploy config ora2ora F:\IDR_deploy\ora2ora_deploy.properties F:\IDR_deploy\backup\  
passwd=idr_user_password
```

redeploy config

Deploys changes to a replication configuration to a local or remote Server Manager Main server. The redeploy config command is equivalent to the **Partial** deployment type in the Data Replication Console.

Syntax

Enter the command with the required *configuration_name* and *path_to_deploy_properties_file* parameters and any of the optional parameters. The following command syntax includes all of the parameters:

```
redeploy config configuration_name path_to_deploy_properties_file  
[path_to_save_updated_deploy_properties_file] [passwd=password] [-l] [-m] [-r] [-e] [-a]  
[-f]
```

Parameters

The following table describes the command parameters:

Parameter	Description
<i>configuration_name</i>	The name of the replication configuration for which to deploy changes.
<i>path_to_deploy_properties_file</i>	An absolute path to the directory from which Data Replication uploads the deployment properties file that contains deployment information for the selected configuration, including the required <code>prodSrcConnection</code> and <code>prodTgtConnection</code> parameters. For more information, see "Deployment Properties File" on page 67 .
<i>path_to_save_updated_deploy_properties_file</i>	An absolute path to the directory to which Data Replication writes a copy of the updated deployment properties file after deployment. The properties file name has the following format: <i>new_configuration_name_deploy.properties</i> .
passwd	A valid password for the login user that is specified in the deployment settings file. Do not specify a password for local deployment. Optionally specify a password for remote deployment. If you do not specify a password in the command for remote deployment, the Server Manager CLI prompts you to enter the password in interactive mode. If you include the deploy config command name in a script, specify the password before deployment.
-l	Include this parameter to deploy configuration changes to a local Server Manager Main server. Omit this parameter for a remote deployment.
-m	Include this parameter to deploy only metadata changes. These changes include table and column mappings, DDL capture settings, file locations, audit log table settings, and rules for handling Applier errors.
-r	Include this parameter to deploy only changes to the runtime settings.
-e	Include this parameter to deploy only changes to the extract range specified in the <code>deploy.properties</code> file. The extract range settings identify the source database logs.
-a	Include this parameter to deploy all configuration changes. Important: Do not use the -a option together with the -m, -r, and -e options.

Parameter	Description
-f	Include this parameter to stop all running tasks for this configuration in the destination environment.

Note: If you enter the command without any of the -l, -m, -r, -e, -a options, Data Replication deploys only metadata changes by default.

Example

The following example command deploys all changes in the ora2ora configuration to a remote Server Manager Main server by using the deployment properties in the ora2ora_deploy.properties file:

```
redeploy config ora2ora F:\IDR_deploy\ora2ora_deploy.properties F:\IDR_deploy\backup\
passwd=idr_user_password -a
```

Deployment Properties File

Use the deployment properties file to deploy a configuration or configuration changes to a remote Server Manager Main server or locally. To create a deploy.properties file, customize the default deploy.properties file that is located in the *DataReplication_installation/doc/samples/deploy* directory, and then save your customized copy of this file.

The following sections describe the properties in the deploy.properties file.

Server Manager and Configuration Properties

Define deployment properties for connecting to the Server Manager Main server in the destination environment.

The following table describes these deployment properties:

Property	Description
prodSmAddress	The host name or IP address of the system with the Server Manager Main server instance in the destination environment.
prodSmPort	The port number of the Server Manager Main server instance in the destination environment. Valid values are integers from 1 through 65535.
prodSmUser	A user name that has the authority to connect to the specified Server Manager Main server instance in the destination environment.
prodSmSSL	Indicates whether the Server Manager Main server in the destination environment uses HTTPS connections. Valid values are: <ul style="list-style-type: none"> • true. The Main server in the destination environment uses HTTPS connections. • false. The Main server in the destination environment uses HTTP connections.
prodConfig	The name of the replication configuration to use on the destination system with the Server Manager Main server.

Source and Target Connection Properties

Define deployment properties for connecting to the source and target databases and naming the recovery tables in the destination environment.

Note: The properties for connecting to targets and naming the recovery tables have names ending with an *N* value. Each group of these properties in the `deploy.properties` file must use the same *N* value. *N* is a number that reflects the sequence of the target connection definition in the deployment properties file. For the primary target that you specify on the **Target Database** tab in the Data Replication Console, the *N* value is 1. For each secondary target that you specify on the **Routing** tab, the number is incremented by 1. The maximum *N* value is 2147483647.

The following table describes these deployment properties:

Property	Description
<code>prodSrcConnection</code>	The name of the source database connection in the destination environment. Required for a deployment or redeployment.
<code>prodSrcServer</code>	The name of the source Server Manager instance in the destination environment.
<code>prodTgtConnectionN</code>	The name of the target database connection in the destination environment. Required for a deployment or redeployment. If you are deploying a fanout configuration that has one source and multiple targets, this name can be the name of a secondary target database connection.
<code>prodTgtServerN</code>	The name of the target Server Manager instance in the destination environment. If you are deploying a fanout configuration that has one source and multiple targets, this name can be the name of the Server Manager instance on a secondary target.
<code>prodIsPrimaryTargetN</code>	Indicates whether the target database associated with the <code>prodTgtConnectionN</code> and <code>prodTgtServerN</code> properties is a primary target. Valid values are: <ul style="list-style-type: none"> • true. The target is the primary target. • false. The target is a secondary target. If you are deploying a fanout configuration that has one source and multiple target databases, you must use this setting for a secondary target specification.
<code>prodRecoveryTableN</code>	The fully qualified name of the recovery table, including the schema or owner name, for the primary target or a secondary target that is associated with the <code>prodTgtConnectionN</code> and <code>prodTgtServerN</code> properties. The format of the recovery table name depends on the target database type: <ul style="list-style-type: none"> • For Microsoft SQL Server targets, each element in the recovery table name is enclosed in square brackets: <code>[database].[owner].[IDR_RECOVERY]</code> • For all other targets, each element in the recovery table name is enclosed in double quotation marks: <code>"schema"."IDR_RECOVERY"</code> <p>Note: Enclose the recovery table name in double quotation marks if the name is case-sensitive. The double quotation marks are optional if the recovery table name is not case-sensitive.</p>
<code>testTgtConnectionN</code>	If you are deploying a fanout configuration that has one source and multiple target databases, the name of a secondary target database connection in the original environment. This property is not used for the primary target database.

Schema Replacement Rule Properties

Define deployment properties that create rules for replacing the source and target schema names in the original environment with different source and target schema names in the destination environment.

Note: All of the properties for defining a schema replacement rule have names ending with an *N* value. Each group of these properties in the `deploy.properties` file must use the same *N* value. *N* is a number that reflects the sequence of the schema replacement rule in the `deploy.properties` file. For the first replacement rule, the *N* value is 1. For each additional replacement rule, the number is incremented by 1. The maximum *N* value is 2147483647.

The following table describes these deployment properties:

Property	Description
<code>testSrcSchemaN</code>	The source schema name in the original environment.
<code>prodSrcSchemaN</code>	The replacement source schema name to use in the destination environment.
<code>testTgtSchemaN</code>	The target schema name in the original environment.
<code>prodTgtSchemaN</code>	The replacement target schema name to use in the destination environment.
<code>prodTgtPrimaryTargetN</code>	Indicates whether the schema replacement rule is for a primary target in the destination environment. Valid values are: <ul style="list-style-type: none"> • true. The schema replacement rule is for the primary target in the destination environment. • false. The schema replacement rule is for a secondary target in the destination environment.
<code>testConnNameN</code>	The name of the target database connection in the original environment.

Extract Range Properties

Define deployment properties for identifying the source database logs from which to extract data in the destination environment and the source system type.

Note: The properties for identifying the log files have names that begin with "logFile" and end with an *N* value. Each set of logFile properties that define a log file specification in the `deploy.properties` file must use the same *N* value. *N* is a number that reflects the order of the log file specification in the `deploy.properties` file. The first log file specification has an *N* value of 1. For each additional log file specification, the *N* value is incremented by 1.

The following table describes these deployment properties:

Property	Description
<code>logFilesBaseN</code>	The directory that Data Replication searches for the source database log files.
<code>logFilesRecursiveScanN</code>	Indicates whether to recursively search for database log files in the subfolders of the specified <i>N</i> base directory. Valid values are: <ul style="list-style-type: none"> • true. Search recursively for log files in the subfolders of the base directory. • false. Search for log files only in the base directory.

Property	Description
logFileNameN_M	<p>The name of a database log file or a mask for the names of the database log files in the directory specified by the logFilesBaseN property.</p> <p>If you enter a specific log file name, you can repeat this property to specify another log file. In this case, the appended M value is incremented by 1 for each additional log file that you specify. The maximum M value is 2147483647.</p> <p>To define a mask, you can use any of the following wildcard characters: an asterisk (*), question mark (?), or square brackets ([]). For example, to read all files with the .bak extension, enter *.bak. In this case, the property name has an M value of 1 and you cannot enter another mask.</p>
platform	<p>The source operating system type in the destination environment. Valid values are:</p> <ul style="list-style-type: none"> • AIX • HPUX • Linux • SUN • Windows <p>Important: You must specify the platform property if the Extractor processes logs from a remote source database in the destination environment. If the Extractor runs on the source database system, you can delete this property from the deployment properties file.</p>
systemLittleEndian	<p>Indicates whether the source database in the destination environment is a little-endian system. Use this property only if you specify the platform property. Valid values are:</p> <ul style="list-style-type: none"> • true. The source database is a little-endian system. • false. The source database is a big-endian system. <p>Default value:</p> <ul style="list-style-type: none"> • true for Linux and Windows • false for AIX, HP-UX, and Solaris <p>The default value for the platform property is used if you leave the property empty or delete it from the deployment properties file.</p>
systemBlockSize	<p>The block size of the source database system in the destination environment. Use this property only if you specify the platform property.</p> <p>Default values:</p> <ul style="list-style-type: none"> • 1024 for HP-UX • 512 for AIX, Linux, Solaris, and Windows <p>The default value for the platform property is used if you leave the property empty or delete it from the deployment properties file.</p>
diskAlignSize	<p>The disk align size of the source database system in the destination environment. Use this property only if you specify the platform property.</p> <p>Default values:</p> <ul style="list-style-type: none"> • 8 for HP-UX • 4 for AIX, Linux, Solaris, and Windows <p>The default value for the platform property is used if you leave the property empty or delete it from the deployment properties file.</p>
isRealtime	<p>Indicates whether to read online redo logs. Valid values are:</p> <ul style="list-style-type: none"> • true. Capture change data from online redo logs. • false. Do not capture change data from online redo logs. Data Replication can still capture changes from archived redo logs at the location specified in the logFilesBaseN property.

Managing Database Supplemental Logging

Use the Server Manager CLI to manage database supplemental logging for DB2 for Linux, UNIX, and Windows, Microsoft SQL Server, and Oracle sources.

In Data Replication, database *supplemental logging* is the logging of additional information in database log files by a relational source DBMS whenever an SQL change occurs on a source table. Data Replication requires this information to replicate some DDL operations, get the before images and after images for processing Updates, and resolve conflicts. The method of enabling supplemental logging depends on the source type:

- For DB2 for Linux, UNIX, and Windows sources, supplemental logging is enabled by setting DATA CAPTURE CHANGES for each source table.
- For Microsoft SQL Server sources, supplemental logging is enabled by setting Change Data Capture for each source table.
- For Oracle sources, supplemental logging is enabled by adding columns to supplemental log groups for source tables.

You can manage database supplemental logging from the Data Replication Console and the Server Manager CLI. With Server Manager CLI commands, you can perform the following tasks:

- Open the replication configuration that contains the source tables for which you want to issue commands for managing supplemental logging.
- Enable database supplemental logging for source tables or columns in the following ways:
 - For DB2 source databases, set DATA CAPTURE CHANGES for additional tables.
 - For Microsoft SQL Server databases, enable Change Data Capture for additional tables.
 - For Oracle source tables, add columns to a supplemental log group.
- Disable database supplemental logging for source tables or columns in the following ways:
 - For DB2 source tables, set DATA CAPTURE NONE.
 - For Microsoft SQL Server tables, disable Change Data Capture.
 - For Oracle source tables, remove columns from a supplemental log group.
- Show the DB2 or SQL Server source tables or Oracle columns for which database supplemental logging is enabled.
- Refresh the source database tables to reflect any changes that users made to the database options that enable or disable database supplemental logging.
- Cancel the suplog add and suplog remove commands before you run the generated DDL statements that enable or disable database supplemental logging.
- Run or save the generated DDL statements for enabling or disabling database supplemental logging.

By default, the Server Manager CLI commands perform these tasks for the mapped source tables. Some of the suplog commands also use optional parameters to perform these tasks for unmapped source tables.

Before you use the suplog commands, configure the following parameters in the *DataReplication_installation/uiconf/default.cfg* file on the computer where you use the Server Manager CLI:

- Specify the following parameter to automatically use default values and cause the Server Manager CLI to stay in noninteractive mode:

```
use_default_values_during_saving=true
```

The Server Manager CLI needs to run in noninteractive mode to process the commands for managing database supplemental logging.

- For DB2 and Oracle sources, if the target tables that you mapped in the configuration do not have primary keys, Data Replication creates a virtual index for these tables, by default, to properly process Updates and Deletes. The index contains all of the table columns. Alternatively, in the *DataReplication_installation/uiconf/default.cfg* file, you can specify the following parameter to disable the replication of Update and Delete operations:

```
pk_warning_option parameter=2
```

With this parameter set to 2, Data Replication replicates only Insert operations. Informatica recommends this setting to process the commands for managing database supplemental logging. The default value is 3.

Note: The *DataReplication_installation/uiconf/default.cfg* file is also used by the Replication Configuration CLI.

suplog add

Enables database supplemental logging for DB2 for Linux, UNIX, and Windows, Microsoft SQL Server, and Oracle sources.

- For DB2 sources, marks the tables for which you want to set DATA CAPTURE CHANGES.
- For Microsoft SQL Server sources, marks the tables for which you want to enable Change Data Capture.
- For Oracle sources, marks the columns in a source table that you want to add to a supplemental log group.

Restriction: You cannot include columns that have the following datatypes in an Oracle supplemental log group: BLOB, CLOB, LONG, LONG RAW, NCLOB, RAW, and the Oracle datatypes that Data Replication does not support.

Important: The suplog add command only marks the database supplemental logging settings that you want to enable. To persist the changes, run the suplog run command. Alternatively, you can run the suplog save command to save the settings to a script file that your DBA can run later.

Before you can run this command, you must run the suplog open command.

After the suplog add command runs successfully, the Server Manager CLI lists the number of updated objects.

Syntax

For DB2 and Microsoft SQL Server sources, use the following syntax:

```
suplog add source_schema.source_table [-a]
```

For Oracle sources, use the following syntax:

```
suplog add source_schema.source_table [-a] [columns=column_name1,column_name2,...]
```

Parameters

The following table describes the command parameters for DB2 and Microsoft SQL Server sources:

Parameter	Description
<i>source_schema</i>	The name of the source schema or a mask for the name of the source schema or schemas. You can use the following wildcard characters in a mask: <ul style="list-style-type: none"> • An asterisk (*) to match all characters • A question mark (?) to match any single character • Square brackets ([]) to match only one out of several characters

Parameter	Description
<i>source_table</i>	For DB2 sources, the name of the source table or a mask for the name of the source table or tables. For SQL Server sources, the owner name and source table name, or a mask for owner and table names. You can use the following wildcard characters in a mask: <ul style="list-style-type: none"> • An asterisk (*) to match all characters • A question mark (?) to match any single character • Square brackets ([]) to match only one out of several characters
-a	For DB2 sources, include this parameter to set DATA CAPTURE CHANGES for the table or tables that match the <i>source_schema</i> and <i>source_table</i> parameter values and that are <i>not</i> mapped in the open configuration. For SQL Server sources, include this parameter to enable Change Data Capture for the table or tables that match the <i>source_schema</i> and <i>source_table</i> parameter values and that are <i>not</i> mapped in the open configuration.

The following table describes the command parameters for Oracle sources:

Parameter	Description
<i>source_schema</i>	The name of the source schema. No wildcards are allowed.
<i>source_table</i>	The name of the source table. No wildcards are allowed.
-a	If the table that matches the <i>source_schema</i> and <i>source_table</i> parameter values is <i>not</i> mapped in the open configuration, use this parameter to add the columns in the source table to a supplemental log group.
columns	A comma-separated list of the columns to add to the supplemental log group or a mask for these columns. You can use the following wildcard characters in a mask: <ul style="list-style-type: none"> • An asterisk (*) to match all characters • A question mark (?) to match any single character • Square brackets ([]) to match only one out of several characters If you omit this parameter, the command adds all of the columns to the supplemental log group.

Usage Notes

For Oracle sources, the Server Manager generates a supplemental log group when you use this command to add columns to a group. If you add over 33 columns, the Server Manager generates an additional supplemental log group.

To name the generated supplemental log groups, Data Replication uses the following naming pattern:

```
IDR_object_ID_sequence_number
```

The *object_ID* is the object ID of the mapped source table, and the *sequence_number* is a generated sequence number for the supplemental log group.

Examples

For a DB2 source, the following command sets DATA CAPTURE CHANGES for all mapped and unmapped tables in all schemas:

```
supplog add *.* -a
The supplog add command executed successfully. Updated 1000 object(s).
```

For a Microsoft SQL Server source, the following command enables Change Data Capture for the mapped "tab1" table that has the "src" schema and "dbo" owner:

```
supplog add src.dbo.tab1
The supplog add command executed successfully. Updated 1 object(s).
```

For an Oracle source, the following command adds all columns in the mapped "tab1" table to a supplemental log group:

```
supplog add src.tab1
The supplog add command executed successfully. Updated 8 object(s).
```

supplog cancel

Cancels all supplog add and supplog remove commands that were issued before a supplog run command.

Before you can run this command, you must run the supplog open command.

Important: You cannot cancel these commands after you run the supplog run command.

Syntax

Use the following syntax:

```
supplog cancel
```

supplog open

Opens the replication configuration for which you want to manage database supplemental logging with other supplog commands.

You must run the supplog open command before running the supplog add, supplog cancel, supplog remove, supplog run, supplog save, and supplog show commands.

Syntax

Use the following syntax:

```
supplog open configuration_name
```

Parameters

The following table describes the command parameter:

Parameter	Description
<i>configuration_name</i>	The name of the replication configuration that you want to open to manage database supplemental logging for source tables.

Example

The following command opens the Ora2Ora configuration:

```
supplog open Ora2Ora
```

supplog refresh

Refreshes the database supplemental logging settings directly from the source database. For Microsoft SQL Server sources, also refreshes the Server Manager cache.

Use this command to determine if changes have been made to the DATA CAPTURE option for DB2 tables, the Change Data Capture setting for SQL Server tables, or Oracle supplemental log groups.

Before you can run this command, you must run the `supplog open` command.

Syntax

Use the following syntax:

```
supplog refresh source_schema [-a]
```

Parameters

The following table describes the command parameters:

Parameter	Description
<code>source_schema</code>	The name of the source schema or a mask for the name of the source schema. You can use the following wildcard characters in a mask: <ul style="list-style-type: none">• An asterisk (*) to match all characters• A question mark (?) to match any single character• Square brackets ([]) to match only one out of several characters
<code>-a</code>	Refresh all mapped and unmapped tables in the source schema. By default, the <code>supplog refresh</code> command refreshes only the mapped source tables.

Examples

The following command refreshes all mapped tables in all schemas in the source database:

```
supplog refresh *
```

The following command refreshes all mapped and unmapped tables in the `src` schema:

```
supplog refresh src -a
```

supplog remove

Disables database supplemental logging for DB2 for Linux, UNIX, and Windows, Microsoft SQL Server, and Oracle sources.

- For DB2 sources, marks the tables for which you want to set `DATA CAPTURE NONE`.
- For Microsoft SQL Server sources, marks the tables for which you want to disable Change Data Capture.
- For Oracle sources, marks the columns in a source table that you want to remove from a supplemental log group.

Important: The `supplog remove` command only marks the database supplemental logging settings that you want to disable. To persist the changes, run the `supplog run` command. Alternatively, you can run the `supplog save` command to save the settings to a script file that your DBA can run later.

Before you can run this command, you must run the `supplog open` command.

After the `supplog remove` command runs successfully, the Server Manager CLI lists the number of updated objects.

Syntax

For DB2 and Microsoft SQL Server sources, use the following syntax:

```
supplog remove source_schema.source_table [-a]
```

For Oracle sources, use the following syntax:

```
supplog remove source_schema.source_table [-a] [columns=column_name1,column_name2,...]
```

Parameters

The following table describes the command parameters for DB2 and Microsoft SQL Server sources:

Parameter	Description
<i>source_schema</i>	The name of the source schema or a mask for the name of the source schema or schemas. You can use the following wildcard characters in a mask: <ul style="list-style-type: none">• An asterisk (*) to match all characters• A question mark (?) to match any single character• Square brackets ([]) to match only one out of several characters
<i>source_table</i>	For DB2 sources, the name of the source table or a mask for the name of the source table or tables. For SQL Server sources, the owner name and the source table name, or a mask for the owner and table names. You can use the following wildcard characters in a mask: <ul style="list-style-type: none">• An asterisk (*) to match all characters• A question mark (?) to match any single character• Square brackets ([]) to match only one out of several characters
-a	For DB2 sources, include this parameter to set DATA CAPTURE NONE for the table or tables that match the <i>source_schema</i> and <i>source_table</i> parameter values and that are <i>not</i> mapped in the open configuration. For SQL Server sources, include this parameter to disable Change Data Capture for the table or tables that match the <i>source_schema</i> and <i>source_table</i> parameter values and that are <i>not</i> mapped in the open configuration.

The following table describes the command parameters for Oracle sources:

Parameter	Description
<i>source_schema</i>	The name of the source schema. No wildcards are allowed.
<i>source_table</i>	The name of the source table. No wildcards are allowed.
-a	If the table that matches the <i>source_schema</i> and <i>source_table</i> parameter values is <i>not</i> mapped in the open configuration, use this parameter to remove the columns in the source table from a supplemental log group.
columns	A comma-separated list of the columns that you want to remove from a supplemental log group that Data Replication created, or a mask for these columns. You can use the following wildcard characters in a mask: <ul style="list-style-type: none">• An asterisk (*) to match all characters• A question mark (?) to match any single character• Square brackets ([]) to match only one out of several characters If you omit this parameter, this command removes all of the columns from the supplemental log group. Note: Removing all columns also removes any primary key or unique index columns from the supplemental log group. Informatica does not recommend removing all columns from supplemental log groups that contain primary keys or unique indexes.

Examples

For a DB2 source, the following command sets DATA CAPTURE NONE for all mapped and unmapped tables in all schemas:

```
supplog remove *.* -a
The supplog remove command executed successfully. Updated 1000 object(s).
```

For a Microsoft SQL Server source, the following command disables Change Data Capture for the mapped "tab1" table that has the "src" schema and "dbo" owner:

```
supplog remove src.dbo.tab1
The supplog remove command executed successfully. Updated 1 object(s).
```

For an Oracle source, the following command removes the "col1" column in the "tab1" table from a supplemental log group:

```
supplog remove src.tab1 columns=col1
The supplog remove command executed successfully. Updated 1 object(s).
```

supplog run

Executes the DDL statements that the supplog add and supplog remove commands generate to enable or disable database supplemental logging.

Important: Ensure that you have the authority to run the DDL statements. If you do not have the required authority, use the supplog save command to generate a script file that you can give to your DBA to run.

Syntax

For DB2 and Oracle sources, use the following syntax:

```
supplog run
```

For Microsoft SQL Server Enterprise Edition sources, use the following syntax:

```
supplog run [restart_instance={0|1}] [cdc_disable={0|1|2}]
```

For Microsoft SQL Server Standard Edition sources, use the following syntax:

```
supplog run [-r] [cdc_disable={0|1|2}]
```

Parameter

The following table describes the command parameters for Microsoft SQL Server sources:

Parameter	Description
cdc_disable	<p>Indicates how to handle the CDC capture and cleanup jobs that SQL Server generates for databases with Change Data Capture enabled. Valid values are:</p> <ul style="list-style-type: none">• 0. Do not disable the SQL Server CDC jobs. Specify this option if you do not want Data Replication to disable the SQL Server CDC jobs because the jobs are already disabled from Microsoft SQL Server Management Studio or you accept that Data Replication might not capture all of the change data.• 1. Disable the SQL Server CDC jobs if SQL Server Change Data Capture is not enabled for any source tables when you create the configuration in Data Replication. If SQL Server Change Data Capture is enabled for any source table, the SQL Server CDC jobs are not disabled.• 2. Disable the SQL Server CDC jobs regardless of whether SQL Server Change Data Capture is enabled or not enabled for the source tables. <p>Note: After you run the supplog run command with the cdc_disable parameter, the Server Manager CDCJobsDisableMode advanced property is updated to the equivalent setting.</p>

Parameter	Description
restart_instance	<p>Indicates whether the Server Manager restarts the SQL Server Enterprise Edition instance in single-user mode to enable a dedicated administrator connection to the source database when you edit the Change Data Capture settings for source tables. Valid values are:</p> <ul style="list-style-type: none"> • 0. Restart the SQL Server instance in single-user mode only when you edit the Change Data Capture settings for source tables. • 1. Always restart the SQL Server instance in single-user mode. <p>Note: After you run the supalog run command with the restart_instance parameter, the Server Manager RestartMSSQLInstance advanced property is updated to the equivalent setting.</p>
-r	Allows the Server Manager to restart Microsoft SQL Server Standard Edition sources to change the Change Data Capture setting. Use this parameter in non-interactive mode only.

supalog save

Saves the DDL statements that the supalog add and supalog remove commands generate for enabling or disabling database supplemental logging to a script file.

Use this command to create a script file that you can give to your DBA when you do not have the authority to run the DDL against the source tables by using the supalog run command.

Syntax

Use the following syntax:

```
supalog save path_to_sql_script
```

Parameter

The following table describes the command parameter:

Parameter	Description
<i>path_to_sql_script</i>	The path to the SQL script file to which you want to save the DDL statements that enable or disable database supplemental logging for DB2 and SQL Server sources or that add or remove columns from a supplemental log group for Oracle sources. You can specify an absolute path or a path that is relative to the <i>DataReplication_installation</i> directory.

Example

The following command saves the DDL statements that modify Oracle supplemental log groups to the ora2ora_supalog.sql file:

```
supalog save C:\ora2ora_supalog.sql
```

supalog show

For the source tables in the open configuration, shows the tables or columns for which database supplemental logging is not enabled, or shows all mapped and unmapped source tables.

- For a DB2 source, shows the tables with DATA CAPTURE CHANGES set.
- For a Microsoft SQL Server source, shows the tables with Change Data Capture enabled.
- For an Oracle source, shows the columns that are included in supplemental log groups that Data Replication created and the primary keys and indexes for the source tables.

Before you can run this command, you must run the supalog open command.

Note: For Oracle sources, the `supplog show` command does not show the columns in supplemental log groups that were created outside of Data Replication.

Syntax

For DB2 and Microsoft SQL Server sources, use the following syntax:

```
supplog show schema_mask.table_mask [-t] [-a]
```

For Oracle sources, use the following syntax:

```
supplog show schema_mask.table_mask [-t] [-a] [-c]
```

Parameters

The following table describes the command parameters:

Parameter	Description
<i>source_schema</i>	A mask for the name of the source schema or schemas. You can use the following wildcard characters in the mask: <ul style="list-style-type: none">• An asterisk (*) to match all characters• A question mark (?) to match any single character• Square brackets ([]) to match only one out of several characters
<i>source_table</i>	A mask for the name of the source table or tables. You can use the following wildcard characters in the mask: <ul style="list-style-type: none">• An asterisk (*) to match all characters• A question mark (?) to match any single character• Square brackets ([]) to match only one out of several characters
-t	For DB2 and SQL Server sources, shows only the tables that do not have database supplemental logging enabled. For Oracle sources, shows only the tables with columns that are not in a supplemental log group that Data Replication created.
-a	Shows all mapped and unmapped tables for the specified source schema.
-c	For Oracle sources, shows only the columns in the specified source table or tables that are not included in a supplemental log group that Data Replication created.

Examples

For an Oracle source, the following command shows the columns that are not in a supplemental log group for the "tab1" table in the "src" schema:

```
supplog show src.tab1 -c
```

The following command shows all mapped and unmapped tables in the "src" schema:

```
supplog show src.* -a
```

Managing Change Data Capture for Microsoft SQL Server Sources

Use the Server Manager Command Line Interface to manage Change Data Capture (CDC) settings for Microsoft SQL Server sources.

Commands are available to perform the following tasks:

- View the Microsoft SQL Server instances that run on the Server Manager system.
- View the Change Data Capture settings for individual SQL Server databases and tables.
- Enable or disable Change Data Capture for SQL Server tables.

get msdb list

Lists the Microsoft SQL Server databases for a specific SQL Server instance that runs on the Server Manager system, including the Change Data Capture settings for those databases.

Syntax

```
get msdb list Server_Manager_name SQLServer_instance_name [mode={on|off|mixed|error}]  
[pattern=pattern]
```

The following short form is acceptable:

```
msdl Server_Manager_name SQLServer_instance_name [mode={on|off|mixed|error}]  
[pattern=pattern]
```

Parameters

The following table describes the command parameters:

Parameter	Description
Server_Manager_name	The name of the Server Manager instance that runs on the Microsoft SQL Server system.
SQLServer_instance_name	The name of the Microsoft SQL Server instance for which to list databases and their Change Data Capture settings.
mode	A Change Data Capture setting to use for filtering the list of databases. Valid values are: <ul style="list-style-type: none">• on. Change Data Capture is enabled for all of the tables in the database.• off. Change Data Capture is disabled for all of the tables in the database.• mixed. The database includes tables with different Change Data Capture settings.• error. The Server Manager encountered an error when retrieving the Change Data Capture settings for the database. By default, the command lists all of the databases for the specified instance, regardless of their Change Data Capture setting.
pattern	A comma-, semicolon-, or space-separated list of wildcard patterns that the Server Manager matches against database names to filter the databases for which to view the Change Data Capture settings. You can use the asterisk (*) and question mark (?) wildcards in the patterns. By default, the command lists all databases.

Examples

The following command lists databases of the MSSQLSERVER instance that runs on the Server Manager Main server system and the Change Data Capture settings for those databases:

```
get msdb list Main MSSQLSERVER
```


The following command lists databases of the same instance that have names beginning with DB_ and a Change Data Capture setting of **on**:

```
msdl Main MSSQLSERVER pattern=DB_* mode=on
```

get msinstance list

Lists the Microsoft SQL Server instances that run on a specific Server Manager system.

Syntax

```
get msinstance list Server_Manager_name
```

The following short form is acceptable:

```
msil Server_Manager_name
```

Parameters

The following table describes the command parameter:

Parameter	Description
Server_Manager_name	The name of the Server Manager instance on the system for which to list Microsoft SQL Server instances. The Server Manager instance must belong to the replication server group of the Main server to which you are connected. Tip: Use the <code>show servers</code> command to view a list of the Server Manager instances in the replication server group.

Examples

The following command displays a list of the Microsoft SQL Server instances that run on the Server Manager Main server system:

```
get msinstance list Main
```

The following command displays a list of the Microsoft SQL Server instances that run on the Subserver_src subserver system:

```
msil Subserver_src
```

get mstable list

Lists the Microsoft SQL Server tables in a specific SQL Server database, including their Change Data Capture settings. You can filter the list of tables based on Change Data Capture settings and on wildcard patterns that you define for table names.

Syntax

```
get mstable list Server_Manager_name SQLServer_instance_name database_name [mode={on|off|error}] [pattern=pattern]
```

The following short form is acceptable:

```
mstl Server_Manager_name SQLServer_instance_name database_name [mode={on|off|error}] [pattern=pattern]
```

Parameters

The following table describes the command parameters:

Parameter	Description
Server_Manager_name	The name of the Server Manager instance that runs on the Microsoft SQL Server system.
SQLServer_instance_name	The name of the Microsoft SQL Server instance.
database_name	The name of the Microsoft SQL Server database for which to list tables.
mode	A Change Data Capture setting to use for filtering the list of tables. Valid values are: <ul style="list-style-type: none">• on. Change Data Capture is enabled.• off. Change Data Capture is disabled.• error. Tables for which the Server Manager cannot retrieve the Change Data Capture setting because of an error. By default, the command lists all of the tables, regardless of their Change Data Capture settings.
pattern	A comma-, semicolon-, or space-separated list of wildcard patterns that the Server Manager matches against table names to filter the tables for which to view the Change Data Capture settings. You can use the asterisk (*) and question mark (?) wildcards in the patterns. By default, the command lists all tables in the database.

Examples

The following command displays an unfiltered list of tables in the DB_Source database of the MSSQLSERVER instance that runs on the Server Manager Main server system:

```
get mstable list Main MSSQLSERVER DB_Source
```

The following command lists the tables in the DB_Source database that have table names beginning with Table_ and a Change Data Capture setting of **off**:

```
mstl Main MSSQLSERVER DB_Source pattern=Table_* mode=off
```

set replication mode

Enables or disables Change Data Capture for Microsoft SQL Server tables in a specific database. You can enable Change Data Capture for all of the database tables or for a subset of the tables.

Syntax

```
set replication mode Server_Manager_name SQLServer_instance_name database_name  
[mode=state] [pattern=pattern]
```

The following short form is acceptable:

```
srm Server_Manager_name SQLServer_instance_name database_name [mode=state]  
[pattern=pattern]
```

Parameters

The following table describes the command parameters:

Parameter	Description
Server_Manager_name	The name of the Server Manager instance that runs on the Microsoft SQL Server system.

Parameter	Description
SQLServer_instance_name	The name of the Microsoft SQL Server instance.
database_name	The name of the Microsoft SQL Server database.
mode	Specifies the Change Data Capture settings for the specified tables. Valid values are: <ul style="list-style-type: none"> • on. Enable Change Data Capture. • off. Disable Change Data Capture. By default, if this parameter is not specified, Change Data Capture is enabled.
pattern	A comma-, semicolon-, or space-separated list of wildcard patterns that the Server Manager matches against table names to filter the tables for which to set Change Data Capture. You can use the asterisk (*) and question mark (?) wildcards in the patterns. By default, the command sets Change Data Capture for all tables in the database.

Examples

The following command enables Change Data Capture for all of the tables in the DB_Source database of the MSSQLSERVER instance that runs on the Server Manager Main server system:

```
set replication mode Main MSSQLSERVER DB_Source
```

The following command disables Change Data Capture for all of the tables in the DB_Source database of the MSSQLSERVER instance that runs on the Server Manager Main server system:

```
srm Main MSSQLSERVER DB_Source mode=off
```

Managing User Notifications

Use the Server Manager Command Line Interface to configure email and SNMP notifications.

Commands are available to perform the following tasks:

- Add an email subscriber.
- Add an SNMP subscriber.
- Configure the Server Manager email server (SMTP) settings.
- Configure the Server Manager SNMP settings.

add email subscriber

Adds a subscriber of email notifications of various Server Manager events.

Syntax

```
add email subscriber email_address user_name [task=task_options]  
[schedule=schedule_options] [sm=type_options]
```

The following short form is acceptable:

```
aes email_address user_name [task=task_options] [schedule=schedule_options]  
[sm=type_options]
```

Parameters

The following table describes the command parameters:

Parameter	Description
email_address	The email address of the new subscriber who will receive Server Manager notifications.
user_name	The Data Replication user name for the subscriber with the specified email address.
task	The types of Server Manager tasks and events for which the subscriber can receive email notifications. Options are: <ul style="list-style-type: none">• end• error• recovery• restart• start• warning Note: Use commas, spaces, or semicolons to separate multiple options. Enclose a space-separated list in single quotation marks.
schedule	The Server Manager schedule events for which the subscriber can receive email notifications. Options are: <ul style="list-style-type: none">• end• error• start Note: Use commas, spaces, or semicolons to separate multiple options. Enclose a space-separated list in single quotation marks.
sm	The types of Server Manager notifications that the subscriber user can receive. Options are: <ul style="list-style-type: none">• error• information• warning Note: Use commas, spaces, or semicolons to separate multiple options. Enclose a space-separated list in single quotation marks.

Example

The following command adds the idradadmin user as a subscriber of email notifications about Server Manager errors and warnings:

```
add email subscriber admin@mycompany.com idradadmin sm='error warning'
```

add snmp subscriber

Adds a subscriber of Simple Network Management Protocol (SNMP) notifications of various types of Server Manager events. The notifications are sent to a specific IP address.

Syntax

```
add snmp subscriber IP_address user_name [oid=object_ID] [task=task_options]
[schedule=schedule_options] [sm=server_manager_options]
```

Short form:

```
asnmpps IP_address user_name [oid=object_ID] [task=task_options]
[schedule=schedule_options] [sm=server_manager_options]
```

To use a different SNMP OID for each event type, use the following syntax instead of specifying the oid parameter:

```
[task=task_option1:object_ID1;task_option2:object_ID2]
[schedule=schedule_option1:object_ID3;schedule_option2:object_ID4]
[sm=server_manager_option1:object_ID5;server_manager_option2:object_ID6]
```

To enter multiple options in the task, schedule, or sm parameter, use a comma, semicolon, or space separator. If you use a space separator, enclose the entire string of options in single quotation marks. For example:

```
task=task_option1:object_ID1,task_option2:object_ID2
schedule='schedule_option1:object_ID1 schedule_option2:object_ID2'
```

Parameters

The following table describes the command parameters:

Parameter	Description
IP_address	The IP address to which Data Replication will send Server Manager notifications by using SNMP.
user_name	The Data Replication user name for the subscriber.
oid	The object ID for all of the event types that you specify in the task, schedule, and sm parameters. If you want to specify different OIDs for individual task, schedule, and sm events, do not use this parameter in the command syntax. Instead, include the OID value in the event option specification in the task, schedule and sm parameters. Object IDs for events must start with the Enterprise Object ID that is specified in the SNMP community OID field on the General tab in the Data Replication Console.
task	The types of task events for which the subscriber can receive SNMP notifications. Options are: <ul style="list-style-type: none"> • end • error • execution • recovery • restart • start • warning
schedule	The types of schedule events for which the subscriber can receive SNMP notifications. Options are: <ul style="list-style-type: none"> • end • error • start
sm	The types of Server Manager events for which the subscriber can receive SNMP notifications. Options are: <ul style="list-style-type: none"> • error • information • warning

Examples

The following command adds the idradadmin user as a subscriber who receives SNMP notifications about Server Manager errors and warnings at the specified IP address:

```
add snmp subscriber 10.51.4.120 idradadmin sm='error warning'
```

The following command adds the idradmin user as a subscriber who receives SNMP notifications about task error and warning events and Server Manager error and warning events at the specified IP address:

```
add snmp subscriber 10.51.4.120 idradmin task=error:1.3.6.1.2.1.1.1.2.0.1.7;warning:
1.3.6.1.2.1.1.1.2.0.1.10 sm=error:1.3.6.1.2.1.1.1.2.0.1.13;warning:
1.3.6.1.2.1.1.1.2.0.1.12
```

Each event type is associated with a different OID.

configure smtp

Specifies the SMTP Server settings for the Server Manager.

Syntax

Enter the command with the path to a local SMTP Server settings file or enter the command with the individual parameters:

```
configure smtp path_to_a_local_file

configure smtp [server=server] [port=port] [login=login] [passwd=password] [email=email]
[mls=max_log_size] [mas=max_attachment_size] [bs=block_size] [ssl=ssl_option]
[vhost=verify_host_cert] [vpeer=verify_peer_cert]
```

The following short forms are acceptable:

```
smtpconf path_to_a_local_file

smtpconf [server=server] [port=port] [login=login] [passwd=password] [email=email]
[mls=max_log_size] [mas=max_attachment_size] [bs=block_size] [ssl=ssl_option]
[vhost=verify_host_cert] [vpeer=verify_peer_cert]
```

Parameters

The following table describes the command parameters:

Parameter	Description
path_to_a_local_file	The path to a local SMTP Server settings file.
server	The IP address or host name of the SMTP Server that the Server Manager uses to send notifications.
port	The SMTP Server port number that the Server Manager uses to send email notifications. Valid values are integers from 1 through 65535.
login	The user login for the SMTP Server.
passwd	The password for the SMTP Server.
email	The email address that the Server Manager uses to send email notifications.
mls	The maximum log file size, in bytes, that the Server Manager can paste into the email body. If the log file size is larger than this parameter value, the mas parameter is used instead.
mas	The maximum log file size, in bytes, that the Server Manager can attach to an email notification. If the log file size is larger than this parameter value, the bs parameter is used instead.
bs	The block size, in bytes, that the Server Manager takes from the start and end of the log file storage and pastes into an email notification. The intervening bytes of blocked storage for the middle of the log file are not included. This parameter is used only if the log file size is larger than both the mls and mas parameter limits.

Parameter	Description
ssl	Indicates whether the SMTP Server uses the SSL protocol to send email notifications. Options are: <ul style="list-style-type: none"> • 1 or true. Use the SSL protocol. • 0 or false. Do not use the SSL protocol. Default is false.
vhost	Verifies that the SMTP Server host name or IP address that is specified in the SSL certificate matches the SMTP server value that is specified on the Subscribers tab > General subtab in the Data Replication Console. Options are: <ul style="list-style-type: none"> • 1 or true. Verify that the certificate is associated with the correct host name or IP address. • 0 or false. Do not verify that the certificate is associated with the correct host name or IP address. Default is false.
vpeer	Verifies that the SSL certificate is authentic. Options are: <ul style="list-style-type: none"> • 1 or true. Verify that the SSL certificate is authentic. • 0 or false. Do not verify that the SSL certificate is authentic. Default is false.

Examples

The following command uses the `smtp_settings` file in the `C:\notifications` directory to specify the SMTP Server settings:

```
configure smtp C:\notifications\smtp_settings.txt
```

The following command uses individual parameters to specify the SMTP Server settings for a server that does not use SSL protocol to send email notifications to the `idradadmin` user:

```
configure smtp [server=10.51.4.120] [port=9088] [login=idradadmin] [passwd=]
[email=admin@mycompany.com] [mls=500000] [mas=5000000] [bs=100000] [ssl=0] [vhost=0]
[vpeer=0]
```

configure snmp

Specifies the SNMP settings for the Server Manager.

Syntax

```
configure snmp [oid=object_ID] [comm=community_name]
```

The following short form is acceptable:

```
snmpconf [oid=object_ID] [comm=community_name]
```

Parameters

The following table describes the command parameters:

Type	Description
oid	An override for the SNMP community object identifier (OID). An enterprise OID override consists of a series of digits that are separated by periods. Important: Contact your system administrator to ensure that the SNMP community OID value does not conflict with enterprise OIDs of other SNMP agents in your network.

Type	Description
comm	An override for the default SNMP community string. Data Replication sends SNMP traps to the Informatica community by default. If you use the default value, ensure that the SNMP manager that you configure to receive SNMP notifications accepts SNMP traps from this community. Note: Most SNMP managers use the community name of "public" by default.

Example

The following command specifies the SNMP settings for the "public" community:

```
configure snmp oid=1.3.6.1.2.1.1.1.2.0.1 comm=public
```

Auxiliary Commands

Use the auxiliary commands to display help and to exit interactive mode.

Commands are available to perform the following tasks:

- Display a list of commands that the Server Manager Command Line Interface supports.
- Display help information about a specific command.
- Exit interactive mode.

exit

Exits interactive mode.

Syntax

```
exit
```

help

Displays a list of commands that the Server Manager Command Line Interface supports or detailed information about a specific command.

Syntax

```
help [command]
```

Parameters

The following table describes the command parameter:

Parameter	Description
<i>command</i>	The full or short name of the command for which to display detailed information.

Examples

The following example command displays a list of the commands that the Server Manager Command Line Interface supports:

```
help
```

The following example command displays detailed information about the get task detail command:

```
help get task detail
```

INDEX

A

auxiliary commands
 exit [88](#)
 help [88](#)

C

Change Data Capture for Microsoft SQL Server sources
 commands for managing Change Data Capture [80](#)
command line interface
 idrcmd Application [9](#)
 overview [8](#)
commands
 abort schedule [43](#)
 abort task [38](#)
 add connection [52](#)
 add email subscriber [83](#)
 add server [31](#)
 add snmp subscriber [84](#)
 add target [33](#)
 configure smtp [86](#)
 configure snmp [87](#)
 connect [16](#)
 create config [19](#)
 create schedule [44](#)
 delete config [21](#)
 delete connection [55](#)
 delete schedule [46](#)
 delete server [31](#)
 delete task [38](#)
 deploy config [65](#)
 disconnect [17](#)
 download config [22](#)
 edit connection [56](#)
 edit schedule [46](#)
 exit [88](#)
 export config [22](#)
 export scn [23](#)
 force transaction [60](#)
 get msdb list [80](#)
 get msinstance list [81](#)
 get mstable list [81](#)
 get scn [24](#)
 get target list [34](#)
 get task detail [39](#)
 help [88](#)
 import config [25](#)
 import scn [27](#)
 ping [18](#)
 redeploy config [66](#)
 rename schema [34](#)
 restore task [39](#)
 set conflict logging path [35](#)
 set log table schema [36](#)

commands (*continued*)
 set replication mode [82](#)
 set replication type [36](#)
 set scn [28](#)
 show configs [29](#)
 show connection info [59](#)
 show connections [60](#)
 show logs [50](#)
 show open transactions [61](#)
 show processed logs [29](#)
 show running schedules [48](#)
 show schedule info [48](#)
 show schedules [49](#)
 show servers [32](#)
 show task history [40](#)
 show task list [41](#)
 show tasks [41](#)
 show version [32](#)
 start log watcher [51](#)
 start schedule [49](#)
 start task [41](#)
 stop schedule [50](#)
 stop task [42](#)
 supplg add [72](#)
 supplg open [74](#)
 supplg refresh [74](#)
 supplg remove [75](#)
 supplg run [74, 77](#)
 supplg save [78](#)
 supplg show [78](#)
 update config [30](#)
configurations
 commands for managing replication configurations [19](#)
 considerations for deployment [64](#)
 creating a replication configuration [12](#)
 deployment overview [63](#)
 deployment properties file [67](#)
connections [52](#)
connections, Server Manager
 commands for creating and managing connections [16](#)

D

deploying configurations
 deployment properties file [67](#)
deployment
 considerations [64](#)
 overview [63](#)

I

idrcmd application
 sample script for starting in non-interactive mode [10](#)
 starting in interactive mode [9](#)

- idrcmd application (*continued*)
 - starting in non-interactive mode [10](#)
 - use in interactive mode [9](#)
 - use in non-interactive mode [10](#)
- interactive mode
 - described [9](#)
 - running replications in interactive mode [13](#)
 - starting the idrcmd application in interactive mode [9](#)

M

- managing Server Manager instances
 - commands for managing servers [31](#)
- Microsoft SQL Server sources
 - commands for managing Change Data Capture [80](#)
- multiple targets configuration
 - commands for configuring replication from one source to multiple targets [33](#)

N

- non-interactive mode
 - described [10](#)
 - running replications in non-interactive mode [14](#)
 - sample script for starting the idrcmd application [10](#)
 - starting the idrcmd application [10](#)

P

- prerequisite tasks for replication jobs [12](#)

R

- replication configurations
 - commands for managing configurations [19](#)
 - considerations for deployment [64](#)
 - creating a configuration [12](#)
 - deployment overview [63](#)
- replication jobs
 - prerequisite tasks [12](#)

- replication jobs (*continued*)
 - running replications in interactive mode [13](#)
 - running replications in non-interactive mode with scripts [14](#)
- replication schedules
 - commands for managing schedules [43](#)
- replication tasks
 - commands for managing tasks [37](#)
- replication to multiple targets
 - commands for configuring replication from one source to multiple targets [33](#)

S

- schedules, replication
 - commands for managing schedules [43](#)
- scripts
 - creating scripts to run replications in non-interactive mode [14](#)
- Server Manager connections
 - commands for creating and managing connections [16](#)
- Server Manager logs
 - commands for viewing logs [50](#)
- Server Manager management
 - commands for managing servers [31](#)
- Server Manager users
 - privileges [11](#)
- supplemental logging
 - commands for managing supplemental logging [71](#)

T

- third-party software
 - calling scripts that run replications in non-interactive mode [14](#)

U

- User notifications
 - commands for managing notification settings [83](#)