Informatica Fast Clone (Version 9.7.0)

# User Guide

Informatica Fast Clone User Guide

Version 9.7.0
December 2014

DISCLAIMER: Informatica Corporation provides this documentation "as is" without warranty of any kind, either express or implied, including, but not limited to, the implied warranties of noninfringement, merchantability, or use for a particular purpose. Informatica Corporation does not warrant that this software or documentation is error free. The information provided in this software or documentation may include technical inaccuracies or typographical errors. The information in this software and documentation is subject to change at any time without notice.

NOTICES

This Informatica product (the "Software") includes certain drivers (the "DataDirect Drivers") from DataDirect Technologies, an operating company of Progress Software Corporation ("DataDirect") which are subject to the following terms and conditions:

1. THE DATADIRECT DRIVERS ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT.
2. IN NO EVENT WILL DATADIRECT OR ITS THIRD PARTY SUPPLIERS BE LIABLE TO THE END-USER CUSTOMER FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, CONSEQUENTIAL OR OTHER DAMAGES ARISING OUT OF THE USE OF THE ODBC DRIVERS, WHETHER OR NOT INFORMED OF THE POSSIBILITIES OF DAMAGES IN ADVANCE. THESE LIMITATIONS APPLY TO ALL CAUSES OF ACTION, INCLUDING, WITHOUT LIMITATION, BREACH OF CONTRACT, BREACH OF WARRANTY, NEGLIGENCE, STRICT LIABILITY, MISREPRESENTATION AND OTHER TORTS.

Part Number: IFC-UG-97000-0001

# Table of Contents

## Chapter 6: Remote Configuration Management. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 90

## Chapter 7: Fast Clone Command Line Interface. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 94

## Appendix A: Troubleshooting. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 112

## Appendix B: Fast Clone Configuration File Parameters. . . . . . . . . . . . . . . . . . . . . . . 114

# Preface

This *Informatica Fast Clone User Guide* describes how to use the Informatica Fast Clone user interfaces to clone bulk data from an Oracle source to heterogeneous targets.

This guide is intended for users who need to configure, run, and monitor data cloning jobs, including DBAs, application developers, testing engineers, and system administrators.

This guide discusses how to perform the following tasks:

- Use the Fast Clone Console to configure, administer, monitor, and manage data unload jobs.
- Configure and use the Fast Clone Server in a distributed data-cloning topology.
- Run the Fast Clone executable from the Fast Clone Console and the command line.
- Load the unloaded data to the target database.

# Informatica Resources

## Informatica My Support Portal

As an Informatica customer, you can access the Informatica My Support Portal at http://mysupport.informatica.com.

The site contains product information, user group information, newsletters, access to the Informatica customer support case management system (ATLAS), the Informatica How-To Library, the Informatica Knowledge Base, Informatica Product Documentation, and access to the Informatica user community.

## Informatica Documentation

The Informatica Documentation team takes every effort to create accurate, usable documentation. If you have questions, comments, or ideas about this documentation, contact the Informatica Documentation team through email at infa_documentation@informatica.com. We will use your feedback to improve our documentation. Let us know if we can contact you regarding your comments.

The Documentation team updates documentation as needed. To get the latest documentation for your product, navigate to Product Documentation from http://mysupport.informatica.com.

## Informatica Product Availability Matrixes

Product Availability Matrixes (PAMs) indicate the versions of operating systems, databases, and other types of data sources and targets that a product release supports. You can access the PAMs on the Informatica My Support Portal at https://mysupport.informatica.com/community/my-support/product-availability-matrices.

## Informatica Web Site

You can access the Informatica corporate web site at http://www.informatica.com. The site contains information about Informatica, its background, upcoming events, and sales offices. You will also find product and partner information. The services area of the site includes important information about technical support, training and education, and implementation services.

## Informatica How-To Library

As an Informatica customer, you can access the Informatica How-To Library at http://mysupport.informatica.com. The How-To Library is a collection of resources to help you learn more about Informatica products and features. It includes articles and interactive demonstrations that provide solutions to common problems, compare features and behaviors, and guide you through performing specific real-world tasks.

## Informatica Knowledge Base

As an Informatica customer, you can access the Informatica Knowledge Base at http://mysupport.informatica.com. Use the Knowledge Base to search for documented solutions to known technical issues about Informatica products. You can also find answers to frequently asked questions, technical white papers, and technical tips. If you have questions, comments, or ideas about the Knowledge Base, contact the Informatica Knowledge Base team through email at KB_Feedback@informatica.com.

## Informatica Support YouTube Channel

You can access the Informatica Support YouTube channel at http://www.youtube.com/user/INFASupport. The Informatica Support YouTube channel includes videos about solutions that guide you through performing specific tasks. If you have questions, comments, or ideas about the Informatica Support YouTube channel, contact the Support YouTube team through email at supportvideos@informatica.com or send a tweet to @INFASupport.

## Informatica Marketplace

The Informatica Marketplace is a forum where developers and partners can share solutions that augment, extend, or enhance data integration implementations. By leveraging any of the hundreds of solutions available on the Marketplace, you can improve your productivity and speed up time to implementation on your projects. You can access Informatica Marketplace at http://www.informaticamarketplace.com.

## Informatica Velocity

You can access Informatica Velocity at http://mysupport.informatica.com. Developed from the real-world experience of hundreds of data management projects, Informatica Velocity represents the collective knowledge of our consultants who have worked with organizations from around the world to plan, develop, deploy, and maintain successful data management solutions. If you have questions, comments, or ideas about Informatica Velocity, contact Informatica Professional Services at ips@informatica.com.

## Informatica Global Customer Support

You can contact a Customer Support Center by telephone or through the Online Support.

Online Support requires a user name and password. You can request a user name and password at http://mysupport.informatica.com.

The telephone numbers for Informatica Global Customer Support are available from the Informatica web site at http://www.informatica.com/us/services-and-training/support-services/global-support-centers/.

CHAPTER 1

# Fast Clone Overview

This chapter includes the following topics:

## Product Overview

Fast Clone is a high-performance data cloning tool for unloading Oracle data and moving that data quickly and efficiently to other databases and warehouse appliances in a heterogeneous environment.

You can use Fast Clone in many scenarios. For example, use it to migrate data from test to production systems, clone data to another type of system or database, materialize targets for change data replication, or create a copy of your data for development or backup purposes.

Fast Clone runs on most Linux, UNIX, and Windows operating systems. It can move data from an Oracle source system to a broad range of target relational database systems and warehouse appliances on different types of Linux, UNIX, and Windows operating systems. You can also move data from an Oracle source system to a Hadoop Distributed File System (HDFS).

Fast Clone provides the following unload methods:

- Direct path unload. Reads data directly from Oracle data files, without using the Oracle Call Interface (OCI). This method is much faster than the conventional path unload method. However, you cannot use the direct path unload method to run unload jobs from a system that is remote from the Oracle source, perform SQL JOIN operations on two or more tables, or unload data from views, cluster tables, or index-organized tables (IOTs).
- Conventional path unload. Uses the OCI to retrieve metadata and data from the Oracle source. This method is slower than the direct path unload method but does not have the limitations of direct path unload.

You can configure Fast Clone to unload data in ASCII, EBCDIC, COMP-3, or Teradata binary format. Fast Clone prepares the data for input to the native load utility of the target database.

Optionally, for Greenplum, Netezza, and Terdata targets, Fast Clone provides the DataStreamer add-on component for even better performance. DataStreamer sends the unloaded data directly to Teradata Parallel Transporter (TPT), Greenplum Gpfdist, or Netezza external tables. As a result, DataStreamer transfers data faster and reduces I/O and storage requirements.

Fast Clone includes the following additional features:

- Filters the Oracle tables and table partitions to be unloaded.
- Lets you define SQL queries, such as table joins for unloading data.
- Generates scripts for creating the target tables and for loading data.
- Provides a built-in compression feature that can quickly compress data on the fly.
- Supports Oracle Automatic Storage Management (ASM), Oracle Cluster File Systems (OCFS), chained rows, fixed-length rows, and most datatypes including LOB datatypes.
- Supports different output formats including fixed length record format, EBCDIC, COMP-3, and Teradata binary format.
- Provides reporting and monitoring capabilities and can send alerts, such as SNMP traps.
- Provides integration with the Informatica Data Replication product to prepare targets for change data replication.

# Fast Clone Usage Scenarios

You can use Fast Clone in many scenarios.

For example, use Fast Clone to perform the following tasks:

- Migrate data from an Oracle production system to another platform, without increasing overhead on the production system.
- Clone Oracle production data to create testing and development environments.
- Quickly create a backup of Oracle production data.
- Export data from Oracle tables in portable text format.
- Perform initial materialization of targets for Informatica Data Replication change data replication jobs.
- Stream data to Greenplum and Teradata data warehouse targets.
- Clone Oracle data to HDFS.

# Fast Clone Targets

Fast Clone unloads bulk data from Oracle sources. Fast Clone loads data to many types of relational databases and data warehouse appliances.

Fast Clone supports the following target types:

- Cloudera
- Flat files
- Greenplum
- Hive

- Hortonworks
- MapR
- Microsoft SQL Server
- Netezza
- Oracle
- Teradata
- Vertica

# Fast Clone Components

Informatica Fast Clone is composed of multiple components, which can run as separate executables on different systems.

Fast Clone includes the following components:

- Fast Clone Console
- Fast Clone Server
- Fast Clone executable
- DataStreamer

## Fast Clone Console

The Fast Clone Console is the graphical user interface from which you configure and manage data-cloning jobs. From the Fast Clone Console, you can generate a cloning configuration file and run data-cloning jobs on a local or remote system. The cloning configuration file name is unload.ini by default. The Fast Clone Console runs on Linux, UNIX, and Windows. You can run it on the Oracle source system or on a standalone system. To start the Fast Clone Console, run gui.cmd on Windows or gui.sh on Linux or UNIX.

## Fast Clone Executable

The Fast Clone executable, FastReader.exe on Windows or FastReader on Linux or UNIX, runs data-cloning jobs. You can start the Fast Clone executable from the command line or from the Fast Clone Console. If you run the Fast Clone executable from the command line, you can manually enter parameters to control unload processing.

For more information, see Fast Clone Command Line Interface.

## Fast Clone Server

The Fast Clone Server is an optional add-on component that you can purchase to enable network communication across systems in a distributed Fast Clone topology. These systems can include the Fast Clone Console and Fast Clone instances on source and target database systems.

The Fast Clone Server runs as a Windows service or as a daemon on Linux or UNIX.

Use the Fast Clone Server to initiate unloads of Oracle data and metadata on a remote system or to transmit the output files to the target system. For example, use the Fast Clone Server to retrieve Oracle data that was unloaded by scheduled Fast Clone unload operations on separate Oracle systems and then make that data available to another Fast Clone instance for loading to the target database.

## DataStreamer

DataStreamer is an optional add-on component that you can purchase for faster loads of data to Greenplum, Netezza, and Teradata targets.

With DataStreamer, you must use the direct path unload method. Depending on the target type, DataStreamer streams the unloaded Oracle data to the target in one of the following ways:

- For Greenplum targets, DataStreamer sends the unloaded data directly to the Greenplum parallel file distribution server (gpfdist) for loading to the target.
- For Netezza targets, DataStreamer writes the unloaded data to the named pipes that represent the Netezza external tables. The Netezza ODBC driver reads data from these pipes and loads data to the Netezza target tables.
  To use Netezza DataStreamer, you must install the Netezza ODBC driver on the system where you plan to run unload jobs.
- For Teradata targets, DataStreamer sends the unloaded data directly to the Teradata Parallel Data Pump, FastLoad, or MultiLoad utility for loading to the target.
  To use Teradata DataStreamer, you must install the TPT libraries on the system where you plan to run unload jobs.

# Data Unload Methods

Fast Clone supports conventional path and direct path unload methods.

The direct path unload method reads data directly from Oracle data files and requires the Oracle user to have special user permissions, such as SELECT FROM DICTIONARY. This method provides high-performance unloads of data without increasing overhead on the source system. With this method, you can use the DataStreamer component to additionally enhance performance. By default, Fast Clone does not lock the source tables to unload data. However, you can change this behavior to perform a consistent read of the source data.

**Note:** Fast Clone cannot read data from Oracle data files that are on a standalone system. Fast Clone must connect to a running Oracle instance to retrieve metadata for unload processing.

The conventional path unload method uses the OCI to retrieve metadata and data from the Oracle source. This method usually provides an acceptable and efficient level of performance but is slower than the direct path unload method. With the conventional path unload method, you can run Fast Clone on a system that is remote from the source database server without any special permissions.

The following table describes the differences between the direct path unload and conventional path unload methods:

| Feature | Direct Path Unload | Conventional Path Unload |
|---|---|---|
| Reads data directly from the Oracle source files | Yes | No |
| Performs high performance unloads | Yes. Provides very high-performance unloads, up to 20 times faster than the native database exports. | Yes. When Fast Clone unloads data locally, performance is good but CPU consumption can be twice that of the direct path unload method. When Fast Clone unloads data remotely, performance might be slower than that of unload on a local system. |
| Supports remote execution of Fast Clone | Yes, only if the Fast Clone Server is used. | Yes |
| Supports SQL queries that perform joins between tables | No | Yes |
| Works with DataStreamer | Yes | No |
| Unloads data from compressed tables, including tables that use Oracle Advanced Compression | Yes | Yes |
| Unloads data from tables that use Oracle Exadata Hybrid Columnar Compression (EHCC) | No | Yes |
| Unloads data from views, cluster tables, and index-organized tables | No | Yes |
| Performs parallel processing of data unloads using multiple threads | Yes | Yes. For partitioned tables, Fast Clone uses one thread per table partition or subpartition. For tables that are not partitioned, Fast Clone distributes work across threads based on the Oracle ROWID value, provided that unload processing is local. |
| Unloads data from Oracle ASM instances | Yes | Yes |
| Integrates with Informatica Data Replication | Yes | Yes |

# Supported Datatypes for Oracle Sources

Fast Clone supports most Oracle datatypes for bulk data movement.

The following table identifies the datatypes that Fast Clone supports and does not support for Oracle source databases:

| Datatype | Supported? | Comments |
|---|---|---|
| ANY types | No | - |
| BINARY_DOUBLE | Yes | - |
| BINARY_FLOAT | Yes | - |
| BFILE | No | - |
| BLOB | Yes | SECUREFILE is supported with the conventional path unload method |
| CHAR | Yes | - |
| CLOB | Yes | SECUREFILE is supported with the conventional path unload method |
| DATE | Yes | - |
| FLOAT | Yes | - |
| Expression Filter Type | No | - |
| INTERVAL DAY TO SECOND | No | - |
| INTERVAL YEAR TO MONTH | No | - |
| LONG | Yes | - |
| LONG RAW | Yes | - |
| Media types | No | - |
| MLSLABEL | No | - |
| NCHAR | Yes | - |
| NCLOB | Yes | SECUREFILE is supported with the conventional path unload method |
| NUMBER | Yes | Maximum 128 characters |
| NVARCHAR2 | Yes | The extended version of this datatype, which was introduced in Oracle 12c, is not supported. An extended NVARCHAR2 datatype can have a maximum size of 32767 bytes. |

| Datatype | Supported? | Comments |
|---|---|---|
| RAW | Yes | The extended version of this datatype, which was introduced in Oracle 12c, is not supported. An extended RAW datatype can have a maximum size of 32767 bytes. |
| REF | No | - |
| ROWID | Yes | Supported only with the conventional path unload method |
| SDO_GEOMETRY | Yes | Supported only for Oracle targets |
| TIMESTAMP | Yes | - |
| TIMESTAMP WITH LOCAL TIMEZONE | Yes | - |
| TIMESTAMP WITH TIMEZONE | Yes | - |
| URI types | No | - |
| UROWID | Yes | Supported only with the conventional path unload method |
| User-defined types | No | - |
| VARCHAR | Yes | - |
| VARCHAR2 | Yes | The extended version of this datatype, which was introduced in Oracle 12c, is not supported. An extended VARCHAR2 datatype can have a maximum size of 32767 bytes. |
| XML types | Yes | The Fast Clone Console displays these rows as CLOB |

# Fast Clone Deployment Topologies

You can use Fast Clone in different topologies to clone data. You can deploy Fast Clone components on a single system and run data unload jobs locally or on different systems in a distributed environment.

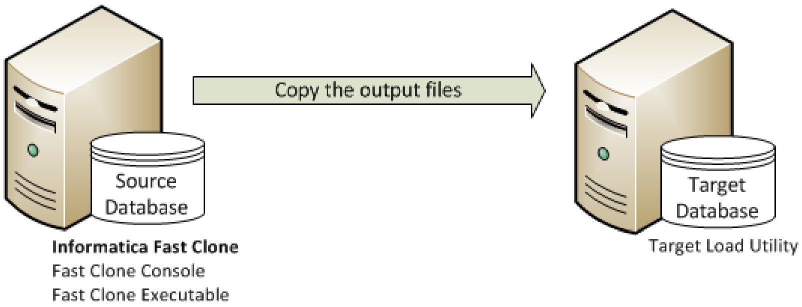
In a distributed deployment topology, run multiple Fast Clone Server instances to initiate unload requests on a remote system or to transmit the output files to the target system.

Review the sample network topologies to learn where to install Fast Clone components under alternative scenarios.

## Running Fast Clone on the Oracle Database Server

Typically, you run Fast Clone on the system where the Oracle source database runs. In this topology, you can use the direct path unload method for faster bulk data movement.

The following image shows a typical topology in which Fast Clone runs on the source database server:



Fast Clone components, such as the Fast Clone Console and Fast Clone executable, run on the source database system. From the Fast Clone Console, you can create cloning configuration files and run data unload jobs. Fast Clone unloads data to output files on the source system.

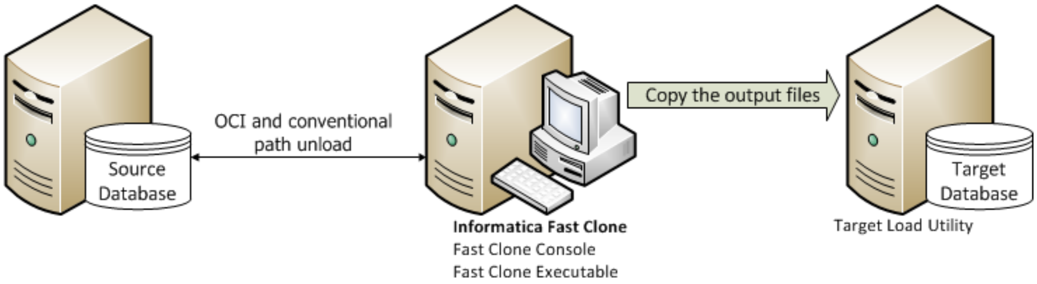
After you copy the output files to the target system, you can run the load script on the target system to load data. Fast Clone uses the target database load utility to load data.

Alternatively, you can install the target database load utility on the source system and run the load script on the source system. If you load data from pipes, install the target load utility on the source system.

## Running Fast Clone on a Standalone Computer

If you cannot install Fast Clone on the Oracle source system because of system or security restrictions, you can deploy Fast Clone on a standalone system. In this topology, Fast Clone uses the conventional path unload method to unload Oracle data from the remote source database.

The following image shows a deployment topology in which Fast Clone runs on a standalone system:



Fast Clone components, such as the Fast Clone Console and Fast Clone executable, run on the standalone system. From the Fast Clone Console, you can create cloning configuration files and run data unload jobs locally on the Fast Clone system. Fast Clone uses the Oracle Call Interface (OCI) to read the Oracle source data from the remote source system. Fast Clone unloads data to output files.

After you copy the output files to the target system, you can run the load script on the target system to load data. Fast Clone uses the target database load utility to load data.

Alternatively, you can install the target database load utility on the system where you run the Fast Clone executable and then run the load script on this system. If you load data from pipes, install the target load utility on the system where you run the Fast Clone executable.

# Running Multiple Fast Clone Instances with the Fast Clone Server

You can deploy multiple Fast Clone instances on different systems in a distributed environment. In this topology, you must use the Fast Clone Server to communicate with remote Fast Clone instances.

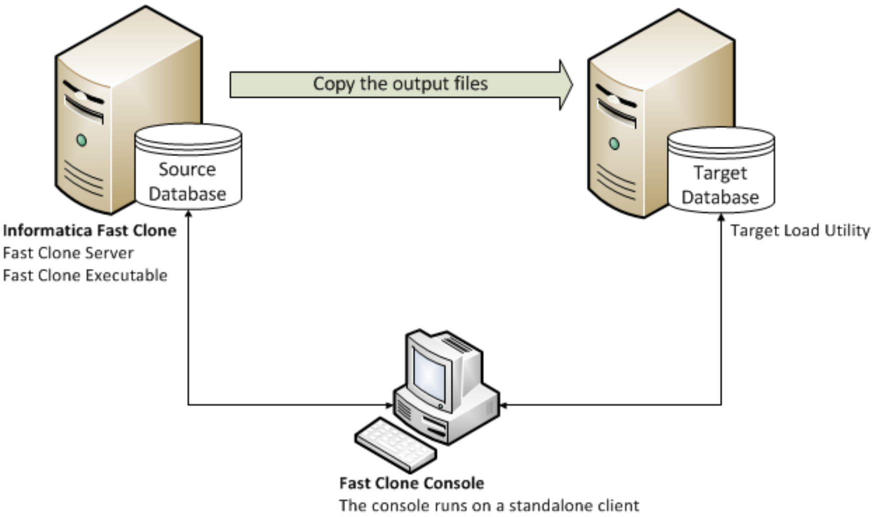
You can run the Fast Clone Server on the source database system, on the target database system, or on both the source and target systems.

When you run the Fast Clone Server on the source system, the Fast Clone Server accepts unload requests from the Fast Clone Console and runs data unload jobs. Run the Fast Clone Server on the source system if you installed the Fast Clone Console on a standalone system and want to run data unload jobs that use the direct path unload method from the Fast Clone Console.

When you run the Fast Clone Server on the target system, the Fast Clone Server accepts the output files from a remote Fast Clone instance. You do not need to copy the output files manually to the target. Run the Fast Clone Server on the target system if you run scheduled tasks for cloning data.

## Running a Fast Clone Server on the Source System

The following image shows a deployment topology in which the Fast Clone Server runs on the source database system:



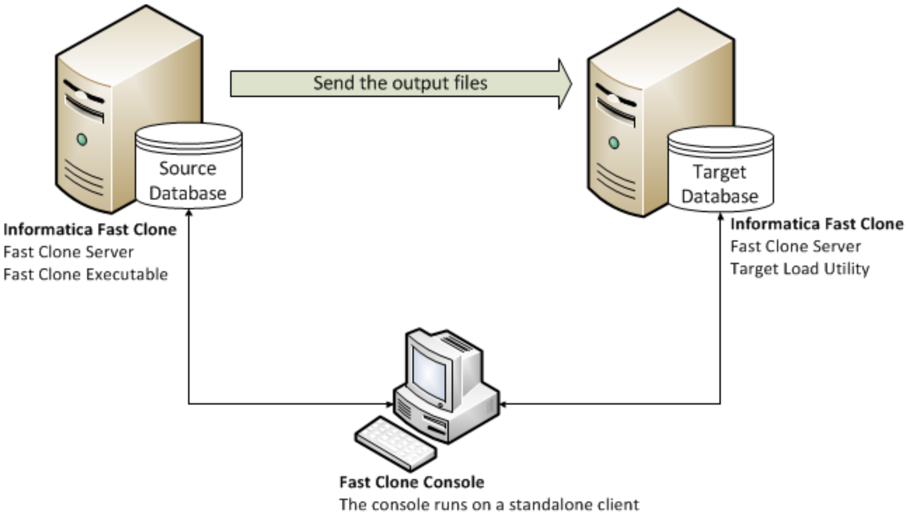
In this topology, the Fast Clone Server and Fast Clone executable run on the source system. The Fast Clone Console runs on a standalone system. From the Fast Clone Console, you can create cloning configuration files and initiate data unload jobs that run under the control of the Fast Clone Server. You can use either the direct path unload method or conventional path unload method. Fast Clone produces output files on the source system.

After you copy the output files to the target system, you can run the load script on the target system to load data. Fast Clone uses the target database load utility to load data. Alternatively, you can install the target database load utility on the source system and run the load script on the source system. Install the target load utility on the source system if you load data from pipes.

## Running a Fast Clone Server on Both the Source and Target Systems

You can also run another Fast Clone Server instance on the target system to accept output files sent from the Fast Clone instance on the source system.

The following image shows a deployment topology that uses two Fast Clone Server instances for distributed data movement:



In this topology, you do not have to manually copy the output files to the target.

# Integration with Informatica Data Replication

If you have a license for Informatica Data Replication, you can use Fast Clone to perform an initial load of Data Replication targets.

Data Replication replicates transactional data between heterogeneous databases and platforms while maintaining transactional integrity and data consistency. Data Replication performs low-latency batched replication or continuous replication of change data and metadata.

Fast Clone can use a Data Replication configuration file to perform a high-speed initial load of Oracle data to the target systems. Fast Clone supports all of the targets that Informatica Data Replication supports.

CHAPTER 2

# Configuring and Using the Fast Clone Server

This chapter includes the following topics:

## Fast Clone Server Overview

The Fast Clone Server enables network communication across systems in a distributed Fast Clone topology. These systems can include the Fast Clone Console and Fast Clone instances on source and target database systems. Use the Fast Clone Server to initiate unloads of Oracle data and metadata or to transmit the output files to the target database system.

The Fast Clone Server configuration file contains parameters that control how the Fast Clone Server runs. The Fast Clone Server reads this file when you start the server. If you edit parameters in this file, restart the Fast Clone Server for the changes to take effect.

When you start the Fast Clone Server from the command line, enter the SERVER_CONFIG parameter to specify the server configuration file.

## Fast Clone Server Configuration File

Fast Clone provides a predefined server configuration file, server.ini, in the top-level Fast Clone installation directory. You can edit this file to override the default Fast Clone Server settings.

The Fast Clone Server configuration file contains the following parameters to control how the Fast Clone Server runs.

## Parameter Descriptions

**allow_local_output_directories=*list_of_directories***

Identifies the directories on the Fast Clone Server system to which Fast Clone can write output files. You can include the question mark (?) and asterisk (*) wildcards to define a mask that matches multiple directories. When you define the cloning configuration file, unload.ini, ensure that the path and directory name that you specify in the **Output directory** field matches one of directories in this list.

Default value: Asterisk (*)

**allow_remote_extraction_hosts=*list_of_hosts***

Identifies the hosts from which the Fast Clone Server can receive the cloning configuration files for remote unload processing. Enter the IP addresses or host names, separated by a semicolon. You can include the question mark (?) and asterisk (*) wildcards to define a mask for the hosts. Enter only the asterisk (*) wildcard to configure the Fast Clone Server to receive unload configuration files from any remote host.

For example:

```
allow_remote_extraction_hosts=84.109.19.100; 84.109.19.101; 84.109.19.102
```

Default value: Asterisk (*)

**allow_remote_extraction_requests={true|false}**

Indicates whether the Fast Clone Server can receive unload requests from remote systems that have a Fast Clone instance. Valid values are:

- **true**. The Fast Clone Server can receive unload requests from a remote Fast Clone instance and run the unload jobs. For example, if you run the Fast Clone Console on a standalone system, the Fast Clone Server accepts unload requests and associated configuration files from the Fast Clone Console and runs the unload jobs.
- **false**. The Fast Clone Server does not accept unload requests from a remote Fast Clone instance. Enter this value for security purposes. This value ensures that the Fast Clone Server does not accept unload requests from a remote system that is not your data-cloning topology.

Default value: true

**allow_recieve_remote_files={true|false}**

Indicates whether the Fast Clone Server can receive output files from a remote Fast Clone system. Valid values are:

- **true**. The Fast Clone Server can receive output data files from another system for load processing. Use this option for a Fast Clone Server that you run on the target database system to receive output files from the source database system.
- **false**. If you run data load jobs on the system where Fast Clone produces the output files, or if you manually transmit the output files to the target system, enter this value for security purposes. This value ensures that the Fast Clone Server does not accept output files from a remote system that is not in your data-cloning topology.

You can specify the **allow_recieve_remote_files_hosts** parameter to identify the host systems from which the Fast Clone Server can receive output files.

Default value: false

**allow_recieve_remote_files_hosts=*list_of_hosts***

Identifies the hosts from which the Fast Clone Server can receive output files. Enter the IP addresses or host names, separated by a semicolon. You can include the question mark (?) and asterisk (*) wildcards to define a mask. Enter only the asterisk (*) wildcard to enable the Fast Clone Server to receive output files from any remote host.

For example:

```
allow_recieve_remote_files_hosts=84.109.19.100; 84.109.19.101; 84.109.19.102
```

Default value: Asterisk (*)

**bind_ip=*list_of_IP_addresses***

Specifies the IP addresses of the remote Fast Clone instances for which the Fast Clone Server listens for incoming requests. Enter 0.0.0.0 for all IPv4 addresses or [::] for all IPv6 addresses.

Default value: 0.0.0.0

**listener_port=*port_number***

Specifies the port on which the Fast Clone Server listens for incoming requests. Valid values are from 1 through 65535.

Default value: 5521

**log={true|false}**

Indicates whether the Fast Clone Server writes informational and error messages to a log file. Options are:

- **true**. The Fast Clone Server creates a log file in the directory that you specify in the logdir parameter.
- **false**. The Fast Clone Server does not use a log file.

Default value: false

**logdir=*directory_path***

If you set the log parameter to true, specifies the directory where the Fast Clone Server creates the message log file.

Default value:

- C:\\temp on Windows
- /tmp on Linux and UNIX

**log_level=*n***

If you set the log parameter to true, specifies the verbose level for the Fast Clone Server logged output. Larger values provide more detailed output. Valid values are from 0 through 4.

Default value: 0

**max_connections=*n***

Specifies the maximum number of Fast Clone client connections to the Fast Clone Server. Valid values are from 1 through 3000.

Default value: 150

# Installing and Uninstalling the Fast Clone Server on Windows

On Windows, you must install the Fast Clone Server service before you can start it. Later, you can uninstall the Fast Clone Server service to upgrade Fast Clone or deploy the Fast Clone instance to another system.

- To install the Fast Clone Server service, run the install_server.cmd script.
  Alternatively, run the following commands from the command line to install the Fast Clone Server service:

  ```
  cd Fast_Clone_installation_directory
  FastReader.exe RUN_AS_SERVICE=true SERVER_CONFIG=server.ini -i
  ```

- To uninstall the Fast Clone Server service, stop the Fast Clone Server and then run the uninstall_server.cmd script.
  Alternatively, run the following commands from the command line to uninstall the Fast Clone Server service:

  ```
  cd Fast_Clone_installation_directory
  FastReader.exe RUN_AS_SERVICE=true SERVER_CONFIG=server.ini -r
  ```

# Starting the Fast Clone Server

Typically, you run Fast Clone Server instances on the source system and on the target system. A Fast Clone Server runs as a service on Windows or as a daemon on Linux or UNIX.

To start the server, perform one of the following actions:

- On Windows, after you install the Fast Clone Server service, run the start_server.cmd script to start the Fast Clone Server service.
  Alternatively, run the following commands from the command line to start the Fast Clone Server service:

  ```
  cd Fast_Clone_installation_directory
  FastReader.exe RUN_AS_SERVICE=true SERVER_CONFIG=server.ini -s
  ```

- On Linux or UNIX, run the server.sh script to start the Fast Clone Server daemon.

# Stopping the Fast Clone Server

You might need to stop the Fast Clone Server service or daemon to upgrade Fast Clone or perform other administrative tasks.

- On Windows, run the stop_server.cmd script to stop the Fast Clone Server.
  Alternatively, run the following commands from the command line to stop the Fast Clone Server:

  ```
  cd Fast_Clone_installation_directory
  FastReader.exe RUN_AS_SERVICE=true SERVER_CONFIG=server.ini -k
  ```

  After you stop the Fast Clone Server service, you can uninstall it, if necessary.

- On Linux or UNIX, run the following command from the command line to stop the Fast Clone Server:

  ```
  kill -9 fast_clone_server_process_id
  ```

CHAPTER 3

# Creating Cloning Configuration Files in the Fast Clone Console

This chapter includes the following topics:

## Configuration File Overview

In the Fast Clone Console, you can create and manage cloning configuration files. Fast Clone uses the configuration files to unload source data, generate load scripts, enable integration with Informatica Data Replication, and configure reporting and SNMP traps for monitoring.

A configuration file is a text file. Informatica recommends that you use the Fast Clone Console to create and manage configuration files to reduce the risk of error. Alternatively, you can create and edit configuration files in a text editor.

By default, the Fast Clone Console saves a configuration file as unload.ini to the *FastClone_installation* directory in a local file system. You can optionally save the configuration file under another name.

In the Fast Clone Console, you can open a configuration file to edit unload settings or use it to run a data unload job. If you open the configuration file to run data unload job, the Fast Clone Console overrides the unload.ini file in the *FastClone_installation* directory if it already exists.

If you use the Fast Clone Server to run data unload jobs on a remote system, you can enable remote configuration management in the Fast Clone Console. You can then open and save the configuration files on the remote system from the Fast Clone Console using an FTP server.

A configuration file is not required for data cloning if you run Fast Clone from the command line. However, you must then manually specify all of the parameters at the command line for the Fast Clone executable.

### Related Topics:

- "Fast Clone Command Line Interface" on page 94
- "Fast Clone Configuration File Parameters" on page 114

# Starting the Fast Clone Console

You can run the Fast Clone Console on the source or target system or on a standalone system.

Before you start the Fast Clone Console the first time, verify that the following system requirements are met:

- The Java Runtime Environment (JRE) 1.5 or later is installed and the JRE_HOME environment variable points to the JRE base directory.

  **Note:** For Hadoop and Hive targets, the Fast Clone Console requires JRE 1.7 x64 or later.
- The JRE_HOME directory is in the PATH environment variable.
- On Linux or UNIX, an X Window environment is configured.

For more information about prerequisites for the Fast Clone Console, see the *Informatica Fast Clone Installation Guide*.

To start the Fast Clone Console, run one of the following scripts that are located in the *FastClone_installation* directory:

- On Windows, run gui.cmd.
- On Linux or UNIX, run gui.sh.

## Toolbar Buttons

Instead of choosing commands from the menu, you can click buttons on the toolbar to quickly initiate common actions.

The following buttons appear on the toolbar:

| Button | Description |
|---|---|
| 🏃 | Exits the Fast Clone Console. |
| 📂 | Opens an existing configuration file. |
| 🌐 | Opens an existing configuration file from a remote system. This button appears only if remote configuration management is enabled. |

| Button | Description |
|---|---|
| 💾 | Generates a configuration file based on your entries and saves it to a local file system, or saves an updated configuration file. |
| 📤 | Transmits a configuration file to a remote system. This button appears only if remote configuration management is enabled. |
| ↻ | Refreshes schemas for the source and target databases. |
| ▶ | Starts a Fast Clone data unload job for the open configuration file. |
| 🟢 | Re-enables or creates target constraints that were previously disabled or dropped. |
| 🔴 | Disables or drops target constraints to avoid constraint violations during data loading. |
| ✓ | Validates the target schema. |
| XML | Generates the data files in XML format. |
| ▦ | Generates SQL CREATE TABLE scripts to create Oracle target tables based on the source table schema. |
| SQL | Manages user-defined SQL queries for data unload jobs. |

# Task Flow: Creating a Configuration File in the Fast Clone Console

Perform the following tasks to create a configuration file for a data-cloning job in the Fast Clone Console:

1. Define the source database. See “Defining a Source” on page 28.
2. Select the source tables from which to unload data. See “Selecting Source Tables and Materialized Views for Unload Processing” on page 32.
3. Optional. Define SQL queries, such as table joins for unloading data. See “Defining SQL Queries for Unload Processing” on page 34.
4. Optional. If you unload data from partitioned tables, exclude selected partitions from the data unload job. See “Customizing Unload Processing for Table Partitions” on page 35.
5. Optional. Exclude selected columns from the data unload job. See “Customizing Unload Processing for Table Columns” on page 36.
6. Define the target database. See “Defining a Target” on page 39.
7. Configure runtime settings. See “Configuring Runtime Settings” on page 45.

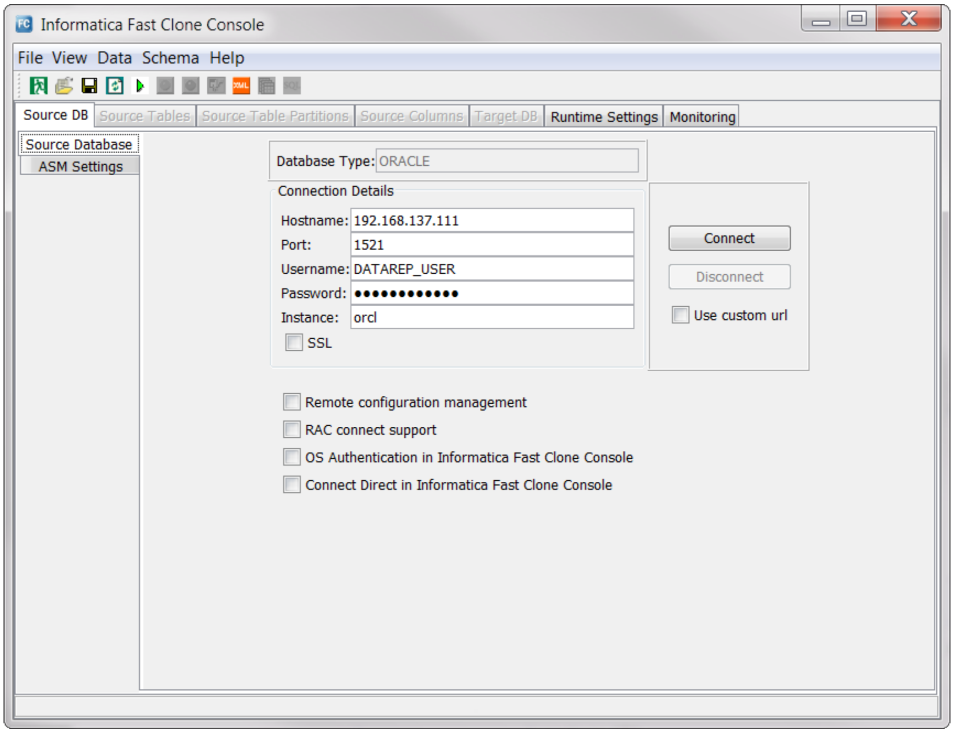
8. Optional. Configure reporting and SNMP monitoring and traps. See “Configuring Reporting and SNMP Traps” on page 69.
9. Save the configuration file. See “Saving a Cloning Configuration File Locally” on page 71.

# Defining a Source

For Fast Clone to connect to the Oracle source database to unload data, you must enter the source database connection information. The Fast Clone Console saves this information to the configuration file. When you run an unload job, Fast Clone uses this information to generate a connection string for Oracle Net Services.

1. Click the **Source DB** tab > **Source Database** view.

   The following image shows this view:



Database Type: ORACLE

Hostname: 192.168.137.111
Port: 1521
Username: DATAREP_USER
Password: ●●●●●●●●●●●●
Instance: orcl
□ SSL

□ Use custom url

□ Remote configuration management
□ RAC connect support
□ OS Authentication in Informatica Fast Clone Console
□ Connect Direct in Informatica Fast Clone Console

   If you successfully connected to the Oracle source previously, Fast Clone displays the saved connection information.

2. Enter the connection information as needed.

The following table describes the connection fields:

| Field | Description |
|---|---|
| Hostname | The host name or IP address of the system where the source database runs. The default value is the local host name. |
| Port | The port number that Fast Clone uses to connect to the source database. The default port number is 1521. |
| Username | A user name that has the privileges required to unload data from the source database. For more information about required user privileges, see *Informatica Fast Clone Installation Guide*. |
| Password | A valid password for the specified user. |
| Instance | The Oracle instance name. If you leave this field blank, Fast Clone uses the ORACLE_SID or TWO_TASK environment variable to determine the Oracle instance name. |
| SSL | Select this option to configure Fast Clone to use the TCP/IP protocol with the Secure Sockets Layer (SSL), also called the TCPS protocol, to connect to the Oracle source.<br>**Important:** If you use TCPS connections between the Fast Clone Console and remote Oracle databases, you must install the 64-bit Oracle Instant Client on the Fast Clone Console system. |
| Remote Configuration Management | Select this option to enable remote configuration management. For more information, see Chapter 6, “Remote Configuration Management” on page 90. |
| RAC connect support | Select this option if you unload data from an Oracle database in a RAC environment. To connect to the Oracle RAC, the Fast Clone Console uses the instance name that is specified in the **Instance** field and the Fast Clone executable uses a service name that is specified in the **Service** field. |
| OS Authentication in Fast Clone | Select this option to configure Fast Clone to use operating system authentication to connect to the Oracle database. |
| Connect Direct in Informatica Fast Clone Console | Select this option to configure the Fast Clone executable to use the Bequeth protocol to connect to the Oracle source. |
| Use custom url | Select this option to configure the Fast Clone executable and Fast Clone Console to use different connection settings. In this case, Fast Clone uses the following connection settings to connect to the source database:<br>- The Fast Clone executable uses the connection information that you provide under **Connection Details** to generate a connection string.<br>- The Fast Clone Console uses the connection string that you enter in the **JDBC custom URL** field. |

| Field | Description |
|---|---|
| JDBC custom URL | If you select the **Use custom url** option, specifies a connection URL for the JDBC driver that the Fast Clone Console uses to connect to an Oracle source. This URL has the following format:<br>`jdbc:oracle:thin@[host][:port]:SID`<br>If the Console uses an SSL connection to an Oracle source, the URL has the following format:<br>`jdbc:oracle:oci@[host][:port]:SID`<br>If you enter an SSL connection URL, clear the **SSL** option.<br>**Important:** JDBC custom connection URLs always use the values that are specified in the **Username** and **Password** fields. Ensure that any JDBC custom connection URL that you specify does not include a user name and a password. |
| Service | For an Oracle database in a RAC environment, specifies the Oracle service name that the Fast Clone executable uses to connect to the source database. |

3. Click **Connect** to connect to the source.

   If the connection is successful, the **Source Tables** tab opens. If the connection is not successful, the Fast Clone Console reports an error.

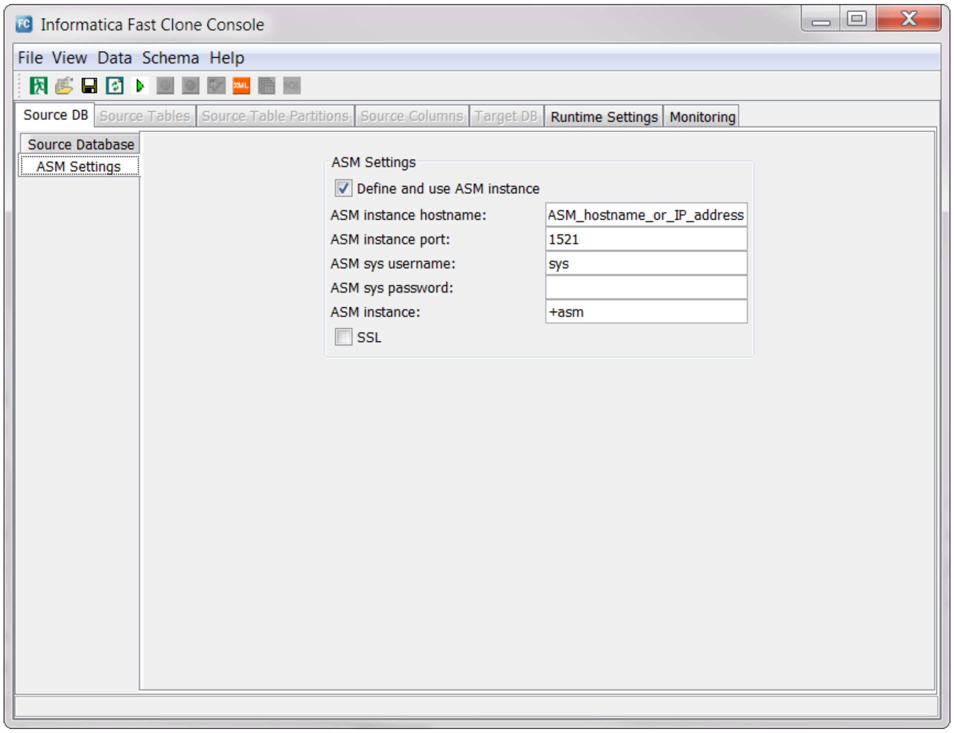
   **Tip:** To edit connection information, click **Disconnect**, edit connection information, and then click **Connect** again.

## Configuring a Connection to an Oracle ASM Instance

If you use Oracle Automatic Storage Management (ASM) to manage storage for Oracle source data, you must enter connection settings for the ASM instance.

1. Click the **Source DB** tab > **ASM Settings** view.

   The following image shows this view:



ASM Settings

☑ Define and use ASM instance

ASM instance hostname: ASM_hostname_or_IP_address

ASM instance port: 1521

ASM sys username: sys

ASM sys password:

ASM instance: +asm

□ SSL

2. Select the **Define and use ASM instance** option.

   All of the fields on the **ASM Settings** view become available.

3. Enter connection information for the ASM instance.

   **Note:** If your Oracle source is in a RAC with multiple ASM instances, enter connection information for any one of the running Oracle ASM instances.

   The following table describes the fields on the **ASM Settings** view:

| Field | Description |
|---|---|
| ASM instance hostname | The host name or IP address of the system with the ASM instance. The default value is the local host name. |
| ASM instance port | The port number that is used to connect to the ASM instance. The default value is 1521. |
| ASM sys username | The user name of the default Oracle ASM SYS user that was created at Oracle installation, or another user that you define. This user must have SYSDBA privileges. |
| ASM sys password | A password for the ASM SYS user. |

| Field | Description |
|---|---|
| ASM instance | A service name for the ASM instance. The default value is +ASM. |
| SSL | Select this option to configure Fast Clone to use the TCP/IP protocol with the Secure Sockets Layer (SSL), also called the TCPS protocol, to connect to the ASM instance.<br>**Important:** If you use TCPS connections between the Fast Clone Console and remote Oracle databases, you must install the Oracle Instant Client on the Fast Clone Console system. |

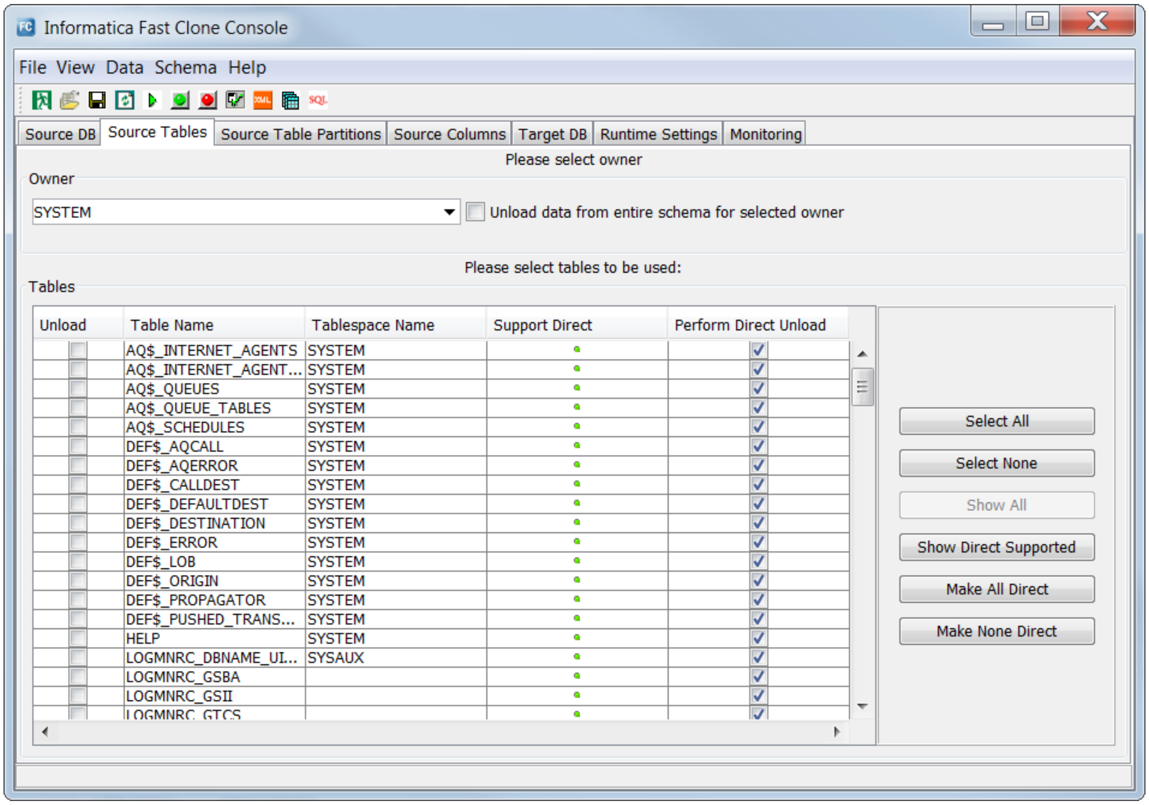
**Note:** If Fast Clone reports an ORA-12528 error when connecting to an ASM instance, add the ASM instance to the Oracle listener.ora configuration file and restart the Oracle listener process.

# Selecting Source Tables and Materialized Views for Unload Processing

On the **Source Tables** tab, select one or more source tables or materialized views from which to unload data and specify the unload method for these tables or materialized views.

1. Click the **Source Tables** tab.

   The following image shows this tab:



Please select owner

Owner: SYSTEM

□ Unload data from entire schema for selected owner

Please select tables to be used:

Tables

| Unload | Table Name | Tablespace Name | Support Direct | Perform Direct Unload |
|---|---|---|---|---|
| □ | AQ$_INTERNET_AGENTS | SYSTEM | ● | ☑ |
| □ | AQ$_INTERNET_AGENT... | SYSTEM | ● | ☑ |
| □ | AQ$_QUEUES | SYSTEM | ● | ☑ |
| □ | AQ$_QUEUE_TABLES | SYSTEM | ● | ☑ |
| □ | AQ$_SCHEDULES | SYSTEM | ● | ☑ |
| □ | DEF$_AQCALL | SYSTEM | ● | ☑ |
| □ | DEF$_AQERROR | SYSTEM | ● | ☑ |
| □ | DEF$_CALLDEST | SYSTEM | ● | ☑ |
| □ | DEF$_DEFAULTDEST | SYSTEM | ● | ☑ |
| □ | DEF$_DESTINATION | SYSTEM | ● | ☑ |
| □ | DEF$_ERROR | SYSTEM | ● | ☑ |
| □ | DEF$_LOB | SYSTEM | ● | ☑ |
| □ | DEF$_ORIGIN | SYSTEM | ● | ☑ |
| □ | DEF$_PROPAGATOR | SYSTEM | ● | ☑ |
| □ | DEF$_PUSHED_TRANS... | SYSTEM | ● | ☑ |
| □ | HELP | SYSTEM | ● | ☑ |
| □ | LOGMNRC_DBNAME_UI... | SYSAUX | ● | ☑ |
| □ | LOGMNRC_GSBA | | ● | ☑ |
| □ | LOGMNRC_GSII | | ● | ☑ |
| □ | LOGMNRC_GTCS | | ● | ☑ |

2. Under **Owner**, select a schema owner name in one of the following ways:

- Select the schema owner from the list.
- Enter the first few letters of the schema owner name. If a matching owner name exists in the source database, the Fast Clone Console completes the name.

**Note:** The owner name is the same as the schema name.

The **Tables** box lists all of the tables and materialized views with the specified schema owner name. The **Support Direct** column indicates whether the table or materialized view supports the direct path unload method.

**Tip:** To view only the tables and materialized views that support the direct path unload method, click **Show Direct Supported**. To view all of the tables and materialized views in the schema again, click **Show All**.

3. To select a subset of source tables or materialized views for unload processing and set the unload method for each one, complete the following steps:

   a. Select the **Unload** check box for each table or materialized view, or click **Select All** and then clear the **Unload** check box for any table or materialized view that you do not want to unload data from.

      **Note:** The **Select None** button clears all selections to exclude all of the tables and materialized views from unload processing.

   b. To use the direct path unload method for a selected table or materialized view, ensure that the **Perform Direct Unload** check box is selected for the table or materialized view.

      By default, the Fast Clone Console selects the direct path unload method for all of the source tables and materialized views for which Fast Clone supports the direct path unload method.

   c. To use the conventional path unload method for a selected table or materialized view, clear the **Perform Direct Unload** check box for the table or materialized view.

      To clear the **Perform Direct Unload** check box for all of the tables and materialized views, click **Make None Direct**. If you configure Fast Clone to unload data from a remote source database server, click **Make None Direct** to use the conventional path method.

4. To select all of the tables and materialized views for the specified schema owner for unload processing, complete the following steps:

   a. Select the **Unload data from entire schema for selected owner** option.

   b. In the **Unload type** list, select an option for the unload method to use for the tables and materialized views in the schema. Options are:

      - **Let Informatica Fast Clone decide during unload**. Fast Clone uses the direct path unload method for the tables and materialized views that support this method. Fast Clone uses the conventional path unload method for the tables and materialized views that do not support the direct path unload method.
      - **Perform direct unload for all tables**. Fast Clone uses the direct path unload method for all of the tables and materialized views. Do not use this option if the source schema includes one or more tables or materialized views that do not support the direct path unload method.
      - **Perform conventional unload for all tables (none direct)**. Fast Clone uses the conventional path unload method for all of the tables and materialized views. Use this option if you need to unload data from a remote source database server.

# Defining SQL Queries for Unload Processing

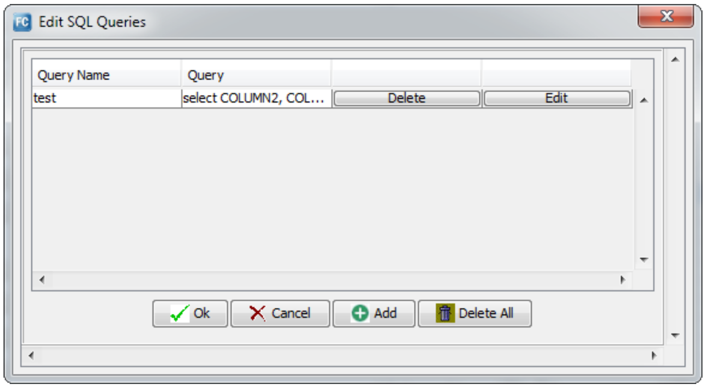
You can define SQL query statements that are processed when the data unload jobs run. For example, create a query that performs table joins or selects data from a view.

Fast Clone uses the conventional path unload method for unload jobs that use configuration files with user-defined SQL queries. During unload processing, Fast Clone generates an output file that contains the query results for each query that you define. Fast Clone uses the query name that you specify in the configuration file as the output file name.

1. Click **Data** > **Add Conventional SQL** on the menu bar, or click the **Add conventional path SQL queries** button on the toolbar.

   The menu command and toolbar button are not available until you connect to the source database.

   The **Edit SQL Queries** dialog box appears, as shown in the following image:

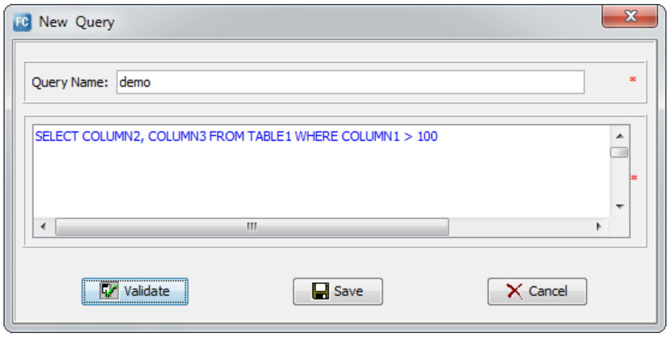
   

   | Query Name | Query | | |
   |---|---|---|---|
   | test | select COLUMN2, COL... | Delete | Edit |

   **Note:** This dialog box lists SQL queries that you previously defined. To edit or delete an existing SQL query, click **Edit** or **Delete** in the query row.

2. To add an SQL query, click **Add**.

   The **New Query** dialog box appears, as shown in the following image:

   Query Name: demo

   ```
   SELECT COLUMN2, COLUMN3 FROM TABLE1 WHERE COLUMN1 > 100
   ```

3. In the **Query Name** field, enter a unique query name.

   Fast Clone uses this value to name the query output file.

4. In the text box, type the query, or copy and paste it from an SQL editor.

5. Click **Validate** to validate the query.

   **Note:** The Fast Clone Console does not execute the query when validating it.

6. If the validation is successful, click **Save** to save the query.

   The query appears in the SQL query list.

### Related Topics:

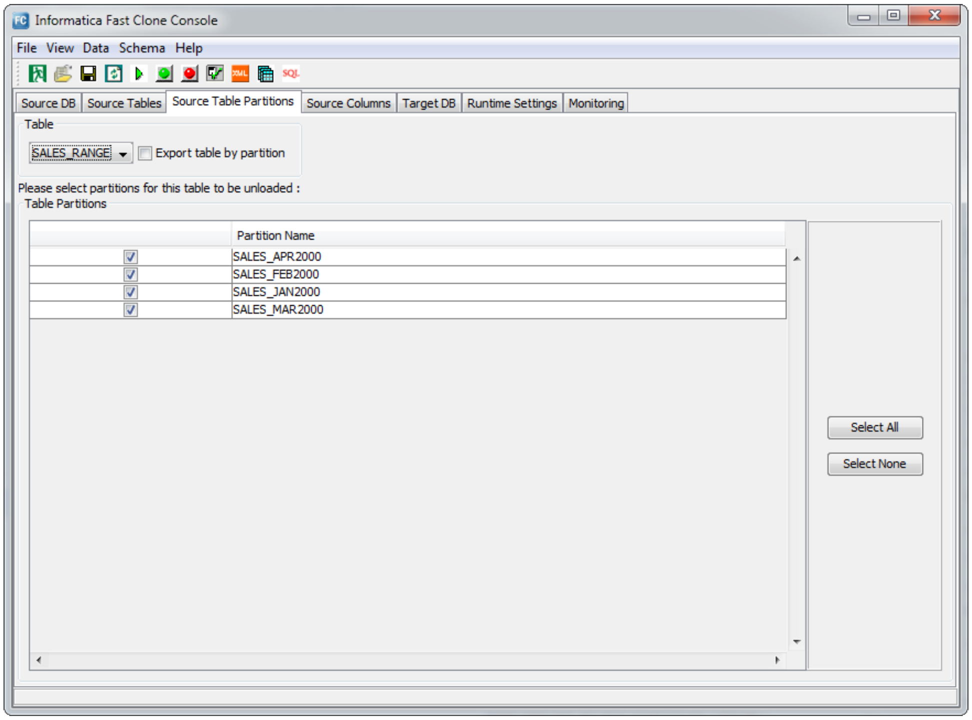
- "Unloading Data from Views" on page 73

# Customizing Unload Processing for Table Partitions

For partitioned source tables, Fast Clone selects all partitions by default for unload processing. You can exclude certain partitions. Also, for the partitions that you select, you can write the unloaded data to separate output files.

1. Click the **Source Table Partitions** tab.

   The following image shows this tab:



Table

SALES_RANGE ☐ Export table by partition

Please select partitions for this table to be unloaded :

Table Partitions

| | Partition Name |
|---|---|
| ☑ | SALES_APR2000 |
| ☑ | SALES_FEB2000 |
| ☑ | SALES_JAN2000 |
| ☑ | SALES_MAR2000 |

2. Under **Table**, select a partitioned source table for which to exclude some partitions from unload processing.

   The table partitions appear in the **Table Partitions** box.

3. To exclude a table partition, clear the check box in the partition row.

   **Tip:** If you want to exclude most table partitions, click **Select None** to clear all partitions and then select the few that you want to include. Click **Select All** to include all of the partitions again.

4. To write the source data that is unloaded from each selected table partition to a separate output data file, select the **Export table by partition** option.

   This option is cleared by default. With the default setting, Fast Clone uses the **Export by partitions when all selected** option on the **Runtime Settings** tab > **Miscellaneous Conditions** view to determine whether to write the source data from all table partitions to separate data files.

# Customizing Unload Processing for Table Columns

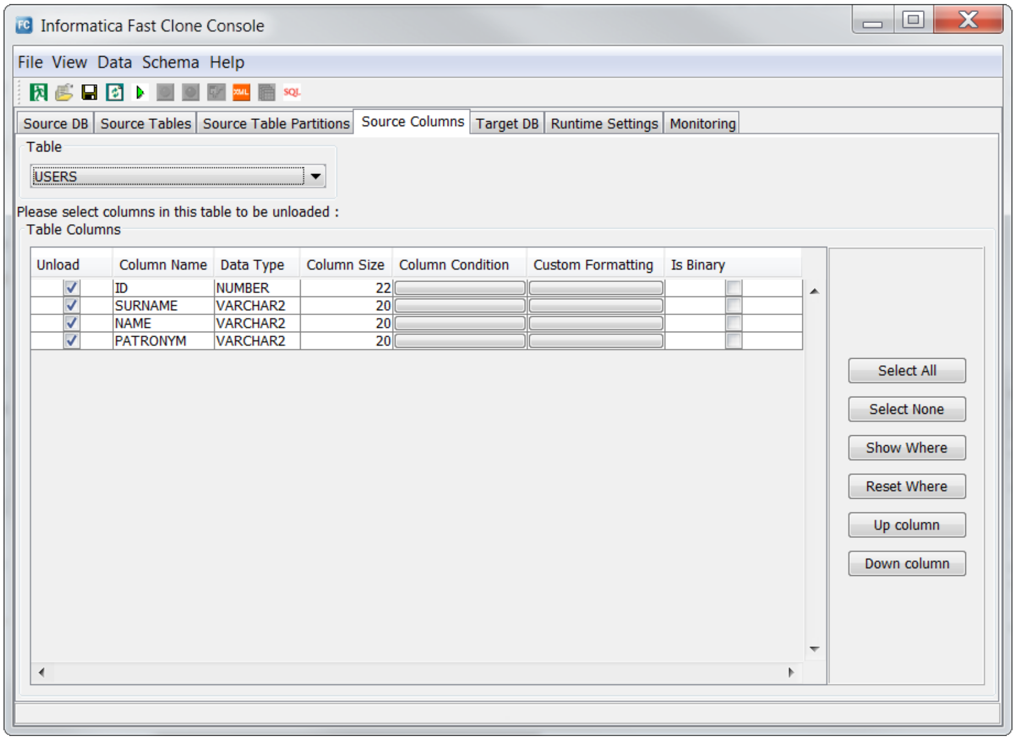
On the **Source Columns** tab, you can exclude source table columns from unload processing, define WHERE clauses for filtering data, customize date and timestamp formats, control numeric precision and padding, and change the order of columns in the output files.

## Excluding Table Columns from Unload Processing

By default, Fast Clone unloads data from all of the columns in a selected source table. You can optionally exclude particular columns from unload processing, such as those with unsupported datatypes.

1. Click the **Source Columns** tab.

   The following image shows this tab:



Table

USERS

Please select columns in this table to be unloaded :

Table Columns

| Unload | Column Name | Data Type | Column Size | Column Condition | Custom Formatting | Is Binary |
|---|---|---|---|---|---|---|
| ☑ | ID | NUMBER | 22 | | | □ |
| ☑ | SURNAME | VARCHAR2 | 20 | | | □ |
| ☑ | NAME | VARCHAR2 | 20 | | | □ |
| ☑ | PATRONYM | VARCHAR2 | 20 | | | □ |

2. In the **Table** list, select a source table for which to exclude columns from unload processing.

   All of the table columns appear in the **Table Columns** box.

3. To exclude a column, clear the **Unload** option in the column row.

   **Important:** Ensure that you exclude all of the columns that have unsupported datatypes. For more information about supported Oracle datatypes, see "Supported Datatypes for Oracle Sources" on page 16.

   **Tip:** If you want to exclude most of the columns, click **Select None** to clear all columns and then select the few columns that you want to include. Click **Select All** to include all of the columns again.

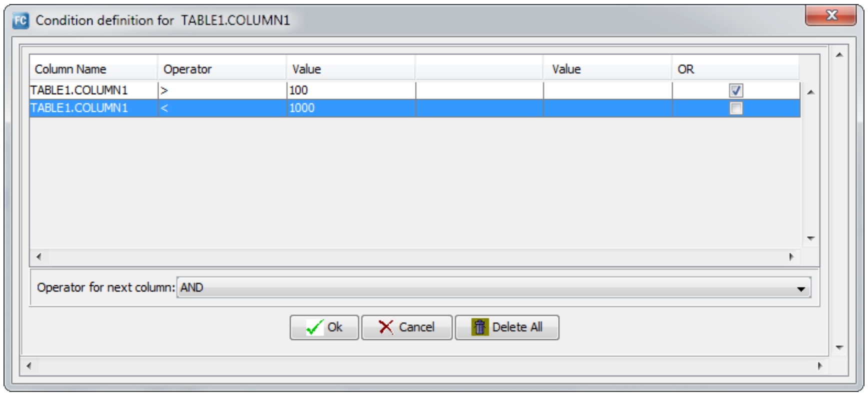
## Filtering Column Data with a WHERE Clause

To filter source data for an unload job, you can define a WHERE clause for one or more source columns. The Fast Clone Console provides a convenient tool for composing WHERE clauses.

1. Click the **Source Columns** tab.
2. In the **Table** list, select the table for which you want to define a WHERE clause for filtering column data.

   All of the table columns appear in the **Table Columns** box.
3. Right-click a column row for which to add a condtition in a WHERE clause and click the button in the **Column Condition** column.

   The **Condition definition for table.'column_name'** dialog box appears, as shown in the following image:



Condition definition for TABLE1.COLUMN1

| Column Name | Operator | Value | | Value | OR |
|---|---|---|---|---|---|
| TABLE1.COLUMN1 | > | 100 | | | ☑ |
| TABLE1.COLUMN1 | < | 1000 | | | □ |

Operator for next column: AND

4. In the **Operator** list, select one of the following comparison operators: =, <>, <, >, >=, <=, IS NULL, IS NOT NULL, LIKE, or BETWEEN.

   The operators that are available depend on the column datatype.
5. In the **Value** field, enter a value to use with the operator.

   If the operator requires one value, leave the second field blank. For example, with the greater than (>) operator, you could enter a single numeric value of 1.
6. If you want to add another condition for the same column, select the **OR** check box. Then define another condition for the current column.

   **Note:** If you clear the **OR** check box, the Fast Clone Console removes any conditions that follow the first one.
7. If you want to add a condition for another column, select one of the following options in the **Operator for next column** list to determine if the conditions for one or both columns must be true to filter data. Options are:

   - **OR**. The condition for one or more columns in the WHERE clause must be true.
   - **AND**. The conditions for all of columns in the WHERE clause must be true.

   Then define the conditions for the next column.
8. Click **OK**.
9. On the **Source Columns** tab, verify the WHERE clause that you defined for the table columns by clicking **Show Where**.

   The Fast Clone Console includes all of the conditions that you defined for the table columns to the WHERE clause.

   **Tip:** To remove the WHERE clauses, click **Reset Where**.

# Specifying a Custom Format for a Source Column

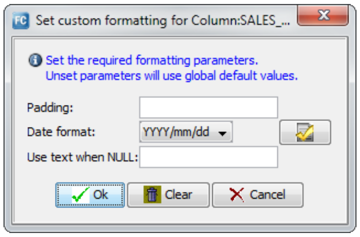
You can customize the date and timestamp formats that Fast Clone uses to unload data from source DATE and TIMESTAMP columns. You can also change the precision for NUMBER columns, add padding, and replace null values with another value.

1. On the **Source Columns** tab, select a table for which to customize the column format.

   All of the table columns appear in the **Table Columns** area.

2. Click a source column row for which to customize formatting, and then click the button in the **Custom Formatting** column.

   The **Set custom formatting for Column:'column_name'** dialog box appears, as shown in the following image:

   Set custom formatting for Column:SALES_...

   Set the required formatting parameters.
   Unset parameters will use global default values.

   Padding:

   Date format: YYYY/mm/dd

   Use text when NULL:

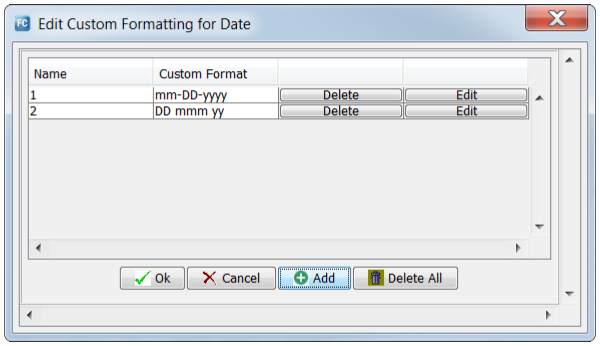
   The fields that appear in this dialog box vary depending on the column datatype.

3. If the selected column is a DATE or TIMESTAMP column, optionally define a custom date or timestamp format that overrides the format that is specified on the **Runtime Settings** tab > **Format Settings** view. To define a custom date or timestamp format, complete the following steps:

   a. Click the button next to the **Date format** or **Timestamp format** field.

      Alternatively, on the menu bar, click **Data** > **Manage Custom Formats** and select **Custom Date Formats** or **Custom Timestamp Formats**.

      The **Edit Custom Formatting for Date** or **Edit Custom Formatting for Timestamp** dialog box appears, as shown in the following image:

      Edit Custom Formatting for Date

      | Name | Custom Format | | |
      |---|---|---|---|
      | 1 | mm-DD-yyyy | Delete | Edit |
      | 2 | DD mmm yy | Delete | Edit |

   b. Click **Add** to add a custom date or timestamp format.

      **Tip:** To edit a previously defined format, click **Edit** on the custom format row. To delete previously defined format, click **Delete** on the custom format row.

   c. Enter a custom date or timestamp format and click **Save**.

      The **Set custom formatting for Column:'column_name'** dialog box appears. To use the new custom format, select it in the **Date format** or **Timestamp format** field. These values override formats that you defined on the **Runtime Settings** tab > **Format Settings** view.

4. Enter other custom format settings for the selected column.

The following table describes the other fields that might be available depending on the column datatype:

| Field | Description |
|---|---|
| Padding | The amount of padding to use for the column data. |
| Number precision | The precision to use for numeric data. This field is available only for columns with the NUMBER datatype.<br>This precision overrides the precision that you defined in the **Truncate numeric precision integer** field on the **Runtime Settings** tab > **Format Settings** view. |
| Use text when NULL | A value that replaces null values in the column.<br>**Note:** If you specify this value, to generate valid output files, you must clear the **Suppress trailing null columns** option on the **Runtime Settings** tab > **Format Settings** view. |

**Tip:** To remove all custom format settings, click **Clear**.

5. Click **OK**.

## Changing the Order of Columns in Output Data Files

By default, Fast Clone unloads columns in the order that they were created in the source tables. Optionally, you can change the order in which Fast Clone writes column values to output data files.

1. Click the **Source Columns** tab.
2. In the **Table** list, select the table for which you want to change the order of columns.
3. In the **Table Columns** list, select a column that you want to reposition in the output data file.
4. Move the column up or down by clicking the **Up column** or **Down column** button.

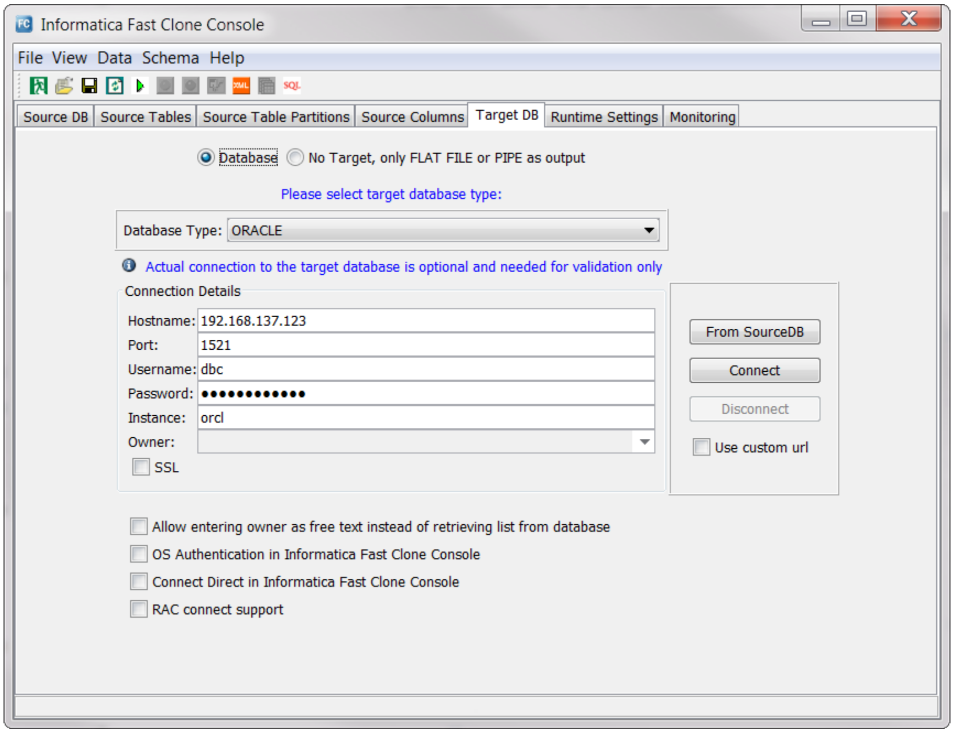
# Defining a Target

To define a target, enter connection information and set a few additional options. The Fast Clone Console saves this information to the configuration file and uses it to generate load scripts. When you run the Fast Clone executable, it uses the connection information to generate a connection string for the target.

After you enter the target connection information, you can connect to the target database to generate or validate the target schema and to manage target constraints that are used for loading data.

1. Click the **Target DB** tab.

   The following image shows the **Target DB** tab for an Oracle target:



● Database ○ No Target, only FLAT FILE or PIPE as output

Please select target database type:

Database Type: ORACLE

Actual connection to the target database is optional and needed for validation only

Connection Details

Hostname: 192.168.137.123

Port: 1521

Username: dbc

Password: ●●●●●●●●●●●●

Instance: orcl

Owner:

□ SSL

□ Use custom url

□ Allow entering owner as free text instead of retrieving list from database

□ OS Authentication in Informatica Fast Clone Console

□ Connect Direct in Informatica Fast Clone Console

□ RAC connect support

2. Select the **Database** option to make the connection fields available.

   This option must be selected for Fast Clone to unload data into output files and generate the load scripts for the target database.

3. Enter connection information for the target in one of the following ways:
   - Manually enter connection information for the target on the **Target DB** tab.
   - Click **From SourceDB** to use the source connection information for the target.

The following table describes the connection fields on the **Target DB** tab:

| Field | Description | Target Types |
|---|---|---|
| Database Type | Select one of the following target types:<br>- CLOUDERA<br>- GREENPLUM/BIZGRES<br>- HIVE<br>- HORTONWORKS<br>- MAPR<br>- MS SQL SERVER<br>- NETEZZA<br>- ORACLE<br>- TERADATA<br>- VERTICA<br>The default value is ORACLE. | All |
| Version | The version of the selected target type. | - Cloudera<br>- Hortonworks<br>- Hive<br>- MapR |
| Hadoop version | For Hive targets, the version of Hadoop for the Hive database. | Hive |
| Hostname | The host name or IP address of the system where the target instance runs. The default value is the local host name. | All |
| Port | The port number that Fast Clone uses to connect to the target. The default value depends on the target type. | All |
| Username | A user ID that has the privileges that Fast Clone requires to write data to the target.<br>**Note:** For Hive targets, this field name is **Hadoop username**. | All |
| Password | A valid password for the specified user. | - Greenplum/ Bizgres<br>- Microsoft SQL Server<br>- Netezza<br>- Oracle<br>- Teradata<br>- Vertica |
| Instance | An Oracle or Teradata instance name. | - Oracle<br>- Teradata |

| Field | Description | Target Types |
|---|---|---|
| Database | The target name.<br>To select the target name from the **Database** list, you must first connect to the target.<br>To manually enter the database name without connecting to the target, select the **Allow entering owner as free text instead of retrieving list from database** option.<br>**Note:** Hive, Netezza, and Vertica targets do not have the option to manually enter a database name. | - Greenplum/ Bizgres<br>- Hive<br>- Microsoft SQL Server<br>- Netezza<br>- Teradata<br>- Vertica |
| Location | For Hive targets, the location of flat files that store table data on the Hadoop Distributed File System (HDFS) for a selected Hive database. The Fast Clone Console gets this location from the Hive target. | Hive |
| Owner | The target owner name.<br>To select the owner name from the **Owner** list, you must first connect to the target.<br>To manually enter the owner name without connecting to the target, select the **Allow entering owner as free text instead of retrieving list from database** option. | Oracle |
| Path | The directory in a distributed file system for the output files. Click **Select** to browse to the directory, or enter the path to the directory.<br>**Important:** The user name that you specify for connecting to the target must have write permissions on this directory. | - Cloudera<br>- Hortonworks<br>- MapR |
| SSL | Select this option to configure Fast Clone to use the TCP/IP protocol with the Secure Sockets Layer (SSL), also called the TCPS protocol, to connect to the Oracle target.<br>**Important:** If you use TCPS connections between the Fast Clone Console and remote Oracle databases, you must install the Oracle Instant Client on the Fast Clone Console system. | Oracle |
| Schema | A Greenplum schema name.<br>To select the owner name from the **Schema** list, you must first connect to the target.<br>To manually enter the schema name without connecting to the target, select the **Allow entering owner as free text instead of retrieving list from database** option. | Greenplum |
| OS Authentication in Informatica Fast Clone Console | Select this option to configure Fast Clone to connect to a target by using the operating system user. | - Microsoft SQL Server<br>- Oracle |

| Field | Description | Target Types |
|---|---|---|
| Connect Direct in Informatica Fast Clone Console | Select this option to connect to the target directly without using a database instance. | - Microsoft SQL Server<br>- Oracle |
| Allow entering owner as free text instead of retrieving list from database | If you do not connect to the target to enter a database, owner, or schema name from a list, select this option to manually enter this information. | - Cloudera<br>- Greenplum/ Bizgres<br>- Hortonworks<br>- MapR<br>- Microsoft SQL Server<br>- Oracle<br>- Teradata |
| RAC connect support | Select this option if you load data to an Oracle database in a RAC environment. To connect to the Oracle RAC, the Fast Clone Console uses an instance name that is specified in the **Instance** field and the Fast Clone executable uses a service name that is specified in the **Service** field. | Oracle in a RAC environment |
| Use custom url | Select this option to configure the Fast Clone executable and Fast Clone Console to use different connection settings. In this case, Fast Clone uses the following connection settings to connect to the target:<br>- The Fast Clone executable uses the connection information that you provide under **Connection Details** to generate a connection string.<br>- The Fast Clone Console uses the connection string that you enter in the **JDBC custom URL** field. | All |
| JDBC custom URL | If you selected the **Use custom url** option, specifies a connection URL for the JDBC driver that the Fast Clone Console uses to connect to an Oracle source. This URL has the following format:<br>`jdbc:oracle:thin@[host][:port]:SID`<br>For the SSL connection to an Oracle target, use the following format:<br>`jdbc:oracle:oci@[host][:port]:SID`<br>If you enter an SSL connection URL, clear the **SSL** option.<br>**Important:** JDBC custom connection URLs always use the values that are specified in the **Username** and **Password** fields. Ensure that any JDBC custom connection URL that you specify does not include a user name and a password. | All |
| Service | An Oracle service name that is defined in a SERVICE_NAME environment variable. | Oracle in a RAC environment |

**Note:** If you do not enter the target connection information, the Fast Clone Console tries to use values that you entered for the source database on the **Source DB** tab.

4. Click **Connect**.

   Verify that the target connection is successful.

   If you did not manually enter the database, owner, and schema, which are required for some target types, you must enter this information at this point.

   **Note:** Fast Clone does not require you to connect to the target to save the target connection information to the configuration file.

## Defining ASCII Flat Files as Targets

You can configure Fast Clone to unload data into ASCII flat files. In this case, Fast Clone does not generate a load script for the target database load utility.

1. Click **Target DB** tab.
2. Select the **No Target, only FLAT FILE or PIPE as output** option.

   When you run data unload jobs, Fast Clone unloads data into ASCII flat files in the output directory.

## Defining Hadoop Targets

You can configure Fast Clone to unload source data to output files on a Cloudera, Hortonworks, or MapR distribution of Hadoop or to a Hive data warehouse target. With these Hadoop target types, Fast Clone does not generate a load script for a target load utility.

1. On the **Target DB** tab, select **CLOUDERA**, **HIVE**, **HORTONWORKS**, or **MAPR** in the **Database Type** list.
2. Enter the version, host name, port number, and user name for connecting to the target.

   For MapR targets, you can specify a cluster name instead of a host name in the **Hostname** field. Use the following syntax:

   ```
   /mapr/cluster_name
   ```

   **Important:** The *cluster_name* value must match the cluster name in the $MAPR_HOME/conf/mapr-clusters.conf configuration file.

   When you specify a cluster name, the Fast Clone Console does not use the port number that you specify in the **Port** field.
3. To specify the directory in the distributed file system for the Cloudera, Hortonworks, or MapR output files, click **Select** and then either browse to the directory or enter the path to the directory in the **Path** field.

   The user name that you specify for connecting to the Hadoop target must have write permissions on this directory.
4. Click **Connect**.

   The Fast Clone Console connects to the target.

   **Important:** The Fast Clone Console requires Java Runtime Environment (JRE) 1.7 x64 or later to connect to these Hadoop targets.
5. For Hive targets, select a database in the **Database** list.

   The Fast Clone Console uses the definition of the selected database to determine the location of flat files that store table data on the HDFS.

Before you unload source data to a Hive target, enter the field delimiter that the Hive target uses in data files in the **Column separator** field on the **Runtime Settings** tab > **Format Settings** view. By default, Hive data warehouses use the ASCII start of heading (SOH) character as the field delimiter.

# Configuring Runtime Settings

On the **Runtime Settings** tab, you can configure runtime settings for data-cloning jobs.

Fast Clone provides the following types of runtime settings:

- Formatting options for output files
- Rules for replacing character sequences in the source data that the target load utility does not support with other character sequences
- Separator and delimiter characters in data to be loaded
- Use of parallel processing for direct path unload and conventional path unload processing
- Names or naming criteria for the configuration file, data files, unload scripts, and load scripts
- Path to the output directory
- Criteria for splitting large output files into chunks
- Options for generating loader input only or for unloading data without generating loader input
- Miscellaneous options, such as those for performing dirty reads, writing data to pipes, encrypting passwords in the configuration file, and compressing data on the fly
- Remote Fast Clone Server options
- Load settings for the target load utility
- Settings for integration with Informatica Data Replication

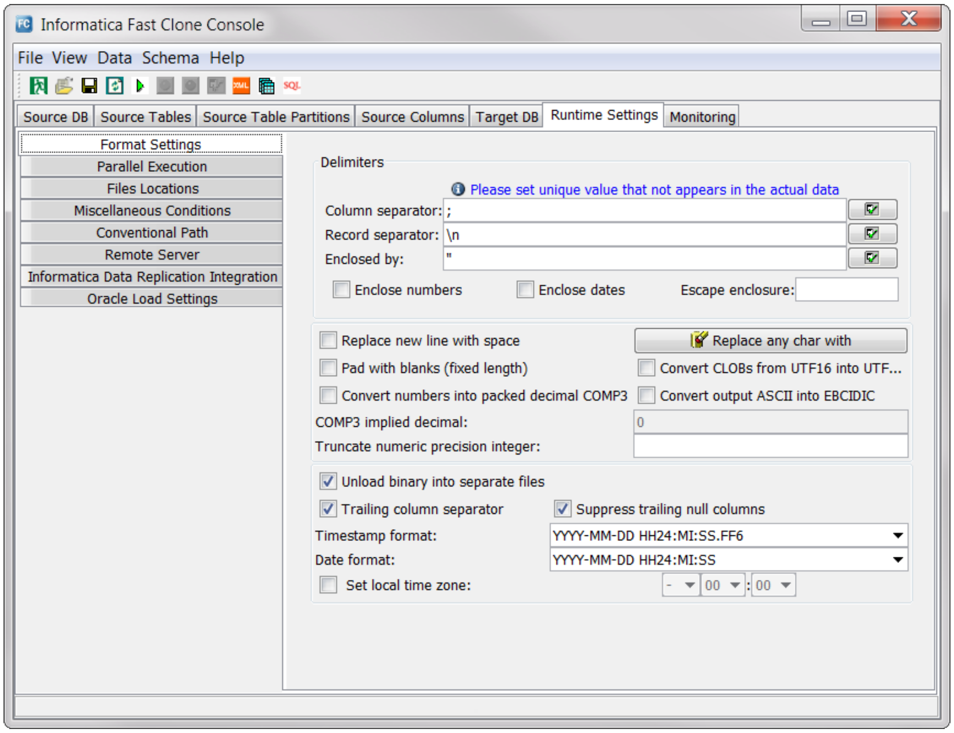
The Fast Clone Console saves these settings to the configuration file, except for the configuration file name and path to unload script.

## Specifying Output File Format Settings for the Loader

You can configure format settings for the output files to conform with the requirements of your target load utility. With the correct format settings, the loader can accurately load data to the target database.

1. Click the **Runtime Settings** tab > **Format Settings** view.

   The following image shows this view:



Delimiters

Please set unique value that not appears in the actual data

Column separator: ;

Record separator: \n

Enclosed by: "

□ Enclose numbers □ Enclose dates Escape enclosure:

□ Replace new line with space

□ Pad with blanks (fixed length)

□ Convert CLOBs from UTF16 into UTF...

□ Convert numbers into packed decimal COMP3

□ Convert output ASCII into EBCIDIC

COMP3 implied decimal: 0

Truncate numeric precision integer:

☑ Unload binary into separate files

☑ Trailing column separator

☑ Suppress trailing null columns

Timestamp format: YYYY-MM-DD HH24:MI:SS.FF6

Date format: YYYY-MM-DD HH24:MI:SS

□ Set local time zone: - 00 : 00

2. Enter information in any of the fields or accept the default values.

The following table describes the fields in the **Format Settings** view:

| Field | Description | Default |
|---|---|---|
| Column separator | A character or character sequence that Fast Clone uses to separate columns in the output data files. Specify a character or character sequence that does not appear in the source data. Otherwise, the target load utility cannot correctly parse the data files. For more information, see "Selecting Column and Row Separator and Enclosure Character to Use for Data Loading" on page 50.<br>Fast Clone allows you to enter column separators that use the \x*hh* escape sequence in hexadecimal notation.<br>Also, you can use the following non-printable ASCII characters:<br>- \a (alert)<br>- \b (backspace)<br>- \n (new line)<br>- \r (carriage return)<br>- \t (horizontal tab)<br>- \v (vertical tab)<br>If you use non-printable characters as column separators, ensure that the target load utility supports these separators. For more information, see the target database documentation. | - ASCII start of heading (SOH) for Hive targets<br>- semicolon (;) for other targets |
| Record separator | A character or character sequence that Fast Clone uses to separate table rows in the output data files. Specify a character or character sequence that does not appear in the source data. Otherwise, the target load utility cannot correctly parse the data files. For more information, see "Selecting Column and Row Separator and Enclosure Character to Use for Data Loading" on page 50.<br>Fast Clone allows you to enter row separators that use the \x*hh* escape sequence in hexadecimal notation.<br>Also, you can use the following non-printable ASCII characters:<br>- \a (alert)<br>- \b (backspace)<br>- \n (new line)<br>- \r (carriage return)<br>- \t (horizontal tab)<br>- \v (vertical tab)<br>If you use non-printable characters in row separators, ensure that the target load utility supports these separators. For more information, see target database documentation. | \n |
| Enclosed by | A character or character sequence that Fast Clone uses to enclose character data in the output data files. Specify a character or character sequence that does not appear in the source data. Otherwise, the target load utility cannot correctly parse the data files. | double-quotation mark (") |
| Enclose numbers | Indicates whether to enclose numeric data with the delimiter value that you specified in the **Enclosed by** field. | Not selected |

| Field | Description | Default |
|---|---|---|
| Enclose dates | Indicates whether to enclose DATE and TIMESTAMP data with the delimiter value that you specified in the **Enclosed by** field. | Not selected |
| Escape enclosure | An escape character that Fast Clone inserts before the value that is specified in the **Enclosed by** field whenever that value appears in data in text columns. If you enter an escape character, Fast Clone truncates the **Enclosed by** value to a single character. | backslash (\) |
| Replace new line with space | Indicates whether to replace the line break character (\n) with the space character in unloaded data. Select this option if you use the line break character as the **Record Separator** value. | Not selected |
| Replace any char with | Click this button to define rules for replacing particular character sequences in the source data with another character sequence. For more information, see "Defining Rules for Replacing Character Sequences in Data" on page 50. | - |
| Pad with blanks (fixed length) | Select this option to unload data to output data files that use fixed-length record format. If the number of bytes of field data is less than the field length, Fast Clone pads the field on the right with the space characters. | Not selected |
| Convert CLOBs from UTF16 into UTF8 | Select this option to convert source character data in CLOB columns from UTF-16 to UTF-8 encoding in the output data files. If you do not use UTF-16 encoding for the source data, clear this option to avoid the overhead of character set conversion. | Not selected |
| Convert numbers into packed decimal COMP3 | Indicates whether to write numeric data to the output files as COMP-3 values. Select this option if you need to read the output file with COBOL or another language that has built-in COMP-3 support. | Not selected |
| COMP3 implied decimal | The COMP-3 implied decimal value. | 0 |
| Convert output ASCII into EBCIDIC | Select this option if you need the output to be in IBM EBCIDIC format. | Not selected |
| Truncate numeric precision integer | The number of significant digits that Fast Clone uses to round down the fractional part of numeric values in the output. Valid values are from 1 to 36.<br>You can override this value for individual source numeric columns. For more information, see "Specifying a Custom Format for a Source Column" on page 38. | No default |

| Field | Description | Default |
|---|---|---|
| Unload binary into separate files | Indicates whether to write data from BLOB, CLOB, NCLOB, LONG, and LONG RAW source columns to separate files. In the data files, Fast Clone includes the file name instead of the source value. | - Not selected for Netezza targets that use DataStreamer and Hive targets<br>- Selected for other targets |
| Trailing column separator | Indicates whether to append a column separator after the last column in each table row.<br>**Note:** For Oracle targets, if you unload data from binary columns to separate files, Fast Clone does not append the trailing column separator after the last column if that column has the BLOB, CLOB, or XMLTYPE datatype. | - Not selected for Netezza targets that use DataStreamer<br>- Selected for other targets |
| Suppress trailing null columns | Indicates whether to omit the last column value in the output data file if the column contains a null value. | - Not selected for Netezza targets that use DataStreamer<br>- Selected for other targets |
| Timestamp format | The format to use for timestamp values from TIMESTAMP source columns. For more information about valid TIMESTAMP formats, see the Oracle documentation.<br>You can override this format for individual TIMESTAMP columns. For more information, see "Specifying a Custom Format for a Source Column" on page 38. | YYYY-MM-DD HH24:MI:SS.FF6 |
| Date format | The format to use for date values from DATE source columns. For more information about valid DATE formats, see the Oracle documentation.<br>You can override this format for individual DATE columns. For more information, see "Specifying a Custom Format for a Source Column" on page 38. | YYYY-MM-DD HH24:MI:SS |
| Set local time zone | Select this option to unload values from columns that have the TIMESTAMP WITH LOCAL TIMEZONE datatype. Then enter your local time zone offset. | Not selected |

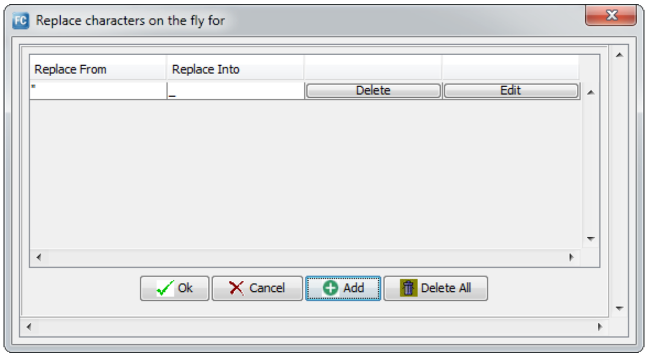
## Defining Rules for Replacing Character Sequences in Data

You can define rules for replacing particular character sequences in the source data with another character sequence. If the source data includes character sequences that the target load utility does not support, define replacement rules to conform to the format requirements of the target load utility.

1. On the **Runtime Settings** tab > **Format Settings** view, click **Replace any char with**.

   Alternatively, click **Data** > **Manage replacements on fly** on the menu bar.

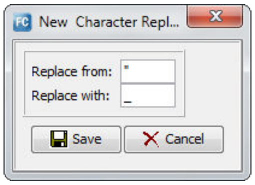
   The **Replace characters on the fly for** dialog box appears, as shown in the following image:



| Replace From | Replace Into |
|---|---|
| " | _ |

2. Click **Add** to add a replacement rule.

   The **New Character Replacement** dialog box appears, as shown in the following image:



Replace from: "
Replace with: _

3. Enter appropriate values in the **Replace from** and **Replace with** fields and click **Save**.

   You can use non-printable characters to specify character sequences.

4. Click **Ok** to save all of the replacement rules.

   **Note:** Click **Edit** to edit an existing replacement rule. Click **Delete** to delete an existing replacement rule.

## Selecting Column and Row Separator and Enclosure Character to Use for Data Loading

To perform accurate data load to a target database, select a valid column and row separator and enclosure character that does not appear in the source data. The Fast Clone Console provides a convenient validation tool that you can use to select valid values for column and row separator and enclosure character using sampling.

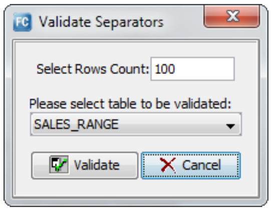
The validation tool checks table data in the sample row set and reports an error if a control character sequence appears in the source data.

Validation of the source data is a time-consuming operation that is comparable to the data unload operation. Use the validation tool for small source tables. For large tables, select a row and column separator and enclosure character sequence that includes multiple characters and is unlikely to appear in the source data.

1. Click the **Runtime Settings** tab > **Format Settings** view.

2. Start the validation tool.
   - For the column separator, click the button to the right of the **Column separator** field.
   - For the row separator, click the button to the right of the **Record separator** field.
   - For the enclosure character, click the button to the right of the **Enclosed by** field.

   The dialog box of the validation tool appears, as shown in the following image:

   Select Rows Count: 100

   Please select table to be validated: SALES_RANGE

3. In the list, select a source table that you want to validate.
4. In the **Select Rows Count** field, enter number of rows in the source table to validate.

   The default value is 100 rows.
5. Click **Validate** and review the validation results.

   If the Fast Clone Console reports a validation error, either select another value for the row or column separator or enclosure character, or define a replacement rule for the specified character value.

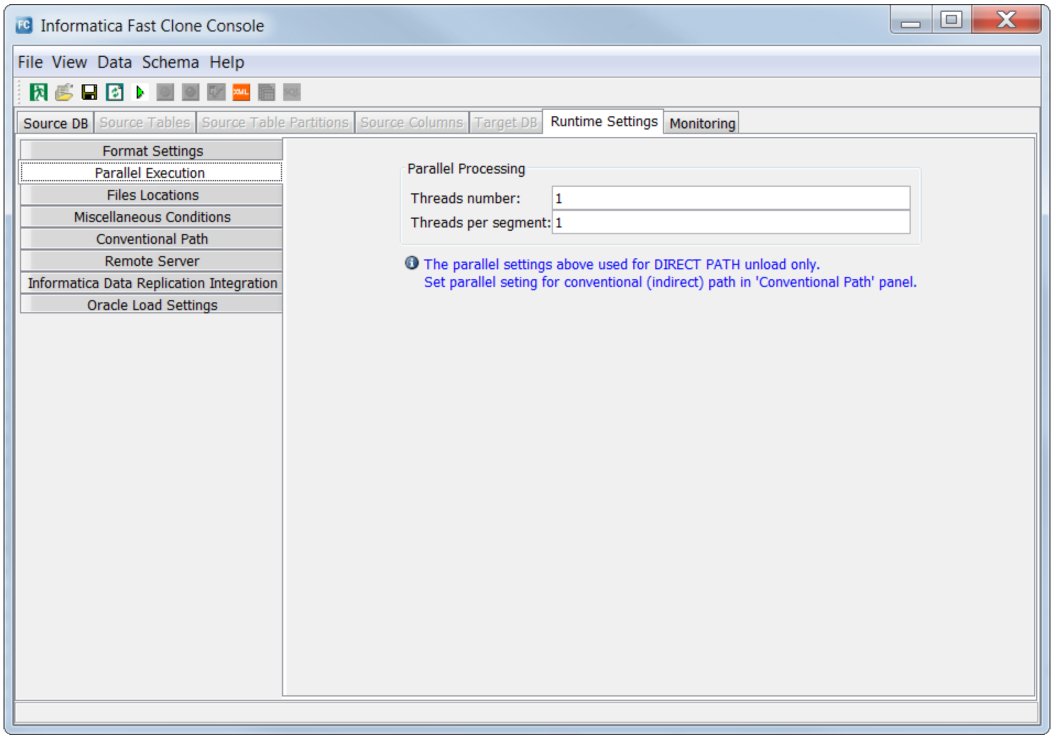
## Configuring Parallel Processing for Direct Path Unload Jobs

When cloning large amounts of data with the direct path unload method, you can improve performance by configuring parallel processing on multiple threads.

With the direct path unload method, Fast Clone uses two threads per core or CPU to unload data from an Oracle source. If the source system has multiple CPUs, Fast Clone can use parallel processing to unload data from source tables.

1. Click the **Runtime Settings** tab > **Parallel Execution** view.

   The following image shows this view:



Threads number: 1
Threads per segment: 1

The parallel settings above used for DIRECT PATH unload only.
Set parallel seting for conventional (indirect) path in 'Conventional Path' panel.

2. Enter information in the following fields:

   **Threads number**

   The maximum number of tables or partitions from which data can be unloaded in parallel. The default value is 1.

   **Threads per segment**

   The maximum number of threads that can be used to unload data from a single table or partition. If you use multiple threads for a single table, CPU usage might increase. Usually, multiple threads are used to unload large tables. The default value is 1.

   To get the total number of unload threads, multiply these field values. For the direct path unload method, calculate the appropriate number of concurrent threads as follows:

   ```
   number of CPUs or cores x 2
   ```

   If I/O bandwidth or the number of allocated CPUs is restricted, enter lower field values that are compatible with the appropriate concurrency for the system.

   If you need to unload data from many tables of similar size, set **Threads number** to the appropriate concurrency and set **Threads per segment** to 1.

   If you need to unload data from a few large tables, increase the **Threads per segment** value and set **Threads number** to the number that you calculate as follows:

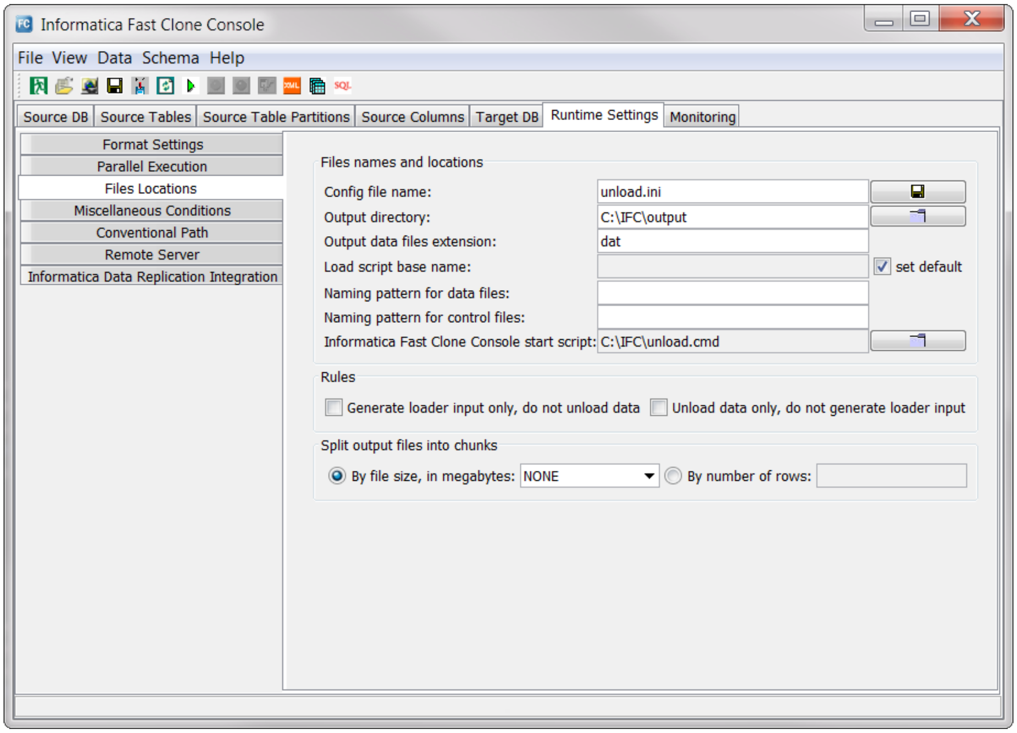
   ```
   (number of CPUs or cores x 2)÷ number of threads per table
   ```

# Specifying Fast Clone File Names and Locations

Optionally, you can override the default configuration file name, output directory, load script name, unload script location, and output file extension. You can also set options that control how the output files are generated and whether Fast Clone generates the load script.

1. Click the **Runtime Settings** tab > **Files Locations** view.

   The following image shows this view:



Files names and locations

Config file name: unload.ini

Output directory: C:\IFC\output

Output data files extension: dat

Load script base name: ☑ set default

Naming pattern for data files:

Naming pattern for control files:

Informatica Fast Clone Console start script: C:\IFC\unload.cmd

Rules

□ Generate loader input only, do not unload data □ Unload data only, do not generate loader input

Split output files into chunks

● By file size, in megabytes: NONE ○ By number of rows:

2. Enter information in any of the fields.

   The following table describes the fields in the **Files Locations** view:

| Field | Description |
|---|---|
| Config file name | The path and file name for the cloning configuration file. The path can be the full path or a path that is relative to the *FastClone_installation* directory. The Fast Clone Console generates this file when you save the configuration settings. When you run data unload jobs, the Fast Clone Console uses the specified configuration file. The default name is *FastClone_installation*\unload.ini. |
| Output directory | The full path to the directory to which Fast Clone writes the generated output files. The output directory can be on a local file system or on a network mapped share, such as NFS or Samba. If you do not specify an output directory, Fast Clone writes the output files to the *target_type*_output subdirectory in the current working directory.<br>**Tip:** Specify a different directory for each type of target database. |
| Output data files extension | A file name extension for the output data files. For each unloaded source table, Fast Clone generates a data file that has a file name in the format *table_name.extension*. The default extension is dat. |

| Field | Description |
|---|---|
| Load script base name | The load script name.<br>By default, Fast Clone uses the source schema name as the load script name. To use another name, clear the **set default** option and enter a script name.<br>On Windows, Fast Clone generates load script files with the .cmd extension. On Linux and UNIX, Fast Clone generates load script files with the .sh extension. |
| Naming pattern for output data files | Specifies the naming pattern that Fast Clone uses to create the base name of the output data files.<br>In the naming pattern, you can include the following variables:<br>- %&schema&% - Represents a default schema name for the user that Fast Clone uses to connect to the source database.<br>- %&source_schema&% - Represents a source schema name.<br>- %&target_schema&% - Represents a target schema name.<br>- %&table&% - Represents a table name.<br>- %&table_counter&% - Represents a sequence number of a table.<br>- %&partition&% - Represents a table partition name. This partition variable is required if you specify partitions in the [SOURCE_TABLES] or [SOURCE_INDIRECT_TABLES] sections of the configuration file.<br>- %&counter&% - Represents a sequence number of a data file for a table. This counter variable is required if Fast Clone splits the output data files based on the mb_per_file and split_files_by_row_count parameters.<br>- %&yyyy&% - Represents a year.<br>- %&mm&% - Represents a month.<br>- %&dd&% - Represents a day.<br>If you do not specify the naming pattern for output data files, Fast Clone uses the table name as the name for output data files. |
| Naming pattern for control files | The naming pattern that Fast Clone uses to generate names for control files, log files, and SQL scripts.<br>In the naming pattern, you can include the following variables:<br>- %&schema&% - Represents a default schema name for the user that Fast Clone uses to connect to the source database.<br>- %&source_schema&% - Represents a source schema name.<br>- %&target_schema&% - Represents a target schema name.<br>- %&table&% - Represents a table name.<br>- %&table_counter&% - Represents a sequence number of a table.<br>- %&partition&% - Represents a partition name.<br>The partition variable is required if you specify partitions in the [SOURCE_TABLES] or [SOURCE_INDIRECT_TABLES] sections of the configuration file.<br>- %&counter&% - Represents the sequence number of a control file for a table. This variable value is always 1.<br>- %&yyyy&% - Represents a year.<br>- %&mm&% - Represents a month.<br>- %&dd&% - Represents a day.<br>If you do not specify the naming pattern for control files, Fast Clone uses the table name as the name for control files, log files, and SQL scripts. |
| Fast Clone start script | The path and file name for the unload script that runs data unload jobs. The path can be the full path or a path that is relative to the *FastClone_installation* directory. By default, the Fast Clone uses *FastClone_installation*\unload.cmd script on Windows and *FastClone_installation*\unload.sh script on Linux and UNIX. |

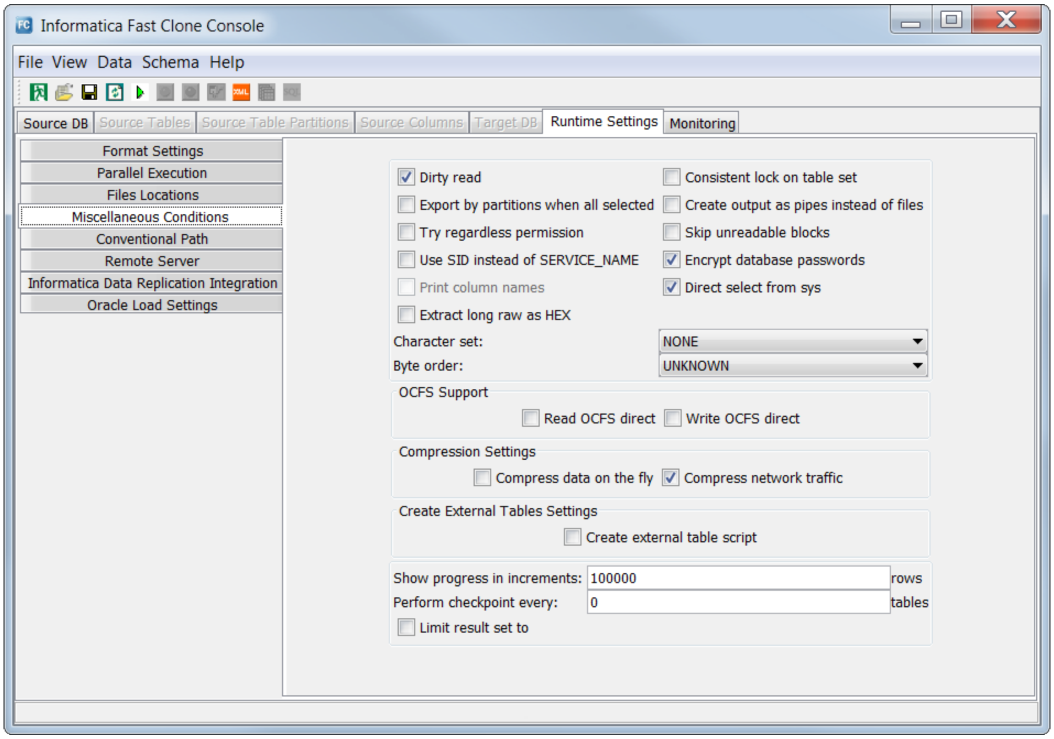
| Field | Description |
|---|---|
| Generate loader input only, do not unload data | Select this option to generate the load script but not unload data from the source. Use this option to prepare load scripts when unloading data to pipes.<br>By default, this option is not selected.<br>**Note:** Fast Clone does not generate load scripts and control files for flat file, Hadoop, and Hive targets. |
| Unload data only, do not generate loader input | After you generate the load script, select this option to unload data to pipes without regenerating the load script.<br>By default, this option is not selected. |
| Split output into chunks, By files size, in megabytes | Select this option to split output into multiple data files based on output chunk size. To specify the maximum amount of output in each file, in megabytes, either select a value from the list or type an appropriate value. If you select the default value of NONE, Fast Clone unloads data to a single data file for each table. |
| Split output into chunks, By number of rows | If you use the direct unload method, select this option to split output into multiple data files based on a fixed number of rows that each file can contain. Then enter the fixed number of rows.<br>**Note:** For parallel unload processing, splitting output into multiple data files based on a number of rows is slower than splitting output based on a chunk size. This performance degradation occurs because of locks between the parallel threads. |

## Configuring Miscellaneous Conditions

Optionally, you can set miscellaneous runtime parameters for data cloning. These parameters provide some important capabilities, such as writing output to pipes, encrypting passwords, compressing data, and configuring advanced settings for the Fast Clone executable.

1. Click the **Runtime Settings** tab > **Miscellaneous Conditions** view.

   The following image shows this view:



☑ Dirty read
☐ Consistent lock on table set
☐ Export by partitions when all selected
☐ Create output as pipes instead of files
☐ Try regardless permission
☐ Skip unreadable blocks
☐ Use SID instead of SERVICE_NAME
☑ Encrypt database passwords
☐ Print column names
☑ Direct select from sys
☐ Extract long raw as HEX

Character set: NONE

Byte order: UNKNOWN

OCFS Support

☐ Read OCFS direct ☐ Write OCFS direct

Compression Settings

☐ Compress data on the fly ☑ Compress network traffic

Create External Tables Settings

☐ Create external table script

Show progress in increments: 100000 rows

Perform checkpoint every: 0 tables

☐ Limit result set to

2. Enter information in any of the fields or accept the default values.

   The following table describes the fields in the **Miscellaneous Conditions** view:

| Field | Description | Default |
|---|---|---|
| Dirty read | If you use the direct path unload method, clear this option to lock the source tables for unload processing. When the tables are locked, Fast Clone can perform a consistent read.<br>By default, this option is selected and Fast Clone does not lock the source tables during unload processing. Fast Clone bypasses this Oracle consistency mechanism. However, Fast Clone preserves block-level consistency because block I/O operations are atomic. | Selected |
| Consistent lock on table set | If you use the direct path unload method and cleared the **Dirty read** option, select this option to lock all of the selected source tables in a table set before unloading the source data and to unlock these tables after unloading data from all of them. By default, this option is not selected and Fast Clone holds a lock on each source table only while unloading data from the table. | Not selected |

| Field | Description | Default |
|---|---|---|
| Export by partitions when all selected | If you unload data from partitioned source tables, indicates whether to write the data from each table partition to a separate output data file.<br>On the **Source Table Partitions** tab, you can override this setting for individual source tables. | Not selected |
| Create output as pipes instead files | Select this option to write output to a pipe instead of to a file system. Use this parameter for integration with other systems or scripts. Fast Clone creates a pipe file for each table prior to unloading table data.<br>**Note:** If you created a pipe file manually before unloading data, clear this option to have Fast Clone use the existing pipe file. | Not selected |
| Skip unreadable blocks | If you use the direct path unload method, select this option to skip blocks that Fast Clone cannot read from Oracle data files and continue unload processing.<br>By default, this option is not selected. When Fast Clone encounters a block it cannot read, it ends with an error and stops unload processing. | Not selected |
| Use SID instead of service name | Select this option to use the Oracle SID instead of the SERVICE_NAME when you cannot connect to the source instance by using the instance name. | Not selected |
| Encrypt database passwords | Indicates whether to encrypt database passwords in configuration files. Fast Clone applies the XOR operator to a password value by using the encryption key. Fast Clone then encrypts the resulting value by using the Base64 encryption schema.<br>By default, this option is selected and database passwords are encrypted. Clear this option if you do not want to use encrypted passwords. | Selected |
| Print column names | If you unload data to an ASCII flat file, select this option to have Fast Clone write column names in the first row of the output files.<br>By default, this option is not selected and Fast Clone does not write column names in the first row. | Not selected |
| Direct select from sys | Indicates whether to use SYS-owned dictionary tables, such as sys.obj$ and sys.tab$, to get Oracle extent boundaries. By default, Fast Clone uses SYS-owned dictionary tables to get extent boundaries. If you clear this option, Fast Clone uses the dba_extents and dba_segments views to get extent boundaries.<br>**Note:** For Oracle sources that use locally managed tablespaces, selecting data from the dba_extents view is slow. If the dictionary is partially analyzed and the optimizer mode is not RULE, selecting data from the dba_extent view might be very slow. Informatica recommends that you grant SELECT ANY DICTIONARY permissions to the user and use the default value for this option. | Selected |

| Field | Description | Default |
|---|---|---|
| Extract long raw as HEX | For Oracle targets, select this option to unload the BLOB and LONG RAW source data in hexadecimal format. For other target types, clear this option to avoid generating incorrect control files. | Not selected |
| Character Set | If you unload data to a Teradata target by using direct path unload method, indicates whether to convert the source character data to UTF-8. The default value is NONE, which causes Fast Clone to preserve the source database encoding. | NONE |
| Read OCFS Direct | For Oracle RAC sources on an OCFS on Linux or UNIX, indicates whether Fast Clone uses the direct I/O mode to read Oracle data files. The direct I/O mode minimizes the effects of cache on reading data from Oracle data files. Select this option if the Oracle data files are located on an OCFS. | Not selected |
| Write OCFS Direct | For Oracle RAC sources on an OCFS on Linux or UNIX, indicates whether Fast Clone uses the direct I/O mode to write the output files. The direct I/O mode minimizes effects of cache on writing the output files. Select this option if you configured Fast Clone to generate the output files on an OCFS. | Not selected |
| Compress data on the fly | Indicates whether to compress the output data files during unload processing to reduce the file size. | Not selected |
| Data compression | If you selected **Compress data on the fly**, select the compression method that Fast Clone uses to compress data. Options are:<br>- **GZIP**. Provides a better compression ratio but is slower.<br>- **QZIP**. Provides a lower compression ratio but is faster. | GZIP |
| GZIP compression level | If you selected GZIP as the data compression method, enter the compression level. Valid values are 1 to 9. Higher values provide faster compression with a lower compression ratio. Lower values provide slower compression with a higher compression ratio. | 1 |
| Compress network traffic | Select this option to compress network traffic between source and target servers when creating Fast Clone files on a remote server and using a Fast Clone Server. | Selected |
| Create external table script | For Oracle targets, select this option to generate an SQL script for creating Oracle external tables based on the output data files. | Not selected |
| Create external table log file | For Oracle targets, if you selected the **Create external table script** option, indicate whether to use a log file. If you select this option, the external table utility logs messages to this file when accessing data in the data file. | Not selected |

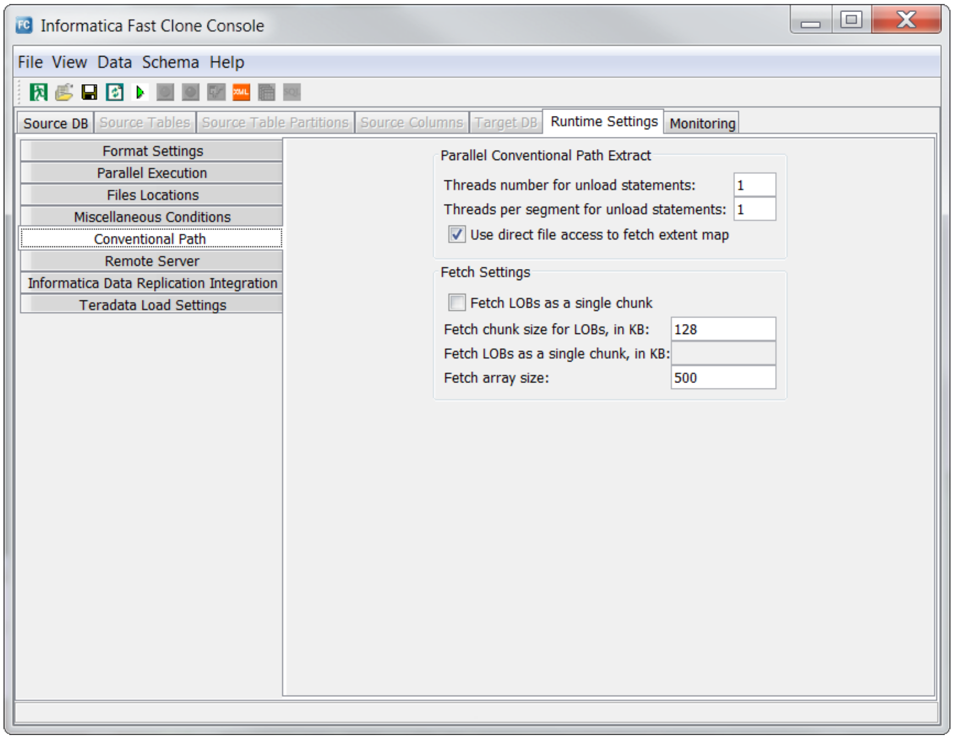
| Field | Description | Default |
|---|---|---|
| External table name includes partition | For Oracle targets, if you selected the **Create external table script** option, indicate whether the Oracle external table names include partition names in the output SQL script. Select this option to append partition names to external table names. | Not selected |
| Show progress in increments | The number of rows that determines the frequency at which Fast Clone updates progress information when a data unload job is running. Each time Fast Clone finishes processing the specified number of rows, Fast Clone updates progress information. | 100000 |
| Perform checkpoint every | The number of tables that determines the frequency with which Fast Clone performs checkpoints on the source database. Each time Fast Clone completes unloading the specified number of tables, Fast Clone performs a checkpoint on the source.<br>Frequent checkpoints can increase the overhead of unload processing on the source system and might degrade unload performance.<br>With the default value of 0, Fast Clone performs a checkpoint only prior to starting the data unload job. | 0 |
| Limit result set to | Select this option to specify the maximum number of rows that Fast Clone can unload from each table. Valid values are -1 to 1073741823.<br>By default, this option is not selected and Fast Clone uses the value of -1 to unload all of the rows from each table. | Not selected |

## Tuning Conventional Path Unload Processing

If you use the conventional path unload method, you can configure parameters for parallel unload processing and fetching LOB data to improve performance.

1. Click the **Runtime Settings** tab > **Conventional Path** view.

   The following image shows this view:



Parallel Conventional Path Extract

Threads number for unload statements: 1

Threads per segment for unload statements: 1

☑ Use direct file access to fetch extent map

Fetch Settings

□ Fetch LOBs as a single chunk

Fetch chunk size for LOBs, in KB: 128

Fetch LOBs as a single chunk, in KB:

Fetch array size: 500

2. To configure parallel processing, complete the following fields:

| Field | Description |
|---|---|
| Threads number for unload statements | The maximum number of tables or partitions from which data can be unloaded in parallel. The default value is 1. |
| Threads per segment for unload statements | The maximum number of threads that Fast Clone can use to unload data from a single table or partition. If you use multiple threads for a single table, CPU usage might increase. Usually, multiple threads are used to unload data from large tables. The default value is 1. |
| Use direct file access to fetch extent map | Select this option to have Fast Clone try to read extent information for parallel unload processing directly from a data file. By default, this option is selected. |

With the conventional path unload method, Fast Clone uses one thread per core or CPU to unload data from an Oracle source. If the system has multiple CPUs, Fast Clone can perform parallel unload processing for the source tables.

To determine the total number of unload threads to use, multiply the number of tables from which data is unloaded in parallel by the number of threads for each table. For conventional path unload, the appropriate number of concurrent threads is equivalent to the number of cores or CPUs. If I/O bandwidth or the number of allocated CPUs is restricted, you might need to use lower the value.

If you need to unload data from many tables of similar size, set **Threads number for unload statements** to the appropriate concurrency and set **Threads per segment for unload statements** to 1.

If you need to unload data from a few large tables, increase the **Threads per segment for unload statements** value and set **Threads number for unload statements** to the number that you calculate as follows:

```
number of cores or CPUs ÷ number of threads per table
```

3. To configure settings for fetching LOB data, complete the fields described in the following table:

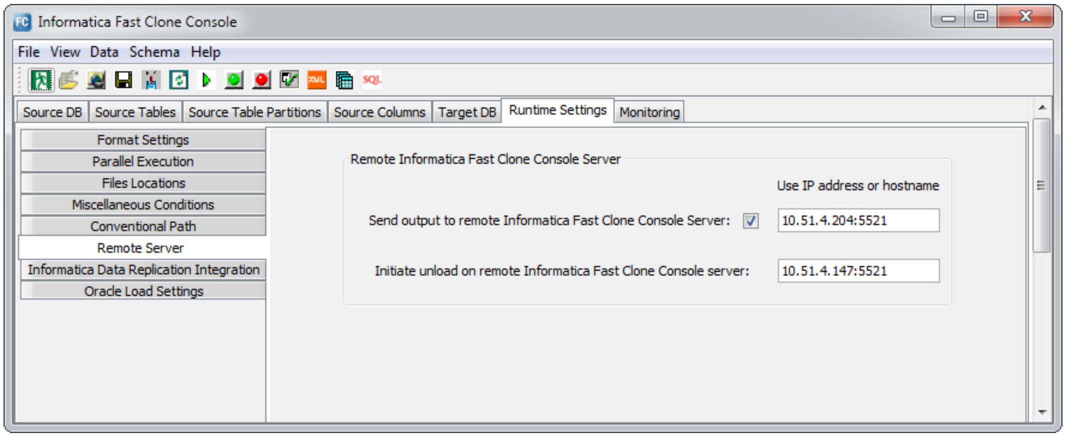
| Field | Description |
|---|---|
| Fetch LOBs as single chunk | Select this option to unload a single chunk of LOB data that has the size specified in the **Fetch LOBs as single chunk, in KB** field. By default, this option is not selected and Fast Clone fetches all LOB data in multiple chunks. |
| Fetch chunk size for LOBs, in KB | To unload all of the LOB data in chunks of a consistent size, enter the size, in kilobytes, to use for each chunk. Default is 128 KB. |
| Fetch LOBs as a single chunk, in KB | If you selected **Fetch LOBs as a single chunk**, enter the maximum size, in kilobytes, of the single chunk. |
| Fetch array size | The number of rows in an array that Fast Clone unloads from the database during each round trip for conventional path unload processing. Default is 500. |

## Configuring Fast Clone Server Tasks

In a distributed Fast Clone environment, you can configure a Fast Clone Server to accept output files from another Fast Clone system or to run unload jobs that are initiated on a standalone Fast Clone Console system.

1. Click the **Runtime Settings** > **Remote Server** view.

   The following image shows this view:



Remote Informatica Fast Clone Console Server

Use IP address or hostname

Send output to remote Informatica Fast Clone Console Server: ☑ 10.51.4.204:5521

Initiate unload on remote Informatica Fast Clone Console server: 10.51.4.147:5521

2. To configure a Fast Clone Server on a target system to accept output files from another Fast Clone system, complete the following steps:

   a. Ensure that the Fast Clone Server is running on the target system.

   b. Select the **Send output to remote Informatica Fast Clone Server** option.

   c. In the adjacent text box, enter the host name or IP address of the Fast Clone Server system followed by the port number. Use the format *host*:*port*.

3. To configure a Fast Clone Server on the source system to accept unload requests from a remote Fast Clone Console system and run unload jobs on the source system, complete the following steps:
   a. Ensure that the Fast Clone Server is running on the source system.
   b. Enter connection information for the Fast Clone Server system in the **Initiate unload on remote Fast Clone Server** field. Enter the host name or IP address followed by the port number. Use the format *host*:*port*.

### Related Topics:

- "Transmitting the Output Files to the Target" on page 88

## Configuring Integration with Informatica Data Replication

If you have the Informatica Data Replication product for change data replication, you can enable Fast Clone to use the Data Replication configuration file to perform an initial load of a data replication target. In the Fast Clone Console, enable integration with Data Replication and specify the Data Replication configuration file.

The Data Replication configuration file can be in XML or SQLite format. Ensure that the mapped source columns in the Data Replication configuration file have datatypes that Fast Clone supports for unload processing.

To integrate Fast Clone with Data Replication, the Fast Clone Console requires source and target connection information. Fast Clone uses this information to generate a minimal .ini configuration file for the data-cloning job. When you run the Fast Clone cloning job, the Fast Clone executable reads the connection information from the generated cloning configuration file and all of the other settings from the Data Replication configuration file.

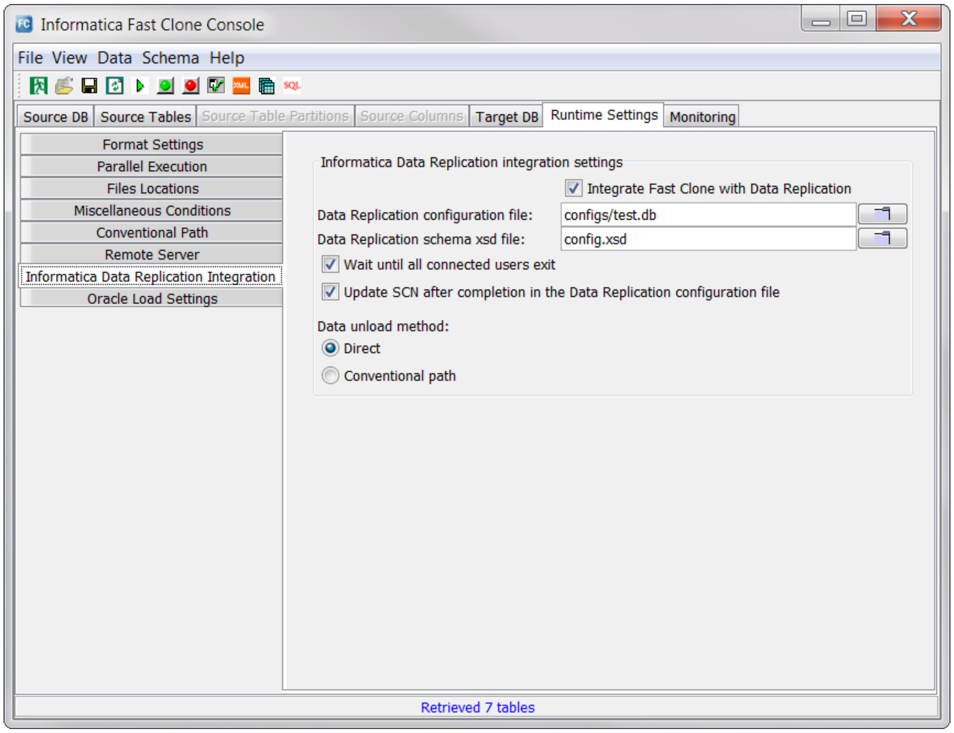
After Fast Clone loads data to a target, it updates the SCN value in the Data Replication configuration file. When you use this configuration file for change data replication, Data Replication uses the updated SCN to replicate the changes that occur after the initial load of the target.

**Note:** Fast Clone does not support Data Replication configurations that include virtual columns.

1. On the **Source DB** tab, enter connection information for the source database and click **Connect** to connect to the source.
2. On the **Target DB** tab, enter connection information for the target database.

   If you want to validate the target connection information, connect to the target database. However, Fast Clone does not require that you connect to the target at this time.
3. Click the **Runtime Settings** tab > **Informatica Data Replication Integration** view.

The following image shows this view:



Informatica Data Replication integration settings

☑ Integrate Fast Clone with Data Replication

Data Replication configuration file: configs/test.db

Data Replication schema xsd file: config.xsd

☑ Wait until all connected users exit

☑ Update SCN after completion in the Data Replication configuration file

Data unload method:

● Direct

○ Conventional path

4. To integrate Fast Clone with Data Replication, complete the fields described in the following table:

| Field | Description | Default |
|---|---|---|
| Integrate Fast Clone with Data Replication | Select this option to enable Fast Clone to use a Data Replication configuration file for a data-cloning job. | Not selected |
| Data Replication configuration file | The path to the Data Replication configuration file relative to the DBSYNC_HOME directory. This file can be in XML or SQLite format. You can click the **Browse** button to the right of the field to browse to the configuration file.<br>Ensure that the DBSYNC_HOME environment variable is defined.<br>If you use a Data Replication configuration in SQLite format, do not copy or move the configuration files from the default *DataReplication_installation*/configs directory to another location. | No default |
| Data Replication schema xsd file | The path to the schema file, *config*.xsd, for the Data Replication configuration file relative to the DBSYNC_HOME directory. The XML structure of the Data Replication configuration file must be consistent with the schema in this schema file. | No default |
| Wait until all connected users exit | Indicates whether Fast Clone waits for all of the user sessions on the mapped source tables to end before beginning unload processing. If you clear this option, Fast Clone ends the active user sessions on the mapped tables during unload processing. | Selected |

| Field | Description | Default |
|---|---|---|
| Update SCN after completion in the Data Replication configuration file | Indicates whether Fast Clone updates the SCN value in the Data Replication configuration file after loading data to the target to set a start point for change data replication. When you run a replication job with the configuration, Data Replication replicates only the changes that have SCN values larger than the one that Fast Clone set in the configuration file. | Selected |
| Data unload method | The unload method that Fast Clone uses to unload data with the specified Data Replication configuration file. Options are:<br>- **Direct**. Use the direct path unload method.<br>- **Conventional path**. Use the conventional path unload method. | Direct |

# Configuring Load Settings for Targets

On the **Runtime Settings** tab > **Load Settings** view, you can configure load settings that are specific to the target type.

The **Load Settings** view is displayed for most target types. It appears after you define the target on the **Target DB** tab.

## Greenplum Load Settings

On the **Runtime Settings** tab > **Greenplum/Bizgres Load Settings** view, configure load settings for Greenplum targets.

The following table describes the fields in this view:

| Field | Description | Default |
|---|---|---|
| Greenplum loader home | When you use output files or pipes to load data, enter the path to the Greenplum loader root directory that Fast Clone uses to locate the gpload utility and generate valid load scripts. | No default |
| Enable GreenPlum gpfdist Direct Data Stream | Indicates whether to use DataStreamer to load data. Select this option to stream data directly to a Greenplum target with the gpfdist utility. | Not selected |
| Skip writing the output data into files | Indicates whether to skip writing unloaded data to output files or pipes.<br>If you use DataStreamer, the Fast Clone Console selects this option. If you do not use DataStreamer, you can select this option to skip writing the unloaded data to files or pipes for troubleshooting purposes. | Not selected |
| Max errors | If you use DataStreamer, enter the maximum number of errors that can occur when streaming data to a Greenplum target. If a number of errors exceeds this maximum value, DataStreamer ends with an error. | 100,000 |
| Remotely accessed IP or hostname | If you use DataStreamer, enter the host name or IP address of the system where you run the Fast Clone instance that starts the gpfdist utility. | Local host name |

| Field | Description | Default |
|---|---|---|
| Listen on specific interface | If you use DataStreamer, enter the IP addresses on which the gpfdist utility listens for incoming requests. Use 0.0.0.0 to listen to all request interfaces. | No default |
| Listening port | If you use DataStreamer, enter the listener port for the gpfdist utility. A Greenplum target uses this port to submit requests to the gpfdist utility. | 8000 |

## Microsoft SQL Server Load Settings

On the **Runtime Settings** > **SQLServer Load Settings** view, you can configure load settings for or Microsoft SQL Server targets.

The following table describes the fields in this view:

| Field | Description | Default |
|---|---|---|
| Batch size | Specifies the number of rows in a batch that is passed to the osql utility for loading data to the target. The osql utility copies each batch to the target server as one transaction.<br>Fast Clone uses this value as the BATCHSIZE option value in the BULK INSERT statement for the osql utility. | 10000 |
| Max errors | Specifies the maximum number of syntax errors that can appear in the output data files before the osql utility cancels the BULK INSERT operation. The osql utility ignores all of the rows that it cannot parse and counts an error for each row.<br>Fast Clone uses this value as the MAXERRORS option value in the BULK INSERT statement for the osql utility. | 10000 |
| Column collations | The name of an SQL Server collation that Fast Clone uses to store character and Unicode data in output data files. Fast Clone uses this value to correctly specify a column collation in the format file that maps the fields in the data file to table columns. Fast Clone uses the specified collation for all of the table columns.<br>Fast Clone uses this value as the FORMATFILE option value in the BULK INSERT statement for the osql utility. | SQL_Latin1_General_Cp437_BIN |

## Netezza Load Settings

On the **Runtime Settings** > **Netezza Load Settings** view, you can configure load settings for Netezza targets.

The following table describes the fields in this view:

| Field | Description | Parameter Name | Default |
|---|---|---|---|
| Enable Netezza Direct Data Stream | Indicates whether to use DataStreamer to load data to a Netezza target. Select this option to stream data directly to the Netezza target by using the Netezza external tables that are represented by the named pipes. | netezza_direct_data_stream | Not selected |
| Pipe directory path | If you use DataStreamer to load data to a Netezza target, specifies the directory in which Fast Clone creates named pipes that represent Netezza external tables.<br>On Windows, DataStreamer ignores this parameter because the named pipes are created in the named pipe directory that is mounted under the special path \\.\pipe\. | netezza_pipe_directory | The output directory |
| Error log directory name | If you use DataStreamer to load data to a Netezza target, specifies the directory in which DataStreamer creates the following Netezza log files:<br>- *table_name*.*schema_name*.nzlog. This log file includes load statistics and diagnostic messages that the Netezza ODBC driver issues when loading data to the target tables from the corresponding external tables.<br>- *table_name*.*schema_name*.nzbad. This log file includes the external table rows that DataStreamer could not load to the corresponding target table. | netezza_error_log_directory | The output directory |

| Field | Description | Parameter Name | Default |
|---|---|---|---|
| Quoted value | If you use DataStreamer to load data to a Netezza target, specifies the value of the QuotedValue option for the Netezza external tables. DataStreamer requires this option to build correct Insert statements when loading data to the target tables from the corresponding external tables. Options are:<br>- **single**. Indicates that the data values in the external tables are enclosed with single quotation marks.<br>- **double**. Indicates that the data values in the external tables are enclosed with double quotation marks.<br>- **no**. Indicates that the data values in the external tables are not quoted.<br>Ensure that the quotation character that you specify for the load operation matches the quotation character that you specify for the unload operation. For unload jobs, Fast Clone uses the quotation character that you specify in the **Enclosed by** filed on the **Runtime Settings** tab > **Format Settings** view. | netezza_quoted_ value | no |
| Control character and carriage returns | If you use DataStreamer to load data to a Netezza target, specifies the value of the EscapeChar option for the Netezza external tables. This option indicates whether the control characters, such as delimiter and backslash characters, are escaped in the data fields of the external tables. The Netezza external tables use a backslash as an escape character. | netezza_control_ chars_enabled | Selected |

## Oracle Load Setting

On the **Runtime Settings** > **Oracle Load Settings** view, you can configure a load setting for Oracle targets.

The following table describes the field in this view:

| Field | Description | Default |
|---|---|---|
| Load data to same partition as source | Select this option to load data from source partitions to Oracle target partitions of the same name. This information is used by the SQL*Loader utility. | Not selected |

## Teradata Load Settings

On the **Runtime Settings** tab > **Teradata Load Settings** view, you can configure data load settings for Teradata targets.

The following table describes the fields in this view:

| Field | Description | Default |
|---|---|---|
| Teradata Load Utilities in Indicator mode | If you selected the **Binary length prefix** option, indicates whether to set the indicator bits in the binary output for column values that contain NULL data. | Selected |
| Use Teradata Parallel Transporter (TPT) | Indicates whether Fast Clone uses Teradata Parallel Transporter to load the source data to the target database. Select this option, if you use Teradata Direct DataStream. | Selected |
| Loader to be used | The Teradata loader type that Fast Clone uses to import unloaded data into Teradata. Fast Clone supports the Teradata FastLoad utility, MultiLoad utility, and Parallel Data Pump. Fast Clone generates the appropriate control input for the utility type.<br>**Note:** Parallel Data Pump is available only if you select the **Use Teradata Parallel Transporter** option. | Teradata FastLoad |
| Teradata Load Sessions | The number of sessions that Teradata FastLoad or MultiLoad opens to load the source data to each target table. | 2 |
| Binary length prefix | Indicates whether to unload source data in the binary format. Each record in the binary format includes a 2-byte field that indicates the total length of the record, optional bytes for null values, and the data. | Not selected |
| Binary length prefix endianness | If you selected the **Binary length prefix** option, specify the endianness of the binary output. This parameter determines the format of the integer that specifies the length of records and fields in the binary files. Options are: LITTLE ENDIAN and BIG ENDIAN. | LITTLE ENDIAN |
| Use Teradata Direct DataStream | Select this option to have DataStreamer call TPT to load the source data to the target database directly. DataStreamer uses the loader that is specified in the **Loader to be used** list. | Not selected |
| Drop and rebuild tables in Teradata prior to load | Select this option if you want DataStreamer to drop and re-create the Teradata tables before loading the source data into the tables. | Not selected |
| Skip writing the output data into files | Select this option to skip writing the unloaded data to output files or a pipe.<br>If you selected the option to use DataStreamer, this option is also selected.<br>If you do not use DataStreamer, you can select this option for troubleshooting purposes. | Not selected |
| Max errors | If you use DataStreamer, the maximum number of errors that can occur when streaming data to a Teradata target. If a number of errors exceeds this maximum, DataStreamer ends with an error. | 100,000 |

## Vertica Load Settings

on the **Runtime Settings** tab > **Vertica Load Settings** view, you can configure data load settings for Vertica targets.

The following table describes the fields in this view:

| Field | Description | Default |
|---|---|---|
| Loader to be used | The name of the Vertica command that is used to load the unloaded data to Vertica. Fast Clone supports the following Vertica loader commands:<br>- ISQL COPY. Loads data from the data files that are located on the target database server.<br>- ISQL LCOPY. Loads data from the data files that are located on the system where you run the target load utility. | ISQL COPY |
| Binary length prefix | Controls whether the data in Fast Clone data files is loaded into Vertica targets in NATIVE VARCHAR or text format.<br>If you select this option, Fast Clone adds the NATIVE VARCHAR keyword to the COPY or LCOPY command in the output SQL load script. The database server converts CHAR or VARCHAR fields in the data files to Vertica datatypes when loading data.<br>By default, this option is not selected and Fast Clone loads data in text format with delimiters. | Not selected |
| Binary length prefix endianness | The endianness of the data in the data files. Valid values are LITTLE ENDIAN and BIG ENDIAN.<br>This field is available only when the **Binary length prefix** option is selected. | LITTLE ENDIAN |

The Vertica ISQL COPY command has a 65 KB character limit on the length of NATIVE and NATIVE VARCHAR data. If Vertica encounters a value larger than 65 KB, it rejects the row, even if the ENFORCELENGTH option for the COPY or LCOPY command is disabled.

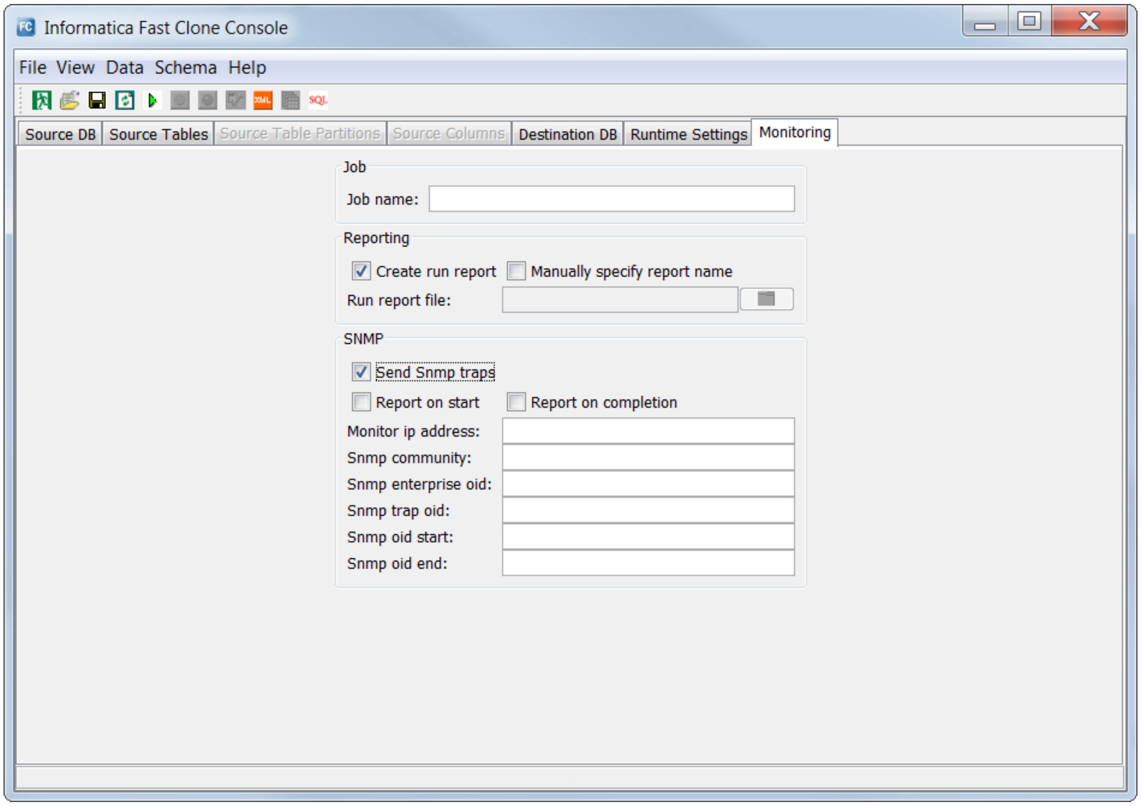
# Configuring Reporting and SNMP Traps

Fast Clone can optionally generate reports for unload jobs in XML and use the Simple Network Management Protocol (SNMP) to send asynchronous notifications, or traps, to an SNMP manager.

The reports specify the number of rows that were unloaded and the number of rows that failed to be unloaded, if any, for each source table. Fast Clone generates the XML report files in the output directory at the completion of unload processing.

If you configure Fast Clone to send SNMP traps, you can monitor the SNMP traps by using PowerSNMP Free Manager or a similar product.

1. Click the **Monitoring** tab. The following image shows this view:



Job

Job name:

Reporting

☑ Create run report ☐ Manually specify report name

Run report file:

SNMP

☑ Send Snmp traps

☐ Report on start ☐ Report on completion

Monitor ip address:

Snmp community:

Snmp enterprise oid:

Snmp trap oid:

Snmp oid start:

Snmp oid end:

2. In the **Job name** field, enter a name that will be used as the report file name base and included in the SNMP trap text.

   You can override this name for reports.

3. To configure Fast Clone to generate an unload run report, complete the following steps:

   a. Select the **Create run report** option.

   b. To override the report name based on the **Job name** value, select the **Manually specify report name** option. Then enter a report name.

4. To configure Fast Clone to send SNMP traps to an SNMP manager, complete the following steps:

   a. Select the **Send Snmp traps** option.

b. Complete the fields described in the following table:

| Field | Description |
|---|---|
| Report on start | Select this option to send SNMP traps to a SNMP manager when the unload job starts.<br>By default, this option is not selected. |
| Report on completion | Select this option to send SNMP traps to a SNMP manager when the unload job starts.<br>By default, this option is not selected. |
| Monitor IP address | Enter the IP address of the SNMP manager.<br>No default is provided. |
| Snmp community | Enter a name to use for the Fast Clone SNMP community.<br>The default name is wisdomforce. |
| Snmp enterprise oid | Enter the Fast Clone enterprise object identifier (OID) in SNMP.<br>The default value is 1.3.6.1.2.1.1.1.2.0.1. |
| Snmp trap oid | Enter the Fast Clone SNMP trap OID.<br>The default value is 1.3.6.1.2.1.1.1.2.0.2. |
| Snmp trap begin oid | If you selected **Report on start**, enter the Fast Clone SNMP trap detail OID for traps that are sent when the unload job starts.<br>The default value is 1.3.6.1.2.1.1.1.2.0.3. |
| Snmp trap end oid | If you selected **Report on completion**, enter the Fast Clone SNMP trap detail OID for traps that are sent when the unload job completes.<br>The default value is 1.3.6.1.2.1.1.1.2.0.4. |

# Saving a Cloning Configuration File Locally

After you finish configuring the data-cloning job, save the configuration file.

1. To save the Fast Clone configuration file on the local computer, perform one of the following actions:
   - On the menu bar, click **File** > **Save**.
   - On the toolbar, click the **Generate and save configuration file** button. The Fast Clone Console generates the configuration file in the current working directory.

     **Note:** To change the directory, specify an **Output directory** value on the **Runtime Settings** tab. For more information, see “Configuring Runtime Settings” on page 45.
2. If you selected source columns that are encrypted with Oracle Transparent Data Encyrption (TDE) when creating the configuration file, open the Oracle wallet:

   a. Click **Specify PKCS#12 wallet** and open the wallet file.

   b. Enter a valid password in the **PKCS#12 password** field.

   c. Click **Open**.

   The Fast Clone Console opens the wallet and writes master key IDs and values to the configuration file.

CHAPTER 4

# Unloading Data from the Source Database

This chapter includes the following topics:

## Unload Overview

Fast Clone can unload data from regular tables, compressed tables, partitioned tables, views, materialized views, index-organized tables, and encrypted tables and tablespaces. For all target types, you can configure Fast Clone to unload data into output files or pipes.

For Greenplum, Netezza, and Teradata targets, if you use the direct path unload method and the DataStreamer add-on component, you can configure Fast Clone to stream data to the target database. Streaming data avoids the use of intermediate storage for output data files.

Fast Clone provides two methods of unloading data from an Oracle source: direct path unload and conventional path unload. The direct path unload method is much faster because it reads Oracle data files directly. It also works with the DataStreamer component. The conventional path unload method uses the standard Oracle Call Level Interface (OCI) to read data and metadata. With the conventional path method, you can unload data over the network. Also, Fast Clone can read data from additional types of Oracle objects, such as views, cluster tables, and index-organized tables.

You can run a data unload job from the Fast Clone Console or command line. If you use the Fast Clone Console, the Console runs the unload script that is specified in the runtime settings to start the Fast Clone executable. By default, the unload script name is unload.cmd on Windows and unload.sh on Linux and UNIX. The Fast Clone executable uses the cloning configuration file to unload data. If you use the command line, you can start the Fast Clone executable directly from the command line or run the unload script. When you enter the Fast Clone executable or unload script at the command line, you can specify the unload parameters either by using the config parameter to point to the cloning configuration file or by entering the parameters at the command line.

# Character Set of the Unloaded Data

For the direct path unload method, Fast Clone does not convert source character data from the source character set to another character set. For the conventional path unload method, the NLS_LANG environment variable determines the character set of the unloaded data.

For the direct path unload method, Fast Clone unloads data in the character set that Oracle database uses to store the source data. For example, if Oracle stores the data in UTF-8 encoding, Fast Clone unloads data in UTF-8. To view the character set of the source database, use the following command:

```
SELECT parameter,value FROM NLS_DATABASE_PARAMETERS
where parameter in ('NLS_CHARACTERSET','NLS_NCHAR_CHARACTERSET');
```

For the conventional path unload method, you can define the NLS_LANG environment variable on the system where you run unload jobs to unload data with the character encoding of the target database. Use the following syntax, depending on the system type:

- On Linux, run the following commands from the Linux console:

  ```
  NLS_LANG=language_territory.charset
  export NLS_LANG
  ```

- On UNIX, run the following command from the UNIX console:

  ```
  setenv NLS_LANG language_territory.charset
  ```

- On Windows, run the following command from the command prompt:

  ```
  SET NLS_LANG=language_territory.charset
  ```

When you run Fast Clone unload processing, Oracle uses the NLS_LANG environment variable to convert the character set and provide the source data in the specified encoding.

For both direct and conventional path unload methods, you can configure Fast Clone to convert character data in CLOB, NCLOB, and NCHAR columns from UTF-16 to UTF-8 encoding. Depending on the source datatypes, set the following parameters to true in the cloning configuration file to convert the source character data to UTF-8 encoding:

- convert_utf16_clobs_to_utf8
- convert_utf16_nclobs_to_utf8
- convert_utf16_nchars_to_utf8

# Unloading Data from Views

You can configure Fast Clone to unload data from Oracle views by using the conventional path unload method.

You cannot select views for a data unload job in the Fast Clone Console because source views are not displayed on the **Source Tables** tab. To unload data from views, use one of the following methods:

- Open the configuration file in a text editor and manually add views to the [SOURCE_INDIRECT_TABLES] section. Use the following syntax:

  ```
  TABLEid=view_name
  ```

  To unload specific columns from a view, add a statement for each view on a separate row in the [SOURCE_INDIRECT_TABLES_COLUMNS_LIST] section. In each view statement, list the columns to unload, using a comma (,) as the separator. Use the following format:

  ```
  TABLEid_COLUMN_LIST=column_name_1,column_name_2,...
  ```

  For example, the following statements unload data from three columns of the DEMO_VIEW view:

  ```
  [SOURCE_INDIRECT_TABLES]
  TABLE1=DEMO_VIEW

  [SOURCE_INDIRECT_TABLES_COLUMNS_LIST]
  TABLE1_COLUMN_LIST=COLUMN1,COLUMN2,COLUMN3
  ```

- Define an SQL query that selects data from the view. In the query statement, use the full view name, including the schema name. You can define an SQL query for a configuration file in the Fast Clone Console.

**Note:** To unload data from Oracle materialized views, select the materialized views on the **Source Tables** tab in the Fast Clone Console. You can use the direct path unload method because Oracle stores materialized view data in the data files.

### Related Topics:

- "Defining SQL Queries for Unload Processing" on page 34

# Unloading Data from TDE-Encrypted Columns and Tablespaces

Fast Clone supports Oracle sources that use Oracle Transparent Data Encryption (TDE) encryption of columns and tablespaces.

**Note:** Fast Clone supports encrypted columns for Oracle 10.2.0.0 and later and encrypted tablespaces for Oracle 11.2.0.2 and later.

Fast Clone supports TDE with all Oracle encryption algorithms, including 3DES and AES with 128-bit, 192-bit, and 256-bit key lengths. Fast Clone also supports TDE with HMAC integrity checking and SALT enhanced security.

To unload data from encrypted columns or tablespaces, you must open the Oracle encryption wallet in one of the following ways:

- If you use the direct path unload method, Fast Clone uses the master key IDs and values in the [MKEY_ID] and [MKEY_VALUES] sections of the cloning configuration file to unload data from encrypted tables. The Fast Clone Console prompts you to open the Oracle wallet when you save the configuration file.

- If you use the conventional path unload method, you must open the wallet before you start the unload job to grant access to the encrypted columns or tablespaces. Use one of the following commands, depending on your Oracle version, to open the wallet in Oracle:

  ```
  -- Oracle 10g version
  ALTER SYSTEM SET ENCRYPTION WALLET OPEN AUTHENTICATED BY "password";
  -- Oracle 11g version
  ALTER SYSTEM SET ENCRYPTION WALLET OPEN IDENTIFIED BY "password";
  ```

# Running Data Unload Jobs

You can run a data unload job from the Fast Clone Console or the command line.

- To start a data unload job from the Fast Clone Console, perform one of the following actions:

  - On the menu bar, click **Data** > **Unload data with Informatica Fast Clone Console**.

  - On the toolbar, click the **Start Informatica Fast Clone Console as specified in Runtime settings** button.

  The Fast Clone Console runs the unload script that you specified in the **Informatica Fast Clone Console start script** field on the **Runtime Settings** tab > **Files Locations** view. By default, the Fast Clone Console uses the unload.cmd script on Windows and the unload.sh script on Linux and UNIX to start the Fast Clone executable. You can edit the default script or create another script based on the default script to set environment variables or perform particular configuration tasks.

  If you run a data unload job with the default script, Fast Clone uses the configuration file that you specified in the **Config file name** field on the **Runtime Settings** tab > **Files Locations** view to unload data.

- To run a data unload job from the command line, you can start the Fast Clone executable directly or use the unload scripts. To start the Fast Clone executable directly, use the following syntax:

  - On Windows:

  ```
  Fastreader.exe config=configuration_file
  ```

  - On Linux and UNIX:

  ```
  Fastreader config=configuration_file
  ```

  To start the Fast Clone executable with the default unload scripts, use the following syntax:

  - On Windows:

  ```
  unload.cmd configuration_file
  ```

  - On Linux and UNIX: .

  ```
  ./unload.sh config=configuration_file
  ```

  Instead of using the cloning configuration file, you can enter unload parameters directly at the command line. For more information, see “Command Line Parameters” on page 95.

# Unloading Data to Named Pipes

You can configure Fast Clone on Linux and UNIX to unload data to named pipes instead of to output files. Use pipes to avoid consuming storage for large output files.

**Note:** Fast Clone does not support unloading data to named pipes on Windows.

Fast Clone uses a named pipe that has fixed size of approximately 6 MB for each source table from which data is unloaded. By default, the pipe name is based on the name of the source table or table partition. After Fast Clone writes a chunk of data to the pipe, the target load utility reads the data from the pipe and loads it to the target. Fast Clone then writes the next chunk of data to the pipe. After the load utility loads the last chunk of data from the pipe, Fast Clone closes the pipe. If the load utility does not read the data from the pipe, Fast Clone unload processing remains pending.

To load data from named pipes to targets, you must generate the load script before you run the unload job. By default, Fast Clone generates the load script after it completes unloading data from the source to output files. However, to stream data through pipes to the target database, you must generate the load script, start unload processing, and then run the load script while unload processing is in progress. To generate the load script and unload data to pipes, you must run the Fast Clone twice:

- The first time you run Fast Clone, select the **Generate loader input only, do not unload data** runtime option. Fast Clone generates the load script in the output directory but does not unload data.
- The second time you run Fast Clone, clear the **Generate loader input only, do not unload data** runtime option and then run Fast Clone either directly or with the unload script. Fast Clone then unloads data to the pipes but does not generate another load script. While unload processing is occurring, you can start the load script from the first Fast Clone run to begin loading data to the target.

Before you begin unload processing, either manually create a pipe for each source table in the output directory or configure Fast Clone to create the pipes. If you configure Fast Clone to create the pipes, monitor pipe creation in the output directory so that you do not start the load jobs until after the pipes are created. Alternatively, you can create a script to delay load processing until after Fast Clone creates all of the pipes.

## Scenarios That Cause Cloning Processes to Remain Pending

When you unload data to named pipes, the Fast Clone cloning process might remain in a pending state in certain scenarios.

Typically, this situation occurs in the following scenarios:

- You configure Fast Clone to create a named pipe for a table prior to unloading data from the table. However, a named pipe was previously created for this table in the output directory and the load utility reads data from this pipe. When you start the cloning job, Fast Clone creates another named pipe in the output directory. The target load utility reads data from the first pipe, and Fast Clone writes data to the second pipe. Consequently, both the unload and load processes remain in a pending state.
- You configure Fast Clone to create a named pipe for a table prior to unloading data from the table. You run the data unload job and then run the load job. However, Fast Clone did not create the named pipe by the time you started the load job. The load job fails because it cannot read data from the named pipe. Consequently, the unload process remains in a pending state.

## Unloading Data to Named Pipes Created by Fast Clone

You can configure Fast Clone to create a named pipe in the output directory for each source table before unloading data from the tables. The target load utility reads data from the pipes and loads it to the target.

When unloading data to named pipes, Fast Clone performs a checkpoint for the source database and then creates the pipes for the tables to unload. Checkpoint processing can take up to 20 seconds.

Before you run the load job for a table, ensure that Fast Clone created the pipe for the table. You can create a script to delay the load job until after the pipe is created.

**Note:** If you start a load job for a table before Fast Clone creates the pipe for the table, the load process fails because the target load utility cannot read data from the pipe. Also, the unload process remains in a pending state.

1. Create a cloning configuration file.

   Specify connection information for the source and target databases, the source tables to unload, the load script base name, and other settings, as needed.

2. To generate a load script for the target utility, complete the following steps:

   a. Configure Fast Clone to generate the load script in one of the following ways:

      - In the Fast Clone Console, select the **Generate loader input only, do not unload data** option on the **Runtime Settings** tab > **Files Locations** view.
      - At the command line, enter create_load_input_only=true.
      - In the configuration file, enter the create_load_input_only=true parameter.

   b. Save the cloning configuration file.

   c. Run the Fast Clone with the updated cloning configuration file.

      Fast Clone generates a .cmd load script on Windows or a .sh load script on Linux and UNIX in the output directory.

3. Configure Fast Clone to unload data without overwriting the generated load script in one of the following ways:

   - In the Fast Clone Console, clear the **Generate loader input only, do not unload data** option and select the **Unload data only, do not generate loader input** option on the **Runtime Settings** tab > **Files Locations** view.
   - At the command line, enter the create_load_input_only=false and unload_data_only=true parameters.
   - In the configuration file, enter the create_load_input_only=false and unload_data_only=true parameters.

4. Configure Fast Clone to create named pipes in the output directory in one of the following ways:

   - In the Fast Clone Console, select the **Create output as pipes instead of files** option on the **Runtime Settings** tab > **Miscellaneous Conditions** view.
   - At the command line, enter the create_output_pipe_instead_of_file=true parameter.
   - In the configuration file, set the create_output_pipe_instead_of_file parameter to true.

5. Run the data unload job.

   Fast Clone performs a checkpoint for the source database and creates a named pipe for each source table. Fast Clone then unloads the source data to the pipes.

6. After Fast Clone creates the pipes, run the load script that was generated in Step 2.

   To delay the load job until after the pipes are created, you can use another script that monitors creation of the pipes and invokes the load script. For example, use the following Perl script on Linux:

   ```
   perl -e "while (not(-p '/output_directory/table_name.dat')) {sleep 1;}";
   /output_directory/load_script.sh
   ```

   The target load utility reads data from the named pipes and loads it to the target.

## Unloading Data to Manually Created Named Pipes

If you want to unload data to pipes, you can create the named pipes manually before running the cloning job instead of configuring Fast Clone to generate the pipes.

When you run the cloning job later, Fast Clone unloads data to the named pipes that you manually created. The target load utility reads data from the pipes and loads it to the target.

The advantage of manually creating pipes, in comparison to Fast Clone generation of the pipes, is that you do not need to monitor when pipes are generated or create a script that delays the load job until after Fast Clone creates the pipes.

1. Create a cloning configuration file.

   Specify connection information for the source and target databases, the source tables to unload, the load script base name, and other settings, as needed.

2. To generate a load script for the target utility, complete the following steps:

   a. Configure Fast Clone to generate the load script in one of the following ways:

      - In the Fast Clone Console, select the **Generate loader input only, do not unload data** option on the **Runtime Settings** tab > **Files Locations** view.
      - At the command line, enter the create_load_input_only=true parameter.
      - In the configuration file, enter the create_load_input_only=true parameter.

   b. Save the cloning configuration file.

   c. Run the Fast Clone with the updated cloning configuration file.

      Fast Clone generates a .cmd load script on Windows or a .sh load script on Linux and UNIX in the output directory.

3. Configure Fast Clone to unload data without overwriting the generated load script in one of the following ways:

   - In the Fast Clone Console, clear the **Generate loader input only, do not unload data** option and select the **Unload data only, do not generate loader input** option on the **Runtime Settings** tab > **Files Locations** view.
   - At the command line, enter the create_load_input_only=false and unload_data_only=true parameters.
   - In the configuration file, enter the create_load_input_only=false and unload_data_only=true parameters.

4. Create a named pipe for each table that you selected for unload processing in the output directory.

   For example, issue the following commands on Linux and UNIX to create a named pipe for a table:

   ```
   rm /output_directory/table_name.dat
   mknod /output_directory/table_name.dat p
   ```

5. Configure Fast Clone to use the pipes that you manually created in one of the following ways:

   - In the Fast Clone Console, verify that the **Create output as pipes instead of files** option on the **Runtime Settings** tab > **Miscellaneous Conditions** view is cleared.
   - In the configuration file, enter the create_output_pipe_instead_of_file= false parameter.

6. Run the data unload job.

   Fast Clone writes the unloaded data to the named pipes that you manually created in the output directory.

7. Run the load script that was generated in Step 2.

   The target load utility reads data from the named pipes and loads it to the target.

# Unloading Data to Files of Fixed-Length Record Format

Fast Clone can unload data to output files that use fixed-length record format. To use output files of fixed-length record format, you must enable padding for the fields.

In data files of fixed-length record format, each field starts and ends at the same positions in every record. Fast Clone records these positions for all of the columns in the output control files. If the number of bytes of data in a field is less than the field length, Fast Clone can pad the field on the right with space characters. Use this format to unload data to files that use blank column and row separators.

To enable padding for all columns in fixed-length record format data files, use one of the following methods:

- In the Fast Clone Console, on the **Runtime Settings** > **Format Settings** view, select the **Pad with Blanks (fixed length)** option.
- Edit the configuration file to set the right_pad_columns parameter to true.
- From the command line, set the RIGHT_PAD_COLUMNS parameter to true.

If you unload data to a fixed-length format file and configured Fast Clone to unload data from columns that have BLOB, CLOB, NCLOB, LONG, and LONG RAW datatypes to a separate file, Fast Clone unloads all of the column data. Otherwise, the maximum amount of data that Fast Clone unloads to a fixed-length format file is 8 KB.

# Generating the Data Files in XML Format

If you run the unload job from the Fast Clone Console, you can optionally generate the data files in XML format based on the regular ASCII data files. You can then use the XML files with third-party software.

1. Open the *FastClone_installation*/uiconf/default.cfg file in a text editor and set the generate_xml_metadata parameter to true.

   This parameter indicates whether Fast Clone creates metadata files in the _xml subdirectory of the output directory when you run a data unload job.
2. Restart the Fast Clone Console for this setting to take effect.
3. Open the configuration file for which to run the data unload job and save it again in the Fast Clone Console.
4. Run the data unload job.

   Fast Clone generates the *table_name*.columns file for each unloaded table in the _xml subdirectory of the output directory. These files include table metadata for generating XML data files.
5. Generate the XML data files based on the generated metadata files and the regular data files in one of the following ways:

   - From the command prompt on Windows, run the xmlconvert.cmd script. From the command line on Linux or UNIX, run the xmlconvert.sh script.
   - From the Fast Clone Console, click **Data** > **Convert to XML** on the menu bar, or click the **Convert CSV output into XML format** button on the toolbar.

   Fast Clone generates the XML data files in the _xml subdirectory for each unloaded table.

If the operation ends with an error, view the informaticaFastClone_mngr.log file that Fast Clone generates in the installation directory to diagnose the problem. The following table describes return codes for the script that generates the XML data files:

| Error Code | Description |
|---|---|
| 0 | Fast Clone successfully generated the XML data files. |
| 1 | Fast Clone encountered an internal error when generating the XML data files. |
| 2 | Fast Clone did not find the output directory. |
| 3 | Fast Clone did not find the output data files for which to generate XML data files. |
| 4 | Fast Clone did not find the _xml subdirectory in the output directory. |
| 5 | Fast Clone could not create an XML file. |
| 6 | Fast Clone did not find the metadata files in the _xml subdirectory. |
| 7 | Fast Clone could not open the configuration file. |

CHAPTER 5

# Loading Data to a Target

This chapter includes the following topics:

## Load Overview

For all target types, you can load data from output files or pipes to the target database by using the native target load utility. If you use the optional DataStreamer add-on component For Greenplum, Netezza, and Teradata targets, Fast Clone can stream data directly to the target by using the appropriate Greenplum and Teradata load tool or Netezza external tables.

The Fast Clone cloning configuration file contains the target connection information and load settings that are required to load data.

### Loading Data from Output Files or Pipes

If you unload data to output files or pipes, Fast Clone generates a load script that you run to load data to the target. In the Fast Clone Console, you specify the load script name on the **Runtime Settings** tab > **File Locations** view. Fast Clone adds the extension .cmd (on Windows) or .sh (on Linux or Windows) to the script name. The generated load script is tailored for the target load utility or command-line tool that Fast Clone uses for the target type. The following table identifies the load utilities that Fast Clone uses:

| Target Type | Load Utility |
|---|---|
| Greenplum | gpload or gpfdist utility |
| Microsoft SQL Server | osql utility |
| Netezza | nzload utility |

| Target Type | Load Utility |
|---|---|
| Oracle | SQL*Loader (sqlldr) utility |
| Teradata | FastLoad or MultiLoad utility |
| Vertica | SQL COPY command |

To load data from output files or pipes to a target, perform the following tasks:

1. Create a configuration file.
2. Run the data unload job with the configuration file.
3. Generate the target tables.
4. Optional. Validate the target schema.
5. Optional. Copy the output files to the target system if you are not running a Fast Clone Server on the target and have enabled that Fast Clone Server to accept output files from a remote system.
6. Optional. Disable or drop constraints on the target.
7. Run the load script.
8. Optional. Enable or add the constraints that you previously disabled or dropped on the target.

## Loading Data When Using DataStreamer

If you use DataStreamer, Fast Clone does not generate the output files including the load script. DataStreamer streams data directly to the target by using the following load mechanisms:

| Target Type | Load Mechanism |
|---|---|
| Greenplum | gpfdist utility |
| Netezza | Netezza external tables and Netezza ODBC driver |
| Teradata | Teradata Parallel Transporter (TPT) libraries with the load and update operators that are equivalent to the FastLoad and MultiLoad utilities, respectively |

With DataStreamer, perform the following tasks to load data:

1. Install connectivity software to connect to the target on the system where you run data unload jobs.
    - For Teradata targets, install the TPT libraries.
    - For Netezza targets, install the Netezza ODBC driver.

   For more information, see *Informatica Fast Clone Installation Guide.*
2. Create a configuration file.
    - For Greenplum targets, select the **Enable Greenplum gpfdist Direct Data Stream** option on the **Runtime Settings** tab > **Greenplum Load Settings** view.
      If you create the configuration file in a text editor, set the greenplum_direct_data_stream parameter to true.
    - For Netezza targets, select the **Enable Netezza Direct Data Stream** option on the **Runtime Settings** tab > **Netezza Load Settings** view.

If you create the configuration file in a text editor, set the netezza_direct_data_stream parameter to true.

- For Teradata targets, select the **Enable Teradata Direct Data Stream** option on the **Runtime Settings** tab > **Teradata Load Settings** view.
  If you create the configuration file in a text editor, set the direct_data_stream parameter to true.

3. Generate the target tables.
4. Optional. Validate the target schema.
5. Optional. Disable or drop constraints on the target.
6. Run the data unload job with the configuration file.
7. Optional. Enable or add the constraints that you previously disabled or dropped on the target.

# Configuration Considerations for Targets

To accurately load data to the target database, review the configuration considerations for your target type and the requirements of the target load utility.

## Considerations That Apply to Multiple Target Types

- Greenplum, Netezza, Teradata, and Vertica targets require data files to use a new-line character (\n) as a row separator. If the new-line character occurs in the Oracle data as a data value, the load fails. To prevent this problem, configure Fast Clone to replace each new-line character with the space character in one of the following ways:

  - In the Fast Clone Console, select the **Replace new line with space** option on the **Runtime Settings** > **Format Settings** view.
  - Manually enter the replace_new_line_with_space=true parameter in the configuration file.
  - At the command line, enter the replace_newline_with_space=true parameter.

  Replacement of the new-line character with the space character can increase the overhead of unload processing.

  **Note:** For Teradata targets, if you unload data in binary format, Fast Clone does not use a row separator in the output data files.

- Greenplum, and Vertica targets require the column separator and the enclosure character to be a single ASCII character. You can also specify a non-printable separator character by using the \x*hh* escape sequence in hexadecimal notation.

- Greenplum, Teradata, and Vertica targets require data files to not use a column separator at the end of each row. For Greenplum targets, Fast Clone sets the trailing_column_separator command-line parameter to false to generate valid output files. For the other targets, you must manually configure Fast Clone to produce data files without a column separator at the end of each row in one of the following ways:

  - In the Fast Clone Console, clear the **Trailing column separator** option on the **Runtime Settings** tab > **Format Settings** view.
  - Manually enter the trailing_column_separator parameter=false parameter in the configuration file.
  - At the command line, enter the trailing_column_separator=false command-line parameter.

## Considerations for Flat File Targets

- Fast Clone does not generate load scripts and control files for flat file targets.

### Considerations for Hadoop Targets

- To unload data to Hadoop targets, your Fast Clone license must include Hadoop and Hive target support.
- Fast Clone does not generate load scripts and control files for Hadoop targets.

### Considerations for Hive Targets

- To unload data to Hive targets, your Fast Clone license must include Hadoop and Hive target support.
- To unload data to remote Hive targets, the underlying Hadoop software must allow remote connections.
- Before you unload source data to a Hive target, enter the field delimiter that the Hive target uses in data files in the **Column separator** field on the **Runtime Settings** tab > **Format Settings** view. By default, Hive data warehouses use the ASCII start of heading (SOH) character as the field delimiter.
- Use the **Custom url** option to connect to a Hive target that requires Kerberos or LDAP authentication.
- Fast Clone does not support unloading source data to partitioned tables on Hive targets.
- Fast Clone does not support unloading source data to Hive tables that use custom locations.
- Fast Clone does not generate load scripts and control files for Hive targets.

### Considerations for Netezza Targets

- Netezza targets require the column separator to be a single ASCII character. You can also specify non-printable separator and enclosure character by using the \x*hh* escape sequence in hexadecimal notation.
- To make unload and nzload processing faster, delete the default double quotation mark (") character in the **Enclosed by** field on the **Runtime Settings** tab > **Format Settings** view.
- If you use the nzload utility to load data to a Netezza target, TIMESTAMP columns must use the format YYYY-MM-DD HH24:MI:SS.FF6, and DATE columns must use the format YYYY-MM-DD. The nzload utility supports these ISO date formats. Ensure that you specify the correct formats in the **Timestamp format** and **Date format** fields on the **Runtime Settings** tab > **Format Settings** view.
- If you unload data from CLOB or LONG columns and load these values with the nzload utility to a Netezza target, clear the **Unload binary into separate files** option. The nzload utility can load up to 32,768 bytes of data into VARCHAR columns.
- If you use DataStreamer to load data to a Netezza target, verify that the **Unload binary into separate files**, **Trailing column separator**, and **Suppress trailing null columns** options are not selected on the **Runtime Settings** tab > **Format Settings** view.
  Alternatively, verify that the export_binary_to_separate_files, trailing_column_separator, and suppress_trailing_nullcols parameters are set to false in the configuration file.

### Considerations for Teradata Targets

- For Teradata targets, delete the default double quotation mark (") character in the **Enclosed by** field on the **Runtime Settings** tab > **Format Settings** view so that the field is empty.
- In the **Truncate numeric precision integer** field on the **Runtime Settings** tab > **Format Settings** view, enter 18. Oracle NUMBER columns can have a maximum of 36 significant digits. Teradata DECIMAL columns can have a maximum of 18 digits. This parameter rounds unloaded numbers to 18 digits.
- If you want the output data file to be in Teradata binary format instead of text-delimited format, select the **Binary length prefix** option on the **Runtime Settings** tab > **Teradata Load Settings** view.

# Generating Target Tables

For each source table, Fast Clone requires a corresponding target table.

The source and target columns must have compatible datatypes. Also, the table and column names on the target must be identical to those on the source.

In the Fast Clone Console, you can perform the following tasks:

- Generate target tables based on the source table schema.
- Generate SQL CREATE TABLE scripts that you can run later to create target tables based on the source table schema.

If the source and target database types are different, Fast Clone uses the rules in the <ConversionRules> section of *FastClone_installation*\uiconf\DataTypes.xml file to convert source column datatypes to the compatible target column datatypes.

## Generating Target Tables Based on Source Table Schema

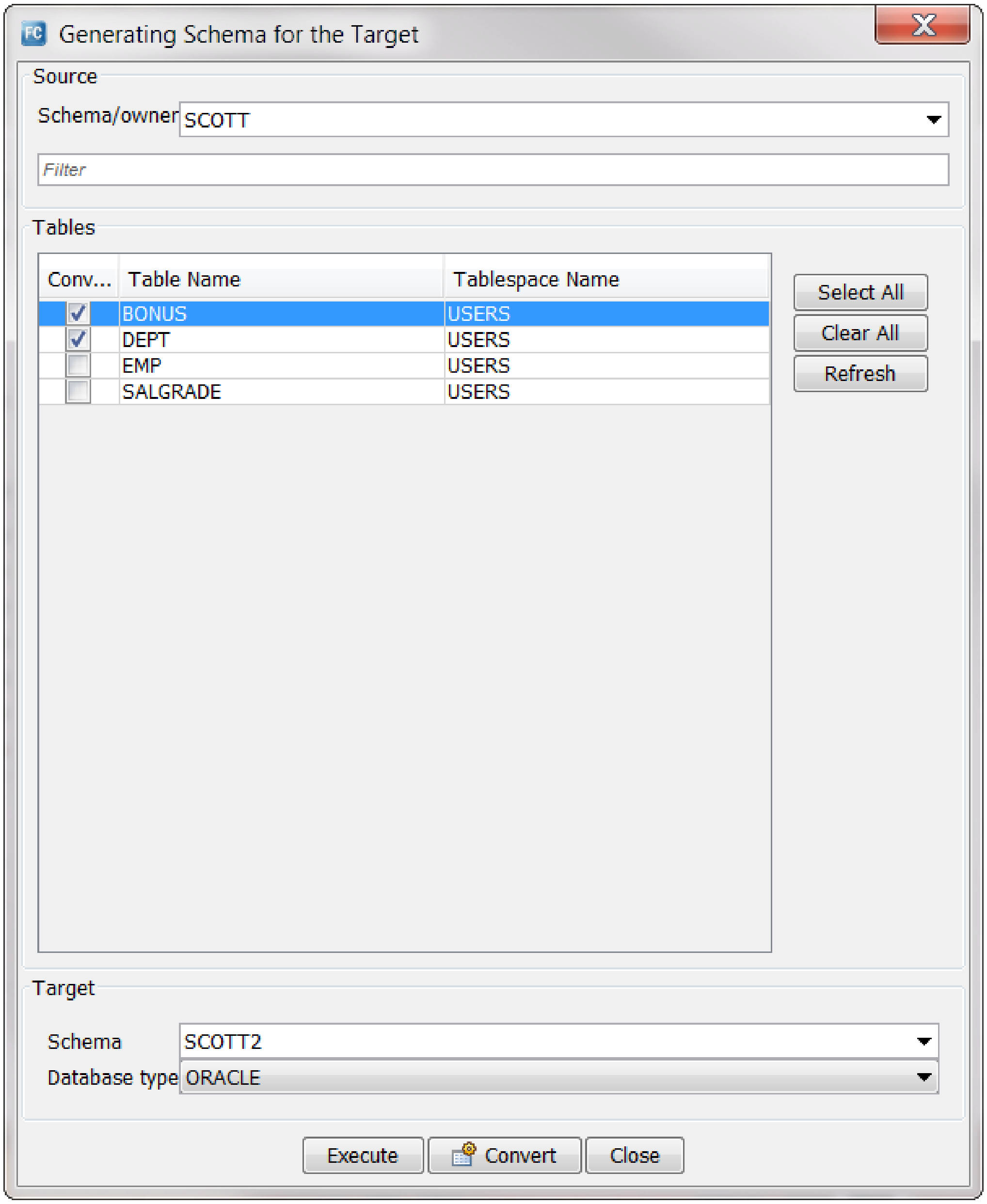
You can optionally generate target tables based on source table schema.

1. In the Fast Clone Console, connect to the source database and the target database.
2. Click **Schema** > **Schema creation script** on the menu bar, or click the **Generate tables on destination script** button on the toolbar.

   The **Generating Schema for the Target** dialog box appears.
3. In the **Schema/owner** field, enter the schema or owner name of source tables that you want to use to generate the target tables.

   The tables that match the selected schema or owner name are listed in the **Tables** box.

The following image shows an example listing:



Generating Schema for the Target

Source

Schema/owner: SCOTT

Filter

Tables

| Conv... | Table Name | Tablespace Name |
|---|---|---|
| ☑ | BONUS | USERS |
| ☑ | DEPT | USERS |
| □ | EMP | USERS |
| □ | SALGRADE | USERS |

Target

Schema: SCOTT2

Database type: ORACLE

4. To filter the list of tables, in the **Table filter** field, enter the first few letters of the source table names that you want to use for generating target tables. For case-sensitive filtering, enclose the filter value in double quotation marks.

   Only the tables that match the filter criteria are listed in the **Tables** box.

5. Select the tables to use for generating the target tables:
   - To select all listed tables, click **Select All**.
   - To select tables individually, select the **Convert** box next to each table.
6. Under **Target**, in the **Schema** field, enter a schema name for the target.
7. In the **Database type** list, select the target database type.
8. To create the tables, complete one of the following actions:

- To generate SQL CREATE TABLE scripts, click **Convert**. Then in the **Save** dialog box, save the generated scripts to a directory. You can run the scripts on the target system to create the target tables.
  The Fast Clone Console generates the SQL CREATE TABLE script in the specified directory. If you use sequence objects on the Oracle source, the Fast Clone Console also creates an SQL script for generating sequence objects in the specified directory. To name this script, the Fast Clone Console uses a name of the SQL CREATE TABLE script with the prefix _sequences.
- Click **Execute** to create the tables on the target immediately.

## Enabling the Extended Procedure for Generating Oracle Target Tables

By default, when generating target tables based on the source table schema, the Fast Clone Console preserves only the primary key constraints and column names that you define on the source.

For Oracle targets, if you also want to preserve default values, foreign key constraints, check constraints, unique constraints, indexes, and triggers that you define on the source, set the newDDLConversion parameter in the default.cfg file to false, as follows:

1. Close the Fast Clone Console.
2. In the *FastClone_installation*/uiconfig/default.cfg file, set the newDDLConversion parameter to false:

```
newDDLConversion=false
```

This parameter affects Oracle targets only. For other targets, the Fast Clone Console always uses the default target generation procedure.

# Validating the Target Schema

From the Fast Clone Console, you can validate the target schema to ensure that all of the source tables and columns have corresponding target tables and columns.

**Note:** The validation feature does not check whether the source column datatypes are compatible with the target column datatypes.

1. Connect to the source database and target database.
2. On the **Target DB** tab, ensure that you select the target schema or owner.
3. Click **Schema** > **Validate target** on the menu bar, or click the **Validate selected tables on destination** button on the toolbar.

   The **Validate tables structure on target** dialog box appears.
4. Review the validation results.

   The Fast Clone Console highlights validation errors in red.
5. Click **OK**.
6. Resolve validation errors, if any, and repeat validation procedure.

# Disabling or Dropping Target Constraints

From the Fast Clone Console, you can generate a script for disabling or dropping target table constraints, depending on the target type.

If you have target tables with primary key-foreign key relationships, run this script prior to loading data to avoid constraint violations on the target.

1. Connect to the source database and target database.
2. On the **Target DB** tab, ensure that you select the target schema or owner.
3. Click the **Schema** > **Disable Constraints** on the menu bar, or click the **Disable all constraints on all selected tables on destination** button on the toolbar.

   The **View Script** dialog box appears. It shows the generated script for disabling or dropping target constraints.
4. Perform one of the following actions:
   - Click **Run** to run the script immediately.
   - Click **Save** to save the generated script to a directory of your choice. You can run the script on the target system later to disable or drop constraints.

# Transmitting the Output Files to the Target

After Fast Clone completes the data unload job, you can either run the data load job remotely or transmit the output files to the target database system.

Typically, you run the data unload job and generate output files on the source database system.

Different strategies are available to transmit the output files to the target system. If the load utility for your target type is not installed on the source system, use one of the following strategies to transmit the output files:

- Run a Fast Clone Server instance on the target system. On the **Runtime Settings** tab > **Remote Server** view, configure the Fast Clone Server to receive output files.
- Manually transmit the output files to the target system by using FTP or another means. For example, archive the output files and then transfer them using FTP.

### Related Topics:

- “Configuring Fast Clone Server Tasks” on page 61

# Loading Data to the Target Database

If you do not use the DataStreamer component, you must run the load script that Fast Clone generates in the output directory to load data from output files or pipes to the target database. Fast Clone generates the load script for the native load utility of the target database.

The load script is a .cmd file on Windows and a .sh file on Linux and UNIX. The load script file name is based on the value of the **Load script base name** field on the **Runtime Settings** tab > **Files Locations** view in the

Fast Clone Console. If you configured the cloning operation from the command line, the load script file name is based on the value of the load_script_base_name command-line parameter.

1. Ensure that the target load utility is installed on the system where you run data load jobs.

   The target load utilities are usually installed as a part of the target database software and you do not need to install them separately. However, if you want to run the load operation from the source system or if you load data from pipes, you must install the target load utilities on the source system.
2. Review the load script that Fast Clone generated in the output directory. Optionally edit load options.

   Verify that Fast Clone correctly set connection information and load settings for the target load utility. For more information about load options, see the target database documentation for the load utility.
3. Run the load script from the command line.

# Enabling or Creating Target Constraints

If you previously disabled or dropped target constraints to avoid constraint violation errors, you can re-enable or re-create the constraints after loading data to the target database.

From the Fast Clone Console, generate a script for enabling or creating target constraints.

1. Connect to the source database and target database.
2. On the **Target DB** tab, ensure that you selected target schema or owner.
3. Click **Schema** > **Enable Constraints** on the menu bar, or click the **Enable all constraints on all selected tables on destination** button on the toolbar.

   The **View Script** dialog box appears and displays the generated script.
4. Perform one of the following actions:
   - Click **Run** to run the script immediately.
   - Click **Save** to save the generated script to a directory of your choice. You can run the script on the target system later to re-enable or re-create the constraints.

CHAPTER 6

# Remote Configuration Management

This chapter includes the following topics:

## Remote Configuration Management Overview

In the Fast Clone Console, you can use remote configuration management to open and save a cloning configuration file on a remote system. You must run an FTP server on the remote system to transfer the cloning configuration files.

After you save a cloning configuration file to a remote system, you can use it to run data unload jobs on the remote system from the command line. For example, if you run the Fast Clone Console on a standalone system, you can save the cloning configuration file to the source system and then run unload jobs from the command line on the source system using the direct path unload method. In this case, a Fast Clone Server is not required.

Also, you can use remote configuration management to store backups of the cloning configuration files on a remote system.

## Enabling Remote Configuration Management

To open and save configuration files on a remote system from the Fast Clone Console, you must run an FTP server on the remote system and enable remote configuration management.

1. Ensure that a FTP server is running on the remote system where you want to store the configuration files.
2. On the **Source DB** tab > **Source Database** view of the Fast Clone Console, select the **Remote configuration management** option.

   A confirmation dialog box appears.

3. Click **OK**.

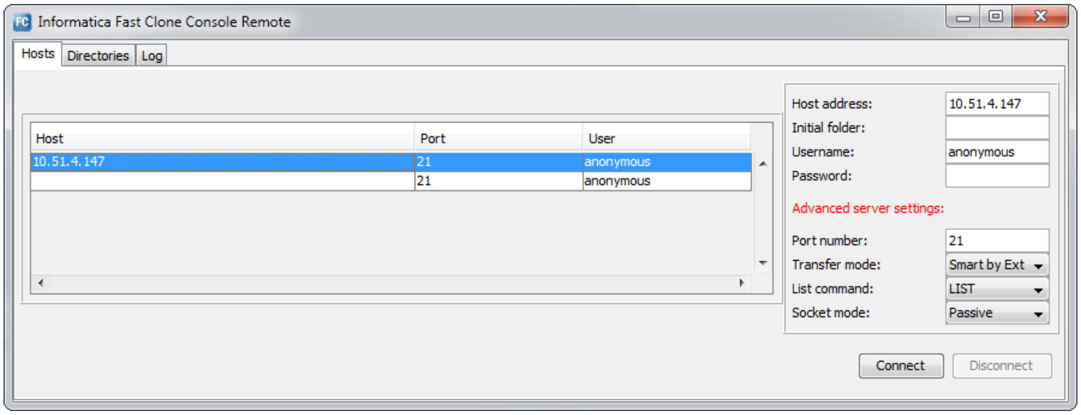
   On the menu bar, the **File** > **Open Remotely** and **File** > **Save Remotely** commands become available.

# Opening a Configuration File on a Remote System from the Fast Clone Console

In the Fast Clone Console, you can open a configuration file that is stored on a remote system by using FTP. You then view or edit the file in the Fast Clone Console. If you edit the file, you can save the updated file back to the remote system.

1. Verify that the **Remote configuration management** option is selected on the **Source DB** tab > **Source Database** view.
2. On the menu bar, click **File** > **Open Remotely**.

   The **Informatica Fast Clone Console Remote** window appears, as shown in the following image:


Informatica Fast Clone Console Remote
Hosts
Directories
Log
Advanced server settings:
Connect
Disconnect

| Host | Port | User |
|---|---|---|
| 10.51.4.147 | 21 | anonymous |
| | 21 | anonymous |

Host address: 10.51.4.147
Initial folder:
Username: anonymous
Password:
Port number: 21
Transfer mode: Smart by Ext
List command: LIST
Socket mode: Passive

3. Enter connection information for the FTP server that runs on the remote system.

   The following table describes the connection fields:

| Field | Description | Default |
|---|---|---|
| Host address | The host name or IP address of the remote system that stores the configuration file. | No default |
| Initial folder | The directory that opens after you connect to the remote system. Leave this field blank to open the root directory. | Blank |
| Username | A user name that is used to access the configuration file with FTP. | anonymous |
| Password | A valid password for the specified user. | Blank |
| Port number | The port number for the FTP server | 21 |

| Field | Description | Default |
|---|---|---|
| Transfer mode | The transfer mode that Fast Clone uses to transfer the cloning configuration files to the FTP server. Options are:<br>- **Smart by Ext**. Automatically determines the transfer mode based on the file extension.<br>- **Ascii Text**. Converts ASCII-formatted text configuration files to another format, if necessary.<br>- **Binary Data**. Transfers the files as a binary stream of data. | Smart by Ext |
| List command | A command that is used to get a list of files and directories on the remote system. | LIST |
| Socket mode | The mode for establishing the data connection. Options are:<br>- **Active**. The Fast Clone Console opens a socket and waits for the server to establish the transfer connection.<br>- **Passive**. The FTP server chooses a port for the data connection. The Fast Clone Console issues the PASV command and initiates data connections to the server by using the PASV port. | Passive |

4. Click **Connect**.

   If the connection is successful, the **Directories** tab appears and displays the file hierarchy on the FTP server.

   If the connection is not successful, click the **Logs** tab to review log information.

5. Select a configuration file to open and click **Open**.

   The Fast Clone Console opens the configuration file.

# Saving a Configuration File to a Remote System

In the Fast Clone Console, you can save a configuration file to a remote system by using FTP.

1. Ensure that the **Remote configuration management** option is selected on the **Source DB** tab > **Source Database** view.
2. On the menu bar, click **File** > **Save Remotely**.

   The **Informatica Fast Clone Console Remote** window appears.

3. Enter connection information for the FTP server.

   The following table describes the connection fields:

| Field | Description | Default |
|---|---|---|
| Host address | The host name or IP address of the remote system to which you are saving the configuration file. | No default |
| Initial folder | A directory that opens after you connect to the remote system. Leave this field blank to open the root directory. | Blank |
| Username | A user name that is used to access the configuration file with FTP. | Anonymous |

| Field | Description | Default |
|---|---|---|
| Password | A valid password for the specified user. | Blank |
| Port number | The port number for the FTP server. | 21 |
| Transfer mode | The transfer mode that Fast Clone uses to transfer the cloning configuration files to the FTP server. Options are:<br>- **Smart by Ext**. Automatically determines the transfer mode based on the file extension.<br>- **Ascii Text**. Converts ASCII formatted text configuration files to another format, if necessary.<br>- **Binary Data**. Transfers the files as a binary stream of data. | Smart by Ext |
| List command | A command that is used to get a list of files and directories on the remote system. | LIST |
| Socket mode | The mode for establishing the data connection. Options are:<br>- **Active**. The Fast Clone Console opens a socket and waits for the server to establish the transfer connection.<br>- **Passive**. The FTP server chooses a port for the data connection. The Fast Clone Console issues the PASV command and initiates data connections to the server using the PASV port. | Passive |

4. Click **Connect**.

   If the connection is successful, the **Directories** tab appears and displays the file hierarchy on the FTP server.

   If the connection is not successful, click the **Logs** tab to review log information.

5. Select a directory to which to save the configuration file to and click **Save**.

   The Fast Clone Console saves the configuration file to the specified directory.

Chapter 7

# Fast Clone Command Line Interface

This chapter includes the following topics:

## Command Line Interface Overview

You can run the Fast Clone executable with a configuration file or inline parameters directly from a command line.

If you use the cloning configuration file, run the Fast Clone executable with the config parameter. This parameter specifies a path to the configuration file that includes all of the parameters for the data-cloning job.

If you enter parameters for the data-cloning job directly from the command line, Fast Clone does not require a cloning configuration file. You enter the parameters at the command line after the Fast Clone executable name.

## Syntax

Fast Clone can accept any number of arguments from the command line. You can use the config parameter to specify a configuration file that contains the parameters, or enter parameters individually at the command line when you run the Fast Clone executable.

Informatica recommends that you run Fast Clone with a configuration file. Use the following syntax:

- On a Windows system:

  ```
  FastReader.exe config=configuration_file [remotehost=Fast_Clone_Server_address:port]
  ```

- On a Linux or UNIX system:

  ```
  ./FastReader config=configuration_file [remotehost=Fast_Clone_Server_address:port]
  ```

Include the optional remotehost parameter to send the unload request to a remote Fast Clone Server.

If you want to enter Fast Clone parameters directly at the command line, do not use the config parameter. Use the following syntax:

- On a Windows system:

```
FastReader.exe parameter_1=value parameter_n=value
```

- On a Linux or UNIX system:

```
./FastReader parameter_1=value parameter_n=value
```

**Important:** For Teradata targets, use the FastReaderTD.exe on Windows or FastReaderTD on Linux or UNIX.

Include a space between each parameter specification. When you specify boolean parameters from the command line, you can use true or false as the value or the shorter form of y or n. For example, both forms of the following parameter are acceptable and have the same function:

```
compress_on_the_fly=true
compress_on_the_fly=y
```

# Command Line Parameters

Review the descriptions of the command line parameters to determine if you want to use them. The parameter descriptions are grouped by general parameter type. The parameter types are similar to the tabs in the Fast Clone Console.

## Source Database Parameters

Specify the following command line parameters to connect the source database:

**asm_debug**

Set this parameter to true only at the request of Informatica Global Customer Support. For Oracle sources in an Automatic Storage Management (ASM) environment, a value of true causes Fast Clone to print an ASM debug log to the asm_debug.log file.

Default value: false

**asm_instance_host**

The host name or IP address of the system with the ASM instance.

**asm_instance_name**

The ASM instance name to which Fast Clone connects to unload data.

Default value: No default value is provided.

**asm_instance_port**

The port number that is used to connect to the ASM instance that is specified in the asm_instance_name parameter.

Default value: 1521

**asm_instance_sid**

The ASM instance SID.

**asm_protocol**

The protocol that Fast Clone uses to connect to an ASM instance. Valid values are:

- **TCP**. Use the TCP/IP protocol.
- **TCPS**. Use the TCP/IP protocol with the Secure Sockets Layer (SSL).

Default value: TCP

**asm_sys_password**

A password that is defined for the ASM SYS user.

**asm_sys_username**

The user name of the default Oracle ASM SYS user that was created at Oracle installation, or another user that you define. This user must have SYSDBA privileges.

Default value: SYS

**owner**

The name of the source schema owner for the tables to be unloaded. This owner name is also the schema name for the tables.

**protocol**

The protocol that Fast Clone uses to connect to the source database. Valid values are:

- **TCP**. Use the TCP/IP protocol.
- **TCPS**. Use the TCP/IP protocol with the Secure Sockets Layer (SSL).

Default value: TCP

**source_connection_string**

The TNS connection string for the Oracle source database. Grant the SELECT ANY DICTIONARY privilege role if you use the direct path unload method.

**userid**

The connection string for the Oracle source database in the following format:

```
username/password@net_service_name
```

Instead of the ***net_service_name***, you can use the IP address, port, and service name of the source database:

```
username/password@//IP_address:port/service_name
```

## Source Tables, Columns, and SQL Queries

Use the following command line parameters to specify the source tables and columns from which to unload data and any SQL queries that you want to use for unload processing:

**all_tables**

Indicates whether to unload all tables in the schema that is identified by the owner parameter and specifies the unload method to use for these tables. Options are:

- **0**. Use the conv_path_tables parameter value to determine the tables to unload with the conventional path unload method, and use the tables parameter to determine the tables to unload with the direct path unload method. Specify at least one table in one of the parameters for the unload job to run.
- **1**. Let Fast Clone determine whether it can perform a direct path unload. If a direct path unload is not possible, Fast Clone performs a conventional path unload.

- **2**. Force Fast Clone to perform a direct path unload for all source tables in the schema that is identified by the owner parameter.
- **3**. Force Fast Clone to perform a conventional path unload for all source tables in the schema that is identified by the owner parameter.

Default value: 0

**conv_path_tables**

A space- or comma-separated list of source tables or table partitions from which to unload data with the conventional path unload.

**sql_queries**

A space- or comma-separated list of SQL queries that are processed with the conventional path unload method. The queries must be enclosed in quotation marks. Add a backslash before each quotation mark. For example:

```
sql_queries=\"SELECT * FROM TABLE1\",\"SELECT * FROM TABLE2\"
```

Fast Clone writes the results of these queries to the QUERY1.dat and QUERY2.dat files.

**tables**

A space- or comma-separated list of source tables or table partitions from which to unload data with the direct path unload method. For example:

```
tables=TABLE1,TABLE2
```

## Target Database Parameters

Specify the following command line parameters to connect to the source database:

**destination_database_type**

The target database type. Options are:

- **flat**. Use this option for flat file targets.
- **greenplum3**. Use this option for Greenplum targets.
- **cloudera**. Use this option for Cloudera targets.
- **hive**. Use this option for Hive targets.
- **hortonworks**. Use this option for Hortonworks targets.
- **mapr**. Use this option for MapR targets.
- **mssql**. Use this option for Microsoft SQL Server targets.
- **netezza**. Use this option for Netezza targets.
- **oracle**. Use this option for Oracle targets.
- **teradata**. Use this option for Teradata targets.
- **vertica**. Use this option for Vertica targets.

Default value: oracle

**destination_host**

The host name or IP address of the system where the target database runs.

**destination_instance**

The target instance name or database name for the tables to be loaded.

**destination_load_batchsize**

The number of rows in a batch that is passed to the Microsoft SQL Server osql utility for loading data to the target. The osql utility copies each batch to the target server as one transaction.

Fast Clone uses this value as the BATCHSIZE option value in the BULK INSERT statement for the osql utility.

Range of values: 1 through 1000000

Default value: 10000

**destination_load_maxerrors**

For Greenplum and Teradata targets, the maximum number of errors that can occur when streaming data to the target database with DataStreamer. If the number of errors exceeds this maximum, DataStreamer ends with an error.

For Microsoft SQL Server targets, specifies the maximum number of syntax errors that can appear in the output data files before the osql utility cancels the BULK INSERT operation. The osql utility ignores all of the rows that it cannot parse and counts an error for each row. Fast Clone uses this value as the MAXERRORS option value in the BULK INSERT statement for the osql utility.

Range of values: 1 through 1000000

Default value: 100000

**destination_password**

A valid password for the user that is specified by the destination_user parameter. If you specify a forward slash (/) in the destination_user parameter, Fast Clone ignores the password and uses operating system authentication

**destination_port**

The port number that Fast Clone uses to connect to the target database.

Default value: No default value is provided.

**destination_schema**

The target schema name or schema owner for the tables to be loaded.

**destination_user**

The name of a user who has permissions to connect to the target database and load the source data. If you specify a forward slash (/) in this parameter, Fast Clone ignores the password and uses operating system authentication.

**greenplum_loader_home**

For Greenplum targets, if you load data to the target database from data files or pipes, the path to the Greenplum loader root directory that Fast Clone uses to locate the gpload utility and generate valid load scripts.

## Runtime Command Line Parameters

You can specify runtime settings when you start Fast Clone in the command line.

**big_raw_device_offset_mul**

For Oracle sources that use raw devices on AIX, the number of offset blocks to skip from the beginning of a DF_LGDSK raw device when the calc_raw_device_offset parameter is set to true. You can specify the size of a single offset block in the raw_device_offset parameter. To get the total number of bytes that

Fast Clone skips on DF_LGDSK raw devices, multiply the raw_device_offset value by the big_raw_device_offset_mul parameter value.

Valid values: 1 through 64

Default value: 32

### binary_lengh_prefix

For Teradata targets, indicates whether to unload source data in binary format. Each record in binary format can include a 2-byte field that indicates the total length of the record, some optional bytes for null values, and the data.

For Vertica targets, indicates whether to unload source data in the NATIVE VARCHAR or text format. If you set this parameter to true, Fast Clone adds the NATIVE VARCHAR keyword to the COPY or LCOPY command in the output SQL load script. The database server converts CHAR or VARCHAR fields in the data files to Vertica datatypes when loading data. If you set this parameter to false, Fast Clone loads source data in text format with delimiters. Valid values are true and false.

Default value: false

### binary_lengh_prefix_indicators

For Teradata targets, if you set the binary_lengh_prefix parameter to true, indicates whether to set the indicator bits in the binary output for columns that contain null values. Valid values are:

- **true**. Set the indicator bits to 1 for null values.
- **false**. Do not set indicator bits.

Default value: false

### binary_lengh_prefix_little_endian

If you set the binary_length_prefix parameter to true, indicates whether to use LITTLE ENDIAN or BIG ENDIAN for the binary output. For Teradata targets, this parameter determines the format of the integer that specifies the length of records and fields in the binary files. Valid values are:

- **true**. The binary output uses the LITTLE ENDIAN format.
- **false**. The binary output uses the BIG ENDIAN format.

Default value: true

### calc_raw_device_offset

Indicates whether Fast Clone skips the offset block in raw devices. Valid values are:

- **true**. Skip the offset block in raw devices.
- **false**. Read data from the beginning of the raw device.

To get the total number of bytes that Fast Clone skips in a DF_LGDSK raw device, multiply the raw_device_offset value by the big_raw_device_offset_mul parameter value.

Default value: true

### checkpoint_each_x_tables

Specifies the number of tables from which Fast Clone unloads data before taking a checkpoint.

Valid values: 0 through 100000

Default value: 0. With this default, Fast Clone takes a checkpoint only before it starts unloading source data.

#### column_separator

Specifies a character or character sequence that Fast Clone uses to separate columns in the output data files. Specify a character or character sequence that does not appear in the source data. Otherwise, the target load utility cannot correctly parse the data files. For more information, see “Selecting Column and Row Separator and Enclosure Character to Use for Data Loading” on page 50.

You can specify a column separator that uses the \x*hh* escape sequence in hexadecimal notation.

Also, you can use the following non-printable ASCII characters as column separators:

- \a (alert)
- \b (backspace)
- \n (new line)
- \r (carriage return)
- \t (horizontal tab)
- \v (vertical tab)

If you use non-printable ASCII characters, ensure that the target load utility supports these characters as separators. For more information, see the target database documentation.

Default value: semicolon (;)

#### compress_network_traffic

Indicates whether to compress network traffic. Valid values are:

- **true**. Compress network traffic.
- **false**. Do not compress network traffic.

Default value: false

#### compress_on_the_fly

Indicates whether to compress the output data files during unload processing to reduce the file sizes. Valid values are:

- **true**. Compress the output data files. The compression method is specified in the compression_type parameter.
- **false**. Do not compress the output data files.

Default value: false

#### compression_level

Specifies the compression level for the GZIP compression method. Higher values result in faster compression and a lower compression ratio. Lower values result in slower compression and a higher compression ratio. For Fast Clone to use this parameter, the compression_type parameter must be set to GZIP and the compress_on_the_fly parameter must be set to true.

Valid values: 1 through 9

Default value: 1

#### compression_type

If you set the compress_on_the_fly parameter to true, specifies the compression method that Fast Clone uses to compress the output data files. Valid values are:

- **GZIP**. Provides a better compression ratio but is slower. You can set the GZIP compression level in the compression_level parameter.
- **QZIP**. Provides a lower compression ratio but is faster.

Default value: GZIP

### consistent_lock_on_table_set

If you use the direct path unload method and set the dirty parameter to false, indicates whether to lock all of the selected tables in a table set before unloading the source data. Valid values are:

- **true**. Fast Clone locks all of the selected source tables in a table set before unloading the source data. Fast Clone unlocks these tables after unloading data from all of them. Enter this value to get consistent point-in-time data across the selected source tables.
- **false**. Fast Clone holds a lock on each source table only while unloading data from the table.

Default value: false

### convert_ascii_to_ebcidic

Indicates whether Fast Clone converts source data to the IBM EBCIDIC encoding when unloading data to output data files and pipes. Valid values are:

- **true**. Convert source data to the IBM EBCIDIC encoding.
- **false**. Preserve the original encoding.

Default value: false

### convert_numbers_to_comp3_fl_8b

Indicates whether to write numeric data to the output data files as COMP-3 values. Valid values are:

- **true**. Convert numeric data to COMP-3 in the output data files. Use this value if you need to read the output file with COBOL or another language that has built-in COMP-3 support.
- **false**. Preserve the original numeric format.

Default value: false

### convert_numbers_to_comp3_implied_decimal

The position of the decimal point in the COMP-3 format.

Valid values: 0 through 14

Default value: 0

### convert_utf16_clobs_to_utf8

Indicates whether Fast Clone converts CLOB data from UTF-16 to UTF-8 encoding in the output data files. Valid values are:

- **true**. Convert source character data in CLOB columns from UTF-16 to UTF-8 encoding.
- **false**. Do not convert source character data in CLOB columns. If you do not use UTF-16 encoding for the source data, set this parameter to false to avoid the overhead of character set conversion.

Default value: false

### convert_utf16_nchars_to_utf8

Indicates whether Fast Clone converts NCHAR data from UTF-16 to UTF-8 encoding in the output data files. Valid values are:

- **true**. Convert source character data in NCHAR columns from UTF-16 to UTF-8 encoding.
- **false**. Do not convert source character data in NCHAR columns.

**Note:** If you do not use UTF-16 encoding for the source data, set this parameter to false to avoid the overhead of character set conversion. To use the parameter, you must manually enter it at the command line or in the cloning configuration file. You cannot enter it in the Fast Clone Console.

Default value: false

#### convert_utf16_nclobs_to_utf8

Indicates whether Fast Clone converts NCLOB data from UTF-16 to UTF-8 encoding in the output data files. Valid values are:

- **true**. Convert source character data in NCLOB columns from UTF-16 to UTF-8 encoding.
- **false**. Do not convert source character data in NCLOB columns.

**Note:** If you do not use UTF-16 encoding for the source data, set this parameter to false to avoid the overhead of character set conversion. To use the parameter, you must manually enter it at the command line or in the cloning configuration file. You cannot enter it in the Fast Clone Console.

Default value: false

#### create_external_table_log_file

For Oracle targets, if you create an SQL script for creating external tables based on the output files, indicates whether to log messages related to this processing to a log file. Valid values are:

- **true**. The Oracle external table utility logs messages to a log file when accessing data in the output data files. Ensure that the create_external_table_script parameter is also set to true.
- **false**. The Oracle external table utility does not log these messages.

Default value: false

#### create_external_table_script

For Oracle targets, indicates whether Fast Clone generates an SQL script to create Oracle external tables based on the output data files. Valid values are:

- **true**. Create an SQL script to create Oracle external tables based on the output data files.
- **false**. Do not create an SQL script to create Oracle external tables based on the output data files.

Default value: false

#### create_load_input_only

Indicates whether Fast Clone creates the load script and control files but skips unloading data from the source database. Valid values are:

- **true**. Create a load script and the control files. Do not unload data from the source database.
- **false**. Create a load script and the control files. Unload data from the source database.

**Note:** Fast Clone does not generate load scripts and control files for flat file, Hadoop, and Hive targets.

Default value: false

#### create_output_pipe_instead_of_file

Indicates whether Fast Clone creates pipes in the output directory and writes the output data to these pipes. Valid values are:

- **true**. Fast Clone creates pipes in the output directory.
- **false**. Fast Clone does not create pipes in the output directory. Use this value to unload data to output data files or to the pipes that you manually created.

Default value: false

### date_format

Specifies the date format to use in output files. For information about datetime format elements, see the Oracle documentation. For example, the following format values are valid:

- YYYY-MM-DD HH24:MI:SS
- DDMMYYYY
- YYYY-MM-DD
- HH24:MI:SS

Default value: YYYY-MM-DD HH24:MI:SS

### dbsync_config_xml

Specifies the path and file name for the Data Replication configuration file that can be in XML or SQLite format. If you specify the config parameter, enter an absolute path. If you do not specify the config parameter, enter a path that is relative to the DBSYNC_HOME directory.

**Important:** If you use a Data Replication configuration in SQLite format, do not copy or move the configuration files from the default *DataReplication_installation*/configs directory to another location.

### dbsync_config_xsd

Specifies the path and file name for the Data Replication configuration schema .XSD file. The path is relative to the DBSYNC_HOME directory.

Default value: config.xsd

### dbsync_integration

Indicates whether Fast Clone integrates with Informatica Data Replication. Also, when integration is enabled, determines whether Fast Clone allows user sessions that are connected to the source tables to complete processing before locking the tables. Valid values are:

- **false**. Do not integrate Fast Clone with Data Replication.
- **lock**. Integrate Fast Clone with Data Replication. Fast Clone waits for all user sessions that are connected to the source tables to complete processing and close before locking the tables for unload processing.
- **lock_disconnect**. Integrate Fast Clone with Data Replication. Fast Clone forces all user sessions that are connected to the source tables to disconnect and then locks the tables for unload processing.

Default value: false

### dbsync_integration_by_direct

If you enable integration with Data Replication, indicates whether to unload data from the source database with the direct path unload method. Valid values are:

- **true**. Use the direct path unload method.
- **false**. Use the conventional path unload method.

Default value: true

### dbsync_integration_update_scn

Indicates whether to update the Data Replication configuration file with the SCN for each mapped table after Fast Clone unloads the source data. Valid values are true and false.

**Important:** Set this parameter to true if you plan to use Data Replication to replicate change data and use Fast Clone to perform initial materialization of the replication target tables.

Default value: true

**decimal_separator**

Specifies the decimal separator that Fast Clone uses in decimal numbers in unloaded data.

Default value: period (.)

**default_long_raw_is_hex**

For Oracle targets, indicates whether to unload BLOB and LONG RAW source data in hexadecimal format. Set this parameter to true to unload data with these datatypes in hexadecimal format. For other target types, set this parameter to false to avoid generating incorrect control files.

Default value: false

**dest_tables**

If you set the dbsync_integration parameter to lock or lock_disconnect, specifies the list of tables that must be resynchronized.

**direct_file_access**

Indicates whether to read extent information for parallel unload processing directly from the data file. Specify true to read extent information from the data file.

Default value: false

**dirty**

Indicates whether to lock the tables that are unloaded with the direct path unload method to perform consistent read operations. Valid values are:

- **true**. Do not lock the tables before unloading the source data. Use this option if you want to bypass the Oracle consistency mechanism. However, block-level consistency is preserved, because block I/O operations are atomic.
- **false**. Lock the tables exclusively before unloading the source data.

Default value: true

**disable_linux_cache_bypass**

For Linux operating systems, indicates whether to use direct I/O mode to read Oracle data files. In direct I/O mode, Fast Clone bypasses operating system caching and reads data directly from disk. Set this parameter to true to use direct I/O mode.

Default value: false

**escape_enclosure**

Specifies an escape character that Fast Clone inserts before the character that is specified in the optionally_enclosed_by parameter when the character appears in text columns. If you define this parameter, Fast Clone truncates the optionally_enclosed_by value to a single character.

Default value: No default is provided.

**explicitly_flush_buffer_cache**

Indicates whether Fast Clone clears all data from the buffer cache of the Oracle source database before unloading the data with the direct path unload method. Valid values are:

- **true**. Fast Clone clears all data from the buffer cache by executing the following command on the Oracle source database:

  ```
  ALTER SYSTEM FLUSH BUFFER_CACHE;
  ```
- **false**. Fast Clone does not clear data from the buffer cache.

Default value: false

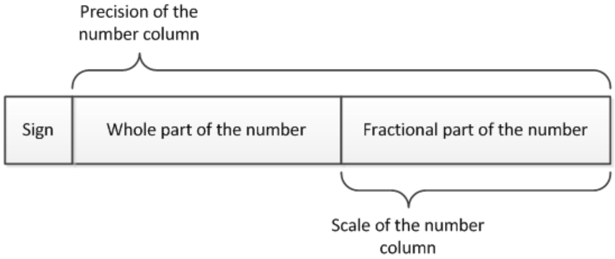
### fixed_size_characters_by_col_definition

Indicates whether to unload data from character columns in fixed-length format. The fixed length is determined by the column length. If a text value is shorter than the column length, Fast Clone pads the column with spaces up to the full column length. Specify true to unload character data in fixed-length format.

Default value: false

### fixed_size_number_by_col_definition

Indicates whether to unload data from NUMBER columns in the fixed-length format. Specify true to unload character data in fixed-length format. The fixed length is defined by the precision of the NUMBER column, without a decimal separator. If a number is shorter than this length, Fast Clone pads the value with zeroes. The following image shows the fixed-length format of a NUMBER column:


Precision of the number column
Sign
Whole part of the number
Fractional part of the number
Scale of the number column

For example, when using fixed-length format, Fast Clone unloads the value 123.45 from the NUMBER(10,5) column as +0012345000.

Default value: false

### limit_result_set

Specifies the maximum number of rows that Fast Clone unloads from each table. Set this parameter to -1 to unload all rows from a table.

Valid values: -1 through 1073741823

Default value: -1

### load_data_to_same_partition_as_source

For Oracle targets, indicates whether SQL*Loader loads data to a table partition that has the same name as the corresponding source table partition. Specify true to load data from a source table partition to a target table partition of the same name.

Default value: false

### load_script_base_name

Specifies the file name of the load script. On Windows, Fast Clone generates load script files with the .cmd extension. On Linux and UNIX, Fast Clone generates load script files with the .sh extension.

Default value: The source schema owner name that is specified by the owner parameter.

### lobs_to_separate_files

Indicates whether to write data from BLOB, CLOB, NCLOB, LONG, and LONG RAW source columns to separate files. In the regular data files, Fast Clone includes the file name instead of the source value. Specify true to write data from columns with these datatypes to separate files. For Hive targets, Fast Clone always uses the parameter value of false.

Default value: false

**Important:** If you use DataStreamer to load source data to Netezza targets, set this parameter to false.

### local_time_zone

For Teradata targets, specifies the local time zone to use for unloading data from TIMESTAMP WITH LOCAL TIME ZONE columns. Specify the time zone as an offset from the Oracle database time zone:

```
"+/-HH:MM"
```

The *HH* variable is a two-digit number from -13 to +13. The *MM* variable is a two-digit number from 0 to 59.

For example, enter "+08:00".

Default value: No default is provided.

### log

Indicates whether Fast Clone produces an extended log. Set this parameter to true only at the request of Informatica Global Customer Support.

Default value: false

### mb_per_file

Defines the maximum amount of data, in megabytes, that a single output data file (.dat file) can contain.

Valid values: -1 through 65530

Default value: -1. With this default value, Fast Clone does not enforce a limit on the size of the output data files.

### multiblockread

Specifies the number of Oracle data blocks that Fast Clone fetches per read operation.

**Important:** Change this parameter only if Fast Clone unload performance is slow. The value of this parameter depends on the extent size for locally managed tablespaces and the table-specific extent sizes for dictionary-managed tablespaces.

Valid values: 1 through 1024

Default value: 32

### nvl_fixed_size_characters_by_space

For character columns that are unloaded in fixed-length format, indicates whether to replace null values with space characters. Specify true to replace nulls with spaces.

Default value: false

### optionally_enclosed_by

Specifies the character or character sequence that Fast Clone uses to enclose character data in output data files. Specify a character or character sequence that does not appear in the source data. The target load utility cannot correctly parse the data files if the enclosure character is a source data value.

If you use the default value and the source columns contain double-quotation marks (") in the data, specify a different character or character string for this parameter or use the escape character in the escape_enclosure parameter to escape the double-quotation marks in the column data.

**Note:** If you use the character that is defined by the escape_enclosure parameter to escape the optionally_enclosed_by value in the source data, Fast Clone truncates the optionally_enclosed_by value to a single character.

Default value: Double-quotation mark (")

**outdir**

Specifies the directory in which Fast Clone creates the output files. The output directory can be on a local file system or on a network mapped share, such as NFS or Samba. For Hive targets, specifies the location of flat files that store table data on the HDFS.

Default value: The FAST_READER_HOME environment variable value.

**output_file_extension_base**

Specifies the extension that Fast Clone uses for output data files. Fast Clone creates a data file for each source table from which data is unloaded.

Default value: dat

**parallel_segments**

Specifies the maximum number of tables or table partitions that Fast Clone can unload in parallel with the conventional path unload method.

Valid values: 1 through 64

Default value: 1

**Note:** If you also specify the threads_num parameter, the parallel_segments parameter overrides it.

**print_column_names**

For flat file targets, indicates whether Fast Clone writes column names in the first row of the output data files. Specify true to write column names in the first row.

Default value: false

**Note:** This parameter is available for all target types when the unload_data_only parameter is set to true.

**raw_device_offset**

The size of a single offset block, in bytes, in a raw device. Fast Clone skips the offset block if you set the calc_raw_device_offset parameter to true. To get the total number of bytes that Fast Clone skips on DF_LGDSK raw devices, multiply the raw_device_offset value by the big_raw_device_offset_mul parameter value.

Default value: 0 for Linux, 4096 for other operating systems

**record_separator**

A character or character sequence that Fast Clone uses to separate table rows in the output data files. Specify a character or character sequence that does not appear in the source data. Otherwise, the target load utility cannot correctly parse the data files. For more information, see "Selecting Column and Row Separator and Enclosure Character to Use for Data Loading" on page 50.

You can specify a row separator that uses the \x*hh* escape sequence in hexadecimal notation.

Also, you can use the following non-printable ASCII characters:

- \a (alert)
- \b (backspace)
- \n (new line)
- \r (carriage return)
- \t (horizontal tab)
- \v (vertical tab)

If you use non-printable ASCII characters in row separators, ensure that the target load utility supports these separators. For more information, see target database documentation.

Default value: \n

### replace_newline_with_space

Indicates whether to replace new line (\n) character in unloaded data with space characters. Specify true if you use the new line character as the row separator in output files and the source character data includes new line characters. However, with a value of true, unload processing performance might be degraded.

Default value: false

### resync

If you set the dbsync_integration parameter to lock or lock_disconnect, indicates whether to resynchronize all of the mapped target tables or a subset of the mapped target tables. Valid values are:

- **y**. Synchronize all of the mapped target tables with the source tables.
- **n**. Synchronize the set of tables that you specify in the dest_tables parameter.
- **p**. Synchronize the mapped target tables for which Sync SCN value is -1.

Default value: p

### right_pad_columns

Indicates whether to pad columns on the right with spaces when the length of a column value is less than the full length of the column. Specify true to enable padding.

Default value: false

### run_as_service

Indicates whether to run the Fast Clone Server instead of the Fast Clone unload engine. Enter true to run the Fast Clone Server.

Default value: false

### select_from_sys

Indicates whether Fast Clone uses SYS-owned dictionary tables, such as sys.obj$ and sys.tab$, to get extent boundaries. Valid values are:

- **true**. Use SYS-owned dictionary tables to get extent boundaries.
- **false**. Use dba_extents and dba_segments views to get extent boundaries.

**Note:** If you set this parameter to false for Oracle sources that use locally managed tablespaces, retrieval of extent information from the dba_extents view is slow, especially if the dictionary is partially analyzed and the optimizer mode is not RULE. In this situation, Informatica recommends that you set this parameter to true and grant the SELECT ANY DICTIONARY permission to the user.

Default value: false

### server_config

Specifies the path to the Fast Clone Server configuration .ini file. Enter this parameter if you set the run_as_service parameter to true.

### show_increments_rows

The number of rows that Fast Clone unloads with the direct path unload method before writing updated progress information for the unload job to the system console or command prompt window.

Valid values: 10 through 10000000

Default value: 50000

### skip_all_out_file_ops

For Greenplum and Teradata targets, indicates whether Fast Clone skips the writing of output files. Set this parameter to true if you use DataStreamer or if you need to troubleshoot jobs that unload a large volume of data. Valid values are true and false.

Default value: The value that is specified in the greenplum_direct_data_stream or teradata_direct_data_streamparameter.

### split_files_by_row_count

The maximum number of rows that Fast Clone writes to an output data file. If Fast Clone needs to unload a greater number of rows from the source table, Fast Clone creates multiple output data files for the table. The file names have the following format:

```
table_name_file_number.dat
```

Specify this parameter only if you use the direct path unload method.

**Note:** For parallel unload processing, splitting output into multiple data files based on a number of rows is slower than splitting output based on a chunk size. This performance degradation occurs because of locks between the parallel threads.

Valid values: -1 through 1000000000

### teradata_direct_data_stream

For Teradata targets, indicates whether DataStreamer sends the source data directly to Teradata Parallel Transporter (TPT) for loading to the target database. Valid values are:

- **true**. Do not create output data files. Send the source data directly to the TPT utility.
- **false**. Unload the source data to the output data files.

Default value: true

### teradata_drop_rebuild_tables

For Teradata targets, indicates whether DataStreamer drops and re-creates tables in the target database before it loads the source data. Valid values are true and false.

Default: false

### teradata_load_sessions

For Teradata targets, specifies the number of sessions that the Teradata FastLoad or MultiLoad utility can open to load the source data to each target table.

Default value: 2

### teradata_loader_to_use

For Teradata targets, specifies the load utility that Fast Clone uses to load source data to the target. Valid values are:

- teradata_fast_load
- teradata_multi_load
- teradata_parallel_dpump

Default value: teradata_fast_load

### teradata_nodes_ip_list

For Teradata targets, specifies a comma- or space-separated list of IP addresses of Teradata nodes.

#### teradata_parallel_transporter

For Teradata targets, indicates whether Fast Clone uses Teradata Parallel Transporter to load the source data to the target database. Valid values are true and false.

Default value: false

#### threads_num

Specifies the maximum number of tables or table partitions that Fast Clone can unload in parallel with the direct path unload method.

Valid values: 1 through 64

Default value: 1

#### threads_per_segment

Specifies the maximum number of threads that Fast Clone can use to unload data from each table or table partition with the direct path unload method.

**Note:** Fast Clone might use fewer threads. Fast Clone does not create a thread to unload a chunk of less than 100 data blocks.

Valid values: 1 through 64

Default value: 1

#### timestamp_format

Specifies the timestamp format in the output. For more information on datetime format elements, see the Oracle documentation. For example:

- YYYY-MM-DD HH24:MI:SS.FF6
- YYYY-MM-DD HH24:MI:SS
- DDMMYYYY
- YYYY-MM-DD
- HH24:MI:SS

**Note:** For the conventional path unload method, Fast Clone supports all timestamp formats.

Default value: YYYY-MM-DD HH24:MI:SS.FF6

#### trailing_column_separator

Indicates whether to add a column separator after the last column value in each row in the output. Valid values are true and false.

**Note:** Fast Clone forces this parameter to false for Oracle targets if the unloaded tables contain LOB columns and the export_binary_to_separate_file parameter is set to true.

Default value: true

**Important:** If you use DataStreamer to load data to a Netezza target, set this parameter to false.

#### truncate_numeric_precision

Specifies the number of significant digits to use for rounding down the source numeric values in the output. This parameter truncates only the fractional part of a number.

Valid values: -1 through 40

Default value: **-1**. Fast Clone does not round down the source numeric values.

**unload_data_only**

Indicates whether to unload source data only and skip the creation of control files and the load script. Valid values are:

- **true**. Do not create control files and a data load script when Fast Clone unloads source data to .dat files.
- **false**. Create control files and a data load script when Fast Clone unloads source data to .dat files.

Default value: false

For flat file, Hadoop, and Hive targets, Fast Clone always uses the parameter value of true.

**use_dbms_diskgroup_instead_of_direct**

Indicates whether Fast Clone uses the DBMS_DISKGROUP package to read data from ASM sources. Typically, use of the DBMS_DISKGROUP package is significantly slower than reading data directly from ASM sources because the package has a PL/SQL limit of 32,600 bytes per read.

However, if you use the conventional path method and set this parameter to true, Fast Clone unloads data from ASM by using multiple threads. Fast Clone can use multiple threads from a remote computer, with an even data distribution across the threads.

Valid values are true and false.

Default value: false

# Command Line Example

An example unload script that is named cmd_unload_example.sh is located in the *FastClone_installation*\doc directory. It demonstrates how to run Fast Clone from the command line.

Use the example script as a template for creating your own scripts. Copy the script and then edit the copy.

**Important:** Ensure that the environment variables in the script are correct for your environment.

The following code sample shows how to run Fast Clone from the command line on Linux or UNIX:

```
FastReader userid=DATAREP_USER/DATAREP_USER@//192.168.137.111/orcl.mshome.net
destination_database_type=oracle owner=DATAREP_USER conv_path_tables=TABLE1
outdir=oracle_output
```

# Appendix A

# Troubleshooting

If you encounter a problem while using Fast Clone, review the following list of previously reported issues and their solutions before contacting Informatica Global Customer Support.

### An ORA-942 "table not found" error occurs when I select an Oracle table owner.

The user that connects to the Oracle database lacks the permissions that Fast Clone requires. Use a user ID that has the required permissions. For more information, see "Setting Up an Oracle User for Connecting to Oracle" in the Fast Clone Installation Guide.

### When I run Fast Clone, the following error occurs:

```
The procedure entry point OCILobClose could not be located in the dynamic link library
OCI.dll
```

The most likely cause of this error is that the Oracle Client is not installed on the remote database server where Fast Clone runs. Download the Oracle Instant Client from the following Web site: http://www.oracle.com/technetwork/database/features/oci/index.html.

### During Fast Clone Console startup, the following error occurs:

```
"'java' is not recognized as an internal or external command, operable program or batch
file."
```

The Fast Clone Console is a Java application that requires the Java Runtime Environment (JRE) 1.5 and later. For Hadoop targets, the Fast Clone Console requires JRE 1.7 x64 or later. If you do not have a supported JRE version installed, download the JRE from the following web site: http://www.oracle.com/technetwork/java/javase/downloads/index.html. If you have a supported JRE version installed, this error might occur because the java.exe or java executable is not be specified in the PATH environment variable.

### The Fast Clone Console requires X Window on Linux or UNIX, but we are not allowed to install X Window on a Linux or UNIX system in our environment.

Install and run the Fast Clone Console on a Windows system that is remote from your Linux or UNIX server. To run an unload operation from the Windows system, complete the following steps:

1. Start gui.cmd on the remote Windows system or gui.sh on the remote Linux or UNIX system. Then create the configuration file.
2. On the **Runtime Settings** tab > **Files Locations** view, ensure that the **Output directory** value points to the Oracle source system where the data unload job will run.
3. Copy or FTP the configuration file to the Oracle source system where the data unload job will run.
4. Start unload.sh on the Oracle source machine.

### I downloaded Fast Clone for Sun Solaris. When I run unload.sh, the following error occurs:

```
FastReader: syntax error at line 1: `(' unexpected
```

The most likely cause of this error is that you have a Fast Clone version for Solaris that is not compatible with the CPU type. For example, you might have Fast Clone for SPARC III but the CPU type is SPARC II. Download the correct Fast Clone version for the CPU type.

To determine your CPU type, execute the following command:

```
/usr/platform/sun4u/sbin/prtdiag
```

### When I try to connect to Oracle while running the Fast Clone engine, I get an error that indicates the TNS listener could not resolve SERVICE_NAME given in connect descriptor. However, when I connect to Oracle from the Fast Clone Console, I do not get an error.

The Fast Clone Console uses a JDBC connection driver. With this driver, the Fast Clone Console first tries to use the Oracle SID and then the SERVICE_NAME to establish a connection. The Fast Clone executable uses the SERVICE_NAME by default to connect to the database.

The SERVICE_NAME is specified in a tnsnames.ora connection string as follows:

```
(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP)(HOST=hostname)(PORT=port)))
(CONNECT_DATA=(SERVER=DEDICATED)(SERVICE_NAME=instancename)))
```

To resolve this issue, try using the Oracle SID value. Select the **Use SID instead of SERVICE_NAME** option on the **Runtime Settings** tab > **Miscellaneous Conditions** view in the Fast Clone Console. Alternatively, specify the following parameter in the configuration file:

```
use_sid_instead_of_service=true
```

You can also configure Fast Clone to connect to the database by using the Oracle Bequeth protocol, without a TNS listener. Either set the parameter `instance=/` in the configuration file or select the **Connect Direct in Informatica Fast Clone Console** option on the **Source DB** tab.

### During startup of the Fast Clone Console on UNIX, the following error message is issued:

```
Exception in thread "main" java.lang.InternalError: Can't connect to X11 window server
using 'x.y.z.w' as the value of the DISPLAY variable.
```

The Fast Clone Console requires an X Window environment on Linux and UNIX systems. However, your X Window environment is not properly configured.

Contact your system administrator for assistance. Verify that you can run X Window programs.

The X Window System is a graphics system on UNIX that provides an infrastructure for displaying windowed graphics. It provides a public protocol by which client programs can query and update information on X servers.

### Fast Clone freezes when I try to unload data into a pipe instead of a flat file.

Fast Clone can write data to a pipe while a program or script is reading data from the pipe. If no program or script is reading data from the pipe, Fast Clone might appear to hang.

### An ORA-12528 error occurs when I run Fast Clone on an Oracle database server that uses Automatic Storage Management (ASM).

Add the ASM instance to your listener.ora configuration file. Use the following syntax:

```
(SID_DESC = (SID_NAME = +ASM) (ORACLE_HOME = ORACLE_HOME_PATH) )
```

Then restart the Oracle listener.

APPENDIX B

# Fast Clone Configuration File Parameters

After you save a configuration file in the Fast Clone Console, you can edit it in any plain text editor.

## Configuration File Overview

You can manually edit the parameters in a cloning configuration file after initially defining it in the Fast Clone Console.

Most parameters have default values and are optional. Review the parameter descriptions to determine if you want to include a parameter.

Ensure that you specify a parameter in the correction section of the configuration file. Use the following syntax:

```
parameter_name=value
```

## Source Database Parameters

In the [SOURCE] section of the configuration file, specify parameters for connecting to the Oracle source.

The following parameters are supported:

**dba_password**

A valid password for the specified dba_userid user. If you set the dba_userid parameter to a forward slash (/), Fast Clone ignores the password and uses operating system authentication to control connection to the source database.

**Important:** If the encrypt_database_passwords parameter is set to false, specify a clear text password. By default, the Fast Clone Console sets the encrypt_database_passwords parameter to true and writes an encrypted password.

**dba_userid**

A user name that has the authority to connect to the source database. Fast Clone uses this user connection to get table metadata and perform checkpoints before unloading the source data. If you specify a forward slash (/) in this parameter, Fast Clone ignores the password and uses operating system authentication.

**Important:** The user must have permissions that allow SELECT ANY DICTIONARY and ALTER SYSTEM operations on the source database. The slower alternative is to scan the system tablespace. For Fast Clone to perform a SELECT from dictionary, the source database must be running.

**host**

The host name or IP address of the computer where the source database runs.

Default value: localhost

**instance**

The Oracle instance name that Fast Clone uses to connect to the source database. If this parameter is not defined or specifies a forward slash (/), Fast Clone uses the ORACLE_SID environment variable to connect to an Oracle source with the Bequeath protocol. If the TWO_TASK environment variable is defined, Fast Clone uses the TWO_TASK environment variable.

**owner**

The name of the source schema owner for the tables to be unloaded. This owner name is also the schema name for the tables.

**port**

The port number that Fast Clone uses to connect to the source database.

Default value: 1521

**protocol**

The protocol that Fast Clone uses to connect to the source database. Valid values are:

- **TCP**. Use the TCP/IP protocol.
- **TCPS**. Use the TCP/IP protocol with the Secure Sockets Layer (SSL).

Default value: TCP

## ASM Connection Settings

Use the following parameters if the source database files are managed by Oracle Automatic Storage Management (ASM). Fast Clone connects to an ASM instance to unload data.

**asm_instance_defined**

Indicates whether Fast Clone connects to an ASM instance. Valid values are:

- **true**. Fast Clone connects to an ASM instance.
- **false**. Fast Clone connects to an Oracle source database without ASM.

Default value: false

**asm_instance_host**

The host name or IP address of the system with the ASM instance.

Default value: localhost

**asm_instance_name**

The ASM instance name to which Fast Clone connects to unload data.

Default value: No default value is provided.

**asm_instance_port**

The port number that is used to connect to the ASM instance that is specified in the asm_instance_name parameter.

Default value: 1521

**asm_protocol**

The protocol that Fast Clone uses to connect to an ASM instance. Valid values are:

- **TCP**. Use the TCP/IP protocol.
- **TCPS**. Use the TCP/IP protocol with the Secure Sockets Layer (SSL).

Default value: TCP

**asm_sys_password**

A password that is defined for the ASM SYS user.

**Important:** If the encrypt_database_passwords parameter is set to false, specify a clear text password. By default, the Fast Clone Console sets the encrypt_database_passwords parameter to true and writes an encrypted password.

**asm_sys_username**

The user name of the default Oracle ASM SYS user that was created at Oracle installation, or another user that you define. This user must have SYSDBA privileges.

Default value: SYS

# Source Table Parameter

In the [SOURCE_TABLES] section of the configuration file, specify the tables or materialized views to unload with the direct path unload method. In the [SOURCE_INDIRECT_TABLES] section, specify the tables or views to unload with the conventional path unload method.

In both sections, use the following parameter to identify the tables or views:

***table_id***

The name of a source table to unload. For partitioned tables, specify the table name followed by the partition name. Each *table_id* must be a unique key in the configuration file.

To unload data from selected partitions in a source table, specify each partition on a separate row. For example:

```
table1=table_1.partition_1
table2=table_1.partition_3
```

To unload data from all partitions in a source table, specify the table name only or list all partitions. For example:

```
table1=table_1
```

or

```
table1=table_1.partition_1
table2=table_1.partition_2
table3=table_1.partition_3
```

**Tip:** The data load script can load table partitions in parallel because Fast Clone creates control and data files for each partition.

# SQL Query Parameter

In the [SOURCE_INDIRECT_STATEMENTS] section of the configuration file, optionally enter one or more SQL query statements to be processed when an unload job runs.

Use the following parameter:

***query_id***

Defines an SQL query, such as a table join. You can specify the *query_id* parameters multiple times to define multiple SQL queries. Each *query_id* must be a unique key in the configuration file. Use the following syntax:

```
query_id=sql_statement
```

For example:

```
query1="SELECT * FROM table1 WHERE ROWNUM<10"
query2="SELECT * FROM table2 WHERE col_id>100"
```

Fast Clone writes the results of these queries to the query1.dat and query2.dat files.

# Source Column Parameters

In the [COLUMN_BIT_MASK], [SOURCE_INDIRECT_TABLES_COLUMNS_LIST], and [BIN_COLUMN_BIT_MASK] sections of the configuration file, list the source columns from which to unload data.

The [COLUMN_BIT_MASK] section uses the following parameter to list column bit masks for the tables that are specified in the [SOURCE_TABLES] section and unloaded with the direct path unload method:

***table_id*_column_bit_mask**

A hexadecimal bit mask that is used to select the columns for unload processing. The least significant bit indicates the first column. The most significant bit indicates the last column. Set a column bit in the bit mask to 1 to unload data from the column. Set a column bit in the bit mask to 0 to not process the column.

For example, a table that is identified by the table1 key in the [SOURCE_TABLES] section contains the following columns: COL1, COL2, COL3, COL4. To unload data from only COL4, set the binary bit mask for the table to 1000. The hexadecimal value of binary 1000 is 0x8:

```
table1_column_bit_mask=0x8
```

**Note:** If a table does not have a bit mask, Fast Clone unloads data from all columns.

If you set the column_list_clause_byname_filtering parameter to true in the [RUN] section of the configuration file, you must replace the bit mask values that the Fast Clone Console generates with comma-separated lists of column names in any order. For example:

```
table1_column_bit_mask=COL2,COL1,COL4
```

Fast Clone does not provide command line parameters to specify column bit masks. You can specify column bit masks only in a configuration file.

The [SOURCE_INDIRECT_TABLES_COLUMNS_LIST] section uses the following parameter to list columns for the tables that are specified in the [SOURCE_INDIRECT_TABLES] section and unloaded with the conventional path method:

***table_id*_column_list**

A comma-separated list of column names that identify the columns in a table from which to unload data. For example:

```
table1_column_list=COL2,COL1,COL4
```

Optionally, you can use SQL expressions to specify custom column formats.

- To specify a custom date or timestamp format, use the following statement syntax:

  ```
  to_char(COLUMN_NAME, 'custom_format') as COLUMN_NAME
  ```

  **Note:** This custom format overrides the date_format or timestamp_format parameter value. For information about Oracle datetime format, see the Oracle documentation.
- To specify a value to replace null values, use the following statement syntax:

  ```
  nvl(COLUMN_NAME, 'null_replacement_value') as COLUMN_NAME
  ```
- To specify a custom precision value for a NUMBER column, use the following statement syntax:

  ```
  round(NUMBER_COLUMN_NAME, custom_precision) as NUMBER_COLUMN_NAME
  ```
- To specify a custom column size to pad column values on the right with spaces when the length of a column value is less than the custom column size, use the following statement syntax:

  ```
  rpad(COLUMN_NAME, column_size, ' ') as COLUMN_NAME
  ```

For example, to use the custom date format dd/mm/yyyy for the COL4 column and replace null values in COL4 data with 14/12/2012, use the following ***table_id***_column_list statement:

```
table1_column_list=COL2,COL1,nvl(to_char(COL4, 'dd/mm/yyyy'), '14/12/2012') as COL4
```

The [BIN_COLUMN_BIT_MASK] section uses the following parameter to list column bit masks for the columns that are unloaded in binary format:

***table_id*_bin_column_bit_mask**

A hexadecimal bit mask that is used to select the columns to unload in binary format. The least significant bit indicates the first column. The most significant bit indicates the last column. Set a column bit in the bit mask to 1 to unload data from the column in binary format. For example:

```
table1_bin_column_bit_mask=0x8
```

### Related Topics:

- “ Custom Column Format Parameters” on page 119

# WHERE Condition Parameter

In the [WHERE_CONDITION] and [SOURCE_INDIRECT_TABLES_WHERE_CONDITION] sections of the configuration file, optionally specify WHERE conditions for the source tables.

The [WHERE_CONDITION] section defines WHERE conditions for the tables that are unloaded with the direct path unload method. The [SOURCE_INDIRECT_TABLES_WHERE_CONDITION] section defines WHERE conditions for the tables that are unloaded with the conventional path unload method.

In both sections, use the following parameter to define the WHERE condition:

***table_id*_where_condition**

Specifies a WHERE clause that is used to filter the row data to be unloaded from the table that is defined by the table_id key in the [SOURCE_TABLES] or [SOURCE_INDIRECT_TABLES] section of the cloning configuration file. For tables that do not have WHERE clause, Fast Clone unloads all rows. For more information about SQL operators for WHERE clauses, see the SQL documentation.

For example, the following statement specifies a WHERE clause for a DATE column:

```
table1_where_condition=( DATE_COLUMN >= to_date('12.14.2012 10:22:14', 'mm.dd.yyyy
hh24:mi:ss') )
```

**Note:** You can specify only one WHERE clause for a table_id table.

# Custom Column Format Parameters

You can customize date and timestamp formats for unload jobs that use the direct path or conventional path unload method

## Customizing Formats for Tables That Use the Direct Path Unload Method

In the [DATE_FORMATS] and [TIMESTAMP_FORMATS] sections of the cloning configuration file, you can define custom formats for DATE and TIMESTAMP columns. By default, the [DATE_FORMATS] and [TIMESTAMP_FORMATS] sections are empty. To add a custom format, use the following parameter:

***custom_format_id***

Specifies a custom DATE or TIMESTAMP format with a unique numeric ID. Ensure that each ID is a one- or two-digit number that is unique within the [DATE_FORMATS] or [TIMESTAMP_FORMATS] section of the configuration file. For information about valid Oracle datetime formats, see the Oracle documentation.

For example:

```
[DATE_FORMATS]
1=dd/mm/yyyy

[TIMESTAMP_FORMATS]
1=dd/mm/yyyy hh24:mi:ss.ff6
```

In the [*table_id*] section of the configuration file, assign the custom formats that you defined to columns in a table from which data will be unloaded. Use the following parameter:

***COLUMN_NAME***

Assigns a custom date or timestamp format to a column. You can also use this parameter to control numeric precision and padding and indicate a specific date or timestamp value that replaces nulls.

Use the following general syntax:

```
COLUMN_NAME=parameter_1:value_1;[parameter_n:value_n;]
```

**Important:** Ensure that you use all uppercase for the *COLUMN_NAME* value.

- To assign a custom date format, use a statement that begins with "d:" and specifies a *custom_format_id* that you defined for a date format:

  ```
  d:custom_format_id;
  ```

  **Note:** This custom format overrides the format that is specified by the date_format parameter.

- To assign a custom timestamp format, use a statement that begins with "t:" and specifies a *custom_format_id* that you defined for a timestamp format:

  ```
  t:custom_format_id;
  ```

  **Note:** This custom format overrides the format that is specified by the timestamp_format parameter.

- To assign a value that replaces null values, use a statement that begins with "n:" and specifies the replacement value:

  ```
  n:null_replacement_value;
  ```

  For DATE and TIMESTAMP columns, you can specify a *null_replacement_value* in any format. However, for consistency, use a custom format that is specified by the *custom_format_id* parameter.

  **Important:** Ensure that the suppress_trailing_nullcols parameter is set to false. If the suppress_trailing_nullcols parameter is set to true, Fast Clone does not unload trailing columns that contain null values, even if you specify a null replacement value.

- To assign a custom precision for NUMBER columns, use statement that beings with "p:" and specifies the precision value:

  ```
  p:custom_precision;
  ```

- To assign a custom padding value, use a statement that begins with "a:" and specifies the padding value:

  ```
  a:custom_padding;
  ```

For example:

```
[DATE_FORMATS]
1=dd/mm/yyyy

[TIMESTAMP_FORMATS]
1=dd/mm/yyyy hh24:mi:ss.ff6

[TABLE1]
TIMESTAMP_WITHOUT_TZ_COL=t:1;n:14/12/2012 00:00:00.0;
DATE_COL=d:1;
```

## Customizing Formats for Tables That Use the Conventional Path Unload Method

To specify custom formats for the columns in source tables that you unload with the conventional path method, use SQL expressions instead of column names that are specified by the *table_id*_column_list parameters in the [SOURCE_INDIRECT_TABLES_COLUMNS_LIST] section of the cloning configuration file.

- To specify a custom date or timestamp format, use the following statement syntax:

  ```
  to_char(COLUMN_NAME, 'custom_format') as COLUMN_NAME
  ```

  **Note:** This custom format overrides the date_format or timestamp_format parameter value. For information about Oracle datetime format, see the Oracle documentation.

- To specify a value to replace null values, use the following statement syntax:

  ```
  nvl(COLUMN_NAME, 'null_replacement_value') as COLUMN_NAME
  ```

- To specify a custom precision value for a NUMBER column, use the following statement syntax:

  ```
  round(NUMBER_COLUMN_NAME, custom_precision) as NUMBER_COLUMN_NAME
  ```

- To specify a custom column size to pad column values on the right with spaces when the length of a column value is less than the custom column size, use the following statement syntax:

  ```
  rpad(COLUMN_NAME, column_size, ' ') as COLUMN_NAME
  ```

For example, to use the custom date format dd/mm/yyyy for the COL4 column and replace null values in COL4 data with 14/12/2012, use the following *table_id*_column_list statement:

```
table1_column_list=COL2,COL1,nvl(to_char(COL4, 'dd/mm/yyyy'), '14/12/2012') as COL4
```

### Related Topics:

- "Source Column Parameters" on page 117

# Target Database Parameters

In the [DESTINATION] section of the configuration file, specify parameters for the target database. The output control files and the data load script use these parameters to load data to the target database.

The following parameters are supported:

**db_type**

The target database type. Options are:

- **cloudera**. Use this option for Cloudera targets.
- **flat**. Use this option for flat file targets.
- **greenplum3**. Use this option for Greenplum targets.
- **hive**. Use this option for Hive targets.
- **hortonworks**. Use this option for Hortonworks targets.
- **mapr**. Use this option for MapR targets.
- **mssql**. Use this option for Microsoft SQL Server targets.
- **netezza**. Use this option for Netezza targets.
- **oracle**. Use this option for Oracle targets.
- **teradata**. Use this option for Teradata targets.
- **vertica**. Use this option for Vertica targets.

Default value: oracle

**dba_password**

A valid password for the user that is specified by the dba_userid parameter. If you specify a forward slash (/) in the dba_userid parameter, Fast Clone ignores the password and uses operating system authentication.

**Important:** By default, the Fast Clone Console sets the encrypt_database_passwords parameter to true and writes an encrypted password. However, if the encrypt_database_passwords parameter is set to false, specify a clear text password.

**dba_userid**

The name of a user who has permissions to connect to the target database and load the source data. If you specify a forward slash (/) in this parameter, Fast Clone ignores the password and uses operating system authentication.

**greenplum_loader_home**

For Greenplum targets, if you load data to the target database from data files or pipes, the path to the Greenplum loader root directory that Fast Clone uses to locate the gpload utility and generate valid load scripts.

**host**

The host name or IP address of the system where the target database runs.

#### instance

The target instance name or database name for the tables to be loaded.

Default value: the value that is specified by the instance parameter in the [SOURCE] section of the cloning configuration file.

#### mssql_batchsize

The number of rows in a batch that is passed to the Microsoft SQL Server osql utility for loading data to the target. The osql utility copies each batch to the target server as one transaction.

Fast Clone uses this value as the BATCHSIZE option value in the BULK INSERT statement for the osql utility.

Range of values: 1 through 1000000

Default value: 10000

#### mssql_maxerrors

The maximum number of syntax errors that can appear in the output data files before the Microsoft SQL Server osql utility cancels the BULK INSERT operation. The osql utility ignores all of the rows that it cannot parse and counts an error for each row.

Fast Clone uses this value as the value of the MAXERRORS option in the BULK INSERT statement for the osql utility.

Default value: 100000

#### owner

The target schema name or schema owner for the tables to be loaded.

#### port

The port number that Fast Clone uses to connect to the target database.

Default value: No default value is provided.

**Note:** The default value that is specified by the Fast Clone Console depends on the target type.

#### protocol

For Oracle targets, the protocol that Fast Clone uses to connect to the target database. Valid values are:

- **TCP**. Use the TCP/IP protocol.
- **TCPS**. Use the TCP/IP protocol with the Secure Sockets Layer (SSL).

Default value: TCP

#### stream_maxerrors

For Greenplum, Netezza, and Teradata targets, the maximum number of errors that can occur when streaming data to the target database with DataStreamer. If the number of errors exceeds this maximum, DataStreamer ends with an error.

**Note:** For Netezza targets, review the .nzbad file to which DataStreamer writes the data rows that caused an error. DataStreamer does not report any errors in the output log if the number of rows that caused an error is less than the value of this parameter.

Range of values: 1 through 2000000000

Default value: 1000 for Greenplum and Netezza, 2 for Teradata

**textfile_charset_collation**

The name of a Microsoft SQL Server collation that is used to store character and Unicode data in the output data files. Fast Clone uses this value to correctly specify a column collation in the format file that maps the fields of the data file to the table columns. Fast Clone uses the specified collation for all of the table columns.

Fast Clone uses this value as the FORMATFILE option value in the BULK INSERT statement for the osql utility.

Default value: SQL_Latin1_General_Cp437_BIN

# Runtime Parameters

In the [RUN] and [INDIRECT] sections of the configuration file, specify runtime parameters.

In the [RUN] section, you can enter the following generic and direct path unload parameters:

**asm_debug**

Set this parameter to true only at the request of Informatica Global Customer Support. For Oracle sources in an Automatic Storage Management (ASM) environment, a value of true causes Fast Clone to print an ASM debug log to the asm_debug.log file.

Default value: false

**Note:** This parameter is not available in the Fast Clone Console. Edit the cloning configuration file to add this parameter manually.

**big_raw_device_offset_mul**

For Oracle sources that use raw devices on AIX, the number of offset blocks to skip from the beginning of a DF_LGDSK raw device when the calc_raw_device_offset parameter is set to true. You can specify the size of a single offset block in the raw_device_offset parameter. To get the total number of bytes that Fast Clone skips on DF_LGDSK raw devices, multiply the raw_device_offset value by the big_raw_device_offset_mul parameter value.

Valid values: 1 through 64

Default value: 32

**binary_length_prefix**

For Teradata targets, indicates whether to unload source data in binary format. Each record in binary format can include a 2-byte field that indicates the total length of the record, some optional bytes for null values, and the data.

For Vertica targets, indicates whether to unload source data in the NATIVE VARCHAR or text format. If you set this parameter to true, Fast Clone adds the NATIVE VARCHAR keyword to the COPY or LCOPY command in the output SQL load script. The database server converts CHAR or VARCHAR fields in the data files to Vertica datatypes when loading data. If you set this parameter to false, Fast Clone loads source data in text format with delimiters. Valid values are true and false.

Default value: false

### binary_length_prefix_indicators

For Teradata targets, if you set the binary_lengh_prefix parameter to true, indicates whether to set the indicator bits in the binary output for columns that contain null values. Valid values are:

- **true**. Set the indicator bits to 1 for null values.
- **false**. Do not set indicator bits.

Default value: false

**Note:** This parameter is not available in the Fast Clone Console. To use the parameter, manually enter it in the cloning configuration file.

### binary_length_prefix_little_endian

If you set the binary_length_prefix parameter to true, indicates whether to use LITTLE ENDIAN or BIG ENDIAN for the binary output. For Teradata targets, this parameter determines the format of the integer that specifies the length of records and fields in the binary files. Valid values are:

- **true**. The binary output uses the LITTLE ENDIAN format.
- **false**. The binary output uses the BIG ENDIAN format.

Default value: true

### blocks_per_lob_read

Specifies the number of Oracle data blocks that Fast Clone fetches from LOB columns per read operation.

**Important:** Change this parameter only if Fast Clone unload performance is slow. The value of this parameter depends on the extent size for locally managed tablespaces and the table-specific extent sizes for dictionary-managed tablespaces.

Valid values: 1 through 128

Default value: 4

### blocks_per_read

Specifies the number of Oracle data blocks that Fast Clone fetches per read operation.

**Important:** Change this parameter only if Fast Clone unload performance is slow. The value of this parameter depends on the extent size for locally managed tablespaces and the table-specific extent sizes for dictionary-managed tablespaces.

Valid values: 1 through 1024

Default value: 32

### calc_raw_device_offset

Indicates whether Fast Clone skips the offset block in raw devices. Valid values are:

- **true**. Skip the offset block in raw devices.
- **false**. Read data from the beginning of the raw device.

To get the total number of bytes that Fast Clone skips in a DF_LGDSK raw device, multiply the raw_device_offset value by the big_raw_device_offset_mul parameter value.

Default value: true

### characterset

For Teradata targets, indicates whether to convert the source character data to UTF-8. By default, Fast Clone preserves the source database encoding. Valid values are:

- **NONE**. Preserve the source character set.
- **UTF8**. Unload the data in the UTF-8 format.

Default value: NONE

### checkpoint_each_x_tables

Specifies the number of tables from which Fast Clone unloads data before taking a checkpoint.

Valid values: 0 through 100000

Default value: 0. With this default, Fast Clone takes a checkpoint only before it starts unloading source data.

### column_list_clause_byname_filtering

Indicates whether Fast Clone unloads columns in the custom order that is specified in the *TABLE_ID*_COLUMN_BIT_MASK parameter in the [COLUMN_BIT_MASK] section of the configuration file. Valid values are:

- **true**. Fast Clone unloads columns in the custom order that is specified in the *TABLE_ID*_COLUMN_BIT_MASK parameter.
- **false**. Fast Clone unloads columns in the order in which the columns appear in the source database.

Default value: false

### column_separator

Specifies a character or character sequence that Fast Clone uses to separate columns in the output data files. Specify a character or character sequence that does not appear in the source data. Otherwise, the target load utility cannot correctly parse the data files. For more information, see “Selecting Column and Row Separator and Enclosure Character to Use for Data Loading” on page 50.

You can specify a column separator that uses the \x*hh* escape sequence in hexadecimal notation.

Also, you can use the following non-printable ASCII characters as column separators:

- \a (alert)
- \b (backspace)
- \n (new line)
- \r (carriage return)
- \t (horizontal tab)
- \v (vertical tab)

If you use non-printable ASCII characters, ensure that the target load utility supports these characters as separators. For more information, see the target database documentation.

Default value: semicolon (;)

### compress_network_traffic

Indicates whether to compress network traffic. Valid values are:

- **true**. Compress network traffic.
- **false**. Do not compress network traffic.

Default value: false

**compress_on_the_fly**

Indicates whether to compress the output data files during unload processing to reduce the file sizes. Valid values are:

- **true**. Compress the output data files. The compression method is specified in the compression_type parameter.
- **false**. Do not compress the output data files.

Default value: false

**compression_level**

Specifies the compression level for the GZIP compression method. Higher values result in faster compression and a lower compression ratio. Lower values result in slower compression and a higher compression ratio. For Fast Clone to use this parameter, the compression_type parameter must be set to GZIP and the compress_on_the_fly parameter must be set to true.

Valid values: 1 through 9

Default value: 1

**compression_type**

If you set the compress_on_the_fly parameter to true, specifies the compression method that Fast Clone uses to compress the output data files. Valid values are:

- **GZIP**. Provides a better compression ratio but is slower. You can set the GZIP compression level in the compression_level parameter.
- **QZIP**. Provides a lower compression ratio but is faster.

Default value: GZIP

**consistent_lock_on_table_set**

If you use the direct path unload method and set the dirty parameter to false, indicates whether to lock all of the selected tables in a table set before unloading the source data. Valid values are:

- **true**. Fast Clone locks all of the selected source tables in a table set before unloading the source data. Fast Clone unlocks these tables after unloading data from all of them. Enter this value to get consistent point-in-time data across the selected source tables.
- **false**. Fast Clone holds a lock on each source table only while unloading data from the table.

Default value: false

**control_file_name_base**

Specifies the naming pattern that Fast Clone uses to generate names for control files, log files, and SQL scripts. You can use the following variables:

- %&schema&% - Represents a default schema name for the user that Fast Clone uses to connect to the source database.
- %&source_schema&% - Represents a source schema name.
- %&target_schema&% - Represents a target schema name.
- %&table&% - Represents a table name.
- %&table_counter&% - Represents a sequence number of a table.
- %&partition&% - Represents a partition name.
  The partition variable is required if you specify partitions in the [SOURCE_TABLES] or [SOURCE_INDIRECT_TABLES] sections of the configuration file.

- %&counter&% - Represents the sequence number of a control file for a table. This variable value is always 1.
- %&yyyy&% - Represents a year.
- %&mm&% - Represents a month.
- %&dd&% - Represents a day.

For example:

```
control_file_name_base=%&source_schema&%.%&table&%.%&partition&%
```

If you do not specify the control_file_name_base parameter, Fast Clone uses the table name as the name for control files, log files, and SQL scripts.

#### convert_ascii_to_ebcidic

Indicates whether Fast Clone converts source data to the IBM EBCIDIC encoding when unloading data to output data files and pipes. Valid values are:

- **true**. Convert source data to the IBM EBCIDIC encoding.
- **false**. Preserve the original encoding.

Default value: false

#### convert_numbers_to_comp3_fl_8b

Indicates whether to write numeric data to the output data files as COMP-3 values. Valid values are:

- **true**. Convert numeric data to COMP-3 in the output data files. Use this value if you need to read the output file with COBOL or another language that has built-in COMP-3 support.
- **false**. Preserve the original numeric format.

Default value: false

#### convert_numbers_to_comp3_implied_decimal

The position of the decimal point in the COMP-3 format.

Valid values: 0 through 14

Default value: 0

#### convert_utf16_clobs_to_utf8

Indicates whether Fast Clone converts CLOB data from UTF-16 to UTF-8 encoding in the output data files. Valid values are:

- **true**. Convert source character data in CLOB columns from UTF-16 to UTF-8 encoding.
- **false**. Do not convert source character data in CLOB columns. If you do not use UTF-16 encoding for the source data, set this parameter to false to avoid the overhead of character set conversion.

Default value: false

#### convert_utf16_nchars_to_utf8

Indicates whether Fast Clone converts NCHAR data from UTF-16 to UTF-8 encoding in the output data files. Valid values are:

- **true**. Convert source character data in NCHAR columns from UTF-16 to UTF-8 encoding.
- **false**. Do not convert source character data in NCHAR columns.

**Note:** If you do not use UTF-16 encoding for the source data, set this parameter to false to avoid the overhead of character set conversion. To use the parameter, you must manually enter it at the command line or in the cloning configuration file. You cannot enter it in the Fast Clone Console.

Default value: false

### convert_utf16_nclobs_to_utf8

Indicates whether Fast Clone converts NCLOB data from UTF-16 to UTF-8 encoding in the output data files. Valid values are:

- **true**. Convert source character data in NCLOB columns from UTF-16 to UTF-8 encoding.
- **false**. Do not convert source character data in NCLOB columns.

**Note:** If you do not use UTF-16 encoding for the source data, set this parameter to false to avoid the overhead of character set conversion. To use the parameter, you must manually enter it at the command line or in the cloning configuration file. You cannot enter it in the Fast Clone Console.

Default value: false

### create_external_table_log_file

For Oracle targets, if you create an SQL script for creating external tables based on the output files, indicates whether to log messages related to this processing to a log file. Valid values are:

- **true**. The Oracle external table utility logs messages to a log file when accessing data in the output data files. Ensure that the create_external_table_script parameter is also set to true.
- **false**. The Oracle external table utility does not log these messages.

Default value: false

### create_external_table_name_includes_partition

Indicates whether Oracle external table names include partition names. Valid values are:

- **true**. Add the partition name to the external table name in the following format:

  ```
  table_partition
  ```
- **false**. Do not add the partition name to the external table name.

Default value: false

### create_external_table_script

For Oracle targets, indicates whether Fast Clone generates an SQL script to create Oracle external tables based on the output data files. Valid values are:

- **true**. Create an SQL script to create Oracle external tables based on the output data files.
- **false**. Do not create an SQL script to create Oracle external tables based on the output data files.

Default value: false

### create_load_input_only

Indicates whether Fast Clone creates the load script and control files but skips unloading data from the source database. Valid values are:

- **true**. Create a load script and the control files. Do not unload data from the source database.
- **false**. Create a load script and the control files. Unload data from the source database.

**Note:** Fast Clone does not generate load scripts and control files for flat file, Hadoop, and Hive targets.

Default value: false

### create_output_pipe_instead_of_file

Indicates whether Fast Clone creates pipes in the output directory and writes the output data to these pipes. Valid values are:

- **true**. Fast Clone creates pipes in the output directory.
- **false**. Fast Clone does not create pipes in the output directory. Use this value to unload data to output data files or to the pipes that you manually created.

Default value: false

### date_format

Specifies the date format to use in output files. For information about datetime format elements, see the Oracle documentation. For example, the following format values are valid:

- YYYY-MM-DD HH24:MI:SS
- DDMMYYYY
- YYYY-MM-DD
- HH24:MI:SS

Default value: YYYY-MM-DD HH24:MI:SS

### dbsync_config_xml

Specifies the path and file name for the Data Replication configuration file that can be in XML or SQLite format. The path is relative to the DBSYNC_HOME directory.

**Important:** If you use a Data Replication configuration in SQLite format, do not copy or move the configuration files from the default *DataReplication_installation*/configs directory to another location.

Default value: config.xml

### dbsync_config_xsd

Specifies the path and file name for the Data Replication configuration schema .XSD file. The path is relative to the DBSYNC_HOME directory.

Default value: config.xsd

### dbsync_integration

Indicates whether Fast Clone integrates with Informatica Data Replication. Also, when integration is enabled, determines whether Fast Clone allows user sessions that are connected to the source tables to complete processing before locking the tables. Valid values are:

- **false**. Do not integrate Fast Clone with Data Replication.
- **lock**. Integrate Fast Clone with Data Replication. Fast Clone waits for all user sessions that are connected to the source tables to complete processing and close before locking the tables for unload processing.
- **lock_disconnect**. Integrate Fast Clone with Data Replication. Fast Clone forces all user sessions that are connected to the source tables to disconnect and then locks the tables for unload processing.

Default value: false

### dbsync_integration_by_direct

If you enable integration with Data Replication, indicates whether to unload data from the source database with the direct path unload method. Valid values are:

- **true**. Use the direct path unload method.
- **false**. Use the conventional path unload method.

Default value: true

#### dbsync_integration_update_scn

Indicates whether to update the Data Replication configuration file with the SCN for each mapped table after Fast Clone unloads the source data. Valid values are true and false.

**Important:** Set this parameter to true if you plan to use Data Replication to replicate change data and use Fast Clone to perform initial materialization of the replication target tables.

Default value: true

#### decimal_separator

Specifies the decimal separator that Fast Clone uses in decimal numbers in unloaded data.

Default value: period (.)

#### default_long_raw_is_hex

For Oracle targets, indicates whether to unload BLOB and LONG RAW source data in hexadecimal format. Set this parameter to true to unload data with these datatypes in hexadecimal format. For other target types, set this parameter to false to avoid generating incorrect control files.

Default value: false

#### destination_loader_to_use

For Teradata and Vertica targets, specifies the load utility or command that Fast Clone uses to load data to the target. Enter a load utility or command that Fast Clone supports.

Valid values for Teradata loaders:

- **teradata_fast_load**
- **teradata_multi_load**
- **teradata_parallel_dpump**

Valid values for Vertica loaders:

- **isql_copy**. Use this command to load data from the data files that are located on the target database server.
- **isql_lcopy**. Use this command to load data from the data files that are located on the system where you run the target load utility.

Default value: teradata_fast_load

#### direct_select_from_sys

Indicates whether Fast Clone uses SYS-owned dictionary tables, such as sys.obj$ and sys.tab$, to get extent boundaries. Valid values are:

- **true**. Use SYS-owned dictionary tables to get extent boundaries.
- **false**. Use dba_extents and dba_segments views to get extent boundaries.

**Note:** If you set this parameter to false for Oracle sources that use locally managed tablespaces, retrieval of extent information from the dba_extents view is slow, especially if the dictionary is partially analyzed and the optimizer mode is not RULE. In this situation, Informatica recommends that you set this parameter to true and grant the SELECT ANY DICTIONARY permission to the user.

Default value: false

### dirty

Indicates whether to lock the tables that are unloaded with the direct path unload method to perform consistent read operations. Valid values are:

- **true**. Do not lock the tables before unloading the source data. Use this option if you want to bypass the Oracle consistency mechanism. However, block-level consistency is preserved, because block I/O operations are atomic.
- **false**. Lock the tables exclusively before unloading the source data.

Default value: true

### disable_linux_cache_bypass

For Linux operating systems, indicates whether to use direct I/O mode to read Oracle data files. In direct I/O mode, Fast Clone bypasses operating system caching and reads data directly from disk. Set this parameter to true to use direct I/O mode.

Default value: false

### enclose_dates

Indicates whether to use the character that is specified by the optionally_enclosed_by parameter to enclose, or delimit, dates. Set this parameter to true to use this character.

Default value: false

### enclose_numbers

Indicates whether to use the character that is specified by the optionally_enclosed_by parameter to enclose, or delimit, numbers. Set this parameter to true to use this character.

Default value: false

### encrypt_database_passwords

Indicates whether the password that is specified by the dba_password parameter in the [SOURCE] section of the configuration file is encrypted. Valid values are:

- **true**. The password is encrypted.
- **false**. The password is not encrypted.

Default value: true

### escape_enclosure

Specifies an escape character that Fast Clone inserts before the character that is specified in the optionally_enclosed_by parameter when the character appears in text columns. If you define this parameter, Fast Clone truncates the optionally_enclosed_by value to a single character.

Default value: No default is provided.

### explicitly_flush_buffer_cache

Indicates whether Fast Clone clears all data from the buffer cache of the Oracle source database before unloading the data with the direct path unload method. Valid values are:

- **true**. Fast Clone clears all data from the buffer cache by executing the following command on the Oracle source database:

  ```
  ALTER SYSTEM FLUSH BUFFER_CACHE;
  ```
- **false**. Fast Clone does not clear data from the buffer cache.

Default value: false

### export_all_tables_for_owner

Indicates whether to unload data from all tables in the specified schema or from only the tables that you specify. If you choose to unload data from all tables in the schema, indicates whether to use the direct path unload or conventional path unload method.

Valid values:

- **NONE**. Fast Clone unloads data from only the source tables that you specified in the [SOURCE_TABLES] or [SOURCE_INDIRECT_TABLES] section of the configuration file. You must specify at least one table in one of these sections.
- **AUTO**. Fast Clone unloads data from all tables in the specified schema by using the direct path unload method if possible. If Fast Clone cannot use the direct path unload method, Fast Clone unloads data by using the conventional path unload method.
- **ALL_DIRECT**. Fast Clone unloads all tables in the specified schema by the using direct path unload method.
- **ALL_CONVENTIONAL**. Fast Clone unloads all tables in the specified schema by using the conventional path unload.

Default value: NONE

### export_binary_to_separate_files

Indicates whether to write data from BLOB, CLOB, NCLOB, LONG, and LONG RAW source columns to separate files. In the regular data files, Fast Clone includes the file name instead of the source value. Specify true to write data from columns with these datatypes to separate files. For Hive targets, Fast Clone always uses the parameter value of false.

Default value: false

**Important:** If you use DataStreamer to load source data to Netezza targets, set this parameter to false.

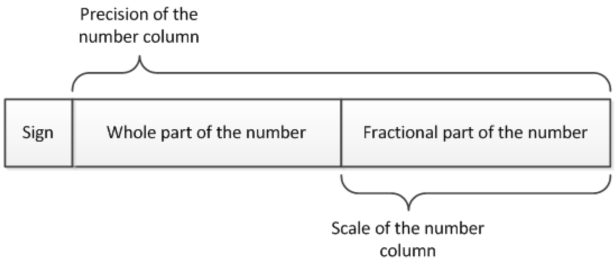
### fixed_size_characters_by_col_definition

Indicates whether to unload data from character columns in fixed-length format. The fixed length is determined by the column length. If a text value is shorter than the column length, Fast Clone pads the column with spaces up to the full column length. Specify true to unload character data in fixed-length format.

Default value: false

### fixed_size_number_by_col_definition

Indicates whether to unload data from NUMBER columns in the fixed-length format. Specify true to unload character data in fixed-length format. The fixed length is defined by the precision of the NUMBER column, without a decimal separator. If a number is shorter than this length, Fast Clone pads the value with zeroes. The following image shows the fixed-length format of a NUMBER column:


Precision of the number column
Sign
Whole part of the number
Fractional part of the number
Scale of the number column

For example, when using fixed-length format, Fast Clone unloads the value 123.45 from the NUMBER(10,5) column as +0012345000.

Default value: false

#### global_date_nvl

Specifies a date value that Fast Clone inserts instead of null values in DATE columns.

**Note:** You can specify a date value in any format. However, for consistency, use the format that is specified by the date_format parameter.

Default value: No default is provided.

#### greenplum_direct_data_stream

For Greenplum targets, indicates whether DataStreamer uses the Greenplum gpfdist utility to load data to the target database. Valid values are true and false.

Default value: The value of the direct_data_stream parameter.

#### greenplum_gpfdist_connection_timeout

For Greenplum targets, specifies the time interval, in seconds, that Fast Clone waits for a response from Greenplum when opening a connection to the target. If this interval elapses without a response, Fast Clone times out.

Valid values: 0 through 1000000

Default value: 55

#### greenplum_gpfdist_current_hostname

For Greenplum targets, specifies the host name or IP address of the system where you run the Fast Clone instance that starts the gpfdist utility for loading data.

Default value: localhost

#### greenplum_gpfdist_listen_hostname

For Greenplum targets, specifies the IP addresses on which the gpfdist utility listens for incoming requests. Use 0.0.0.0 to have the gpfdist utility listen for requests from all interfaces.

Default value: No default is provided.

#### greenplum_gpfdist_listen_port

For Greenplum targets, specifies the listener port for the gpfdist utility. The Greenplum target uses this port to submit requests to the gpfdist utility to load data.

Valid values: 0 through 65536

Default value: 8000

#### ip_instead_of_file

If the use_ip_instead_of_file parameter is set to true, specifies the IP address or host name and the port number of the Fast Clone Server to which Fast Clone sends output files. Use the following format for the value:

```
host_or_ipaddress:port
```

Default value: No default is provided.

#### limit_result_set

Specifies the maximum number of rows that Fast Clone unloads from each table. Set this parameter to -1 to unload all rows from a table.

Valid values: -1 through 1073741823

Default value: -1

### load_data_to_same_partition_as_source

For Oracle targets, indicates whether SQL*Loader loads data to a table partition that has the same name as the corresponding source table partition. Specify true to load data from a source table partition to a target table partition of the same name.

Default value: false

### load_script_base_name

Specifies the file name of the load script. On Windows, Fast Clone generates load script files with the .cmd extension. On Linux and UNIX, Fast Clone generates load script files with the .sh extension.

Default value: The source schema owner name that is specified by the owner parameter in the [SOURCE] section of the cloning configuration file.

### local_time_zone

For Teradata targets, specifies the local time zone to use for unloading data from TIMESTAMP WITH LOCAL TIME ZONE columns. Specify the time zone as an offset from the Oracle database time zone:

```
"+/-HH:MM"
```

The *HH* variable is a two-digit number from -13 to +13. The *MM* variable is a two-digit number from 0 to 59.

For example, enter "+08:00".

Default value: No default is provided.

### log

Indicates whether Fast Clone produces an extended log. Set this parameter to true only at the request of Informatica Global Customer Support.

Default value: false

### mb_per_file

Defines the maximum amount of data, in megabytes, that a single output data file (.dat file) can contain.

Valid values: -1 through 65530

Default value: -1. With this default value, Fast Clone does not enforce a limit on the size of the output data files.

### netezza_control_chars_enabled

If you use DataStreamer to load data to a Netezza target, specifies the value of the EscapeChar option for the Netezza external tables. This option indicates whether the control characters, such as delimiter and backslash characters, are escaped in the data fields of the external tables. The Netezza external tables use a backslash as an escape character.

Default value: true

### netezza_direct_data_stream

Indicates whether to use DataStreamer to load data to a Netezza target. Options are:

- **true**. Select this option to stream data directly to the Netezza target by using the Netezza external tables. Fast Clone unloads data to the named pipes that represent the Netezza external tables. DataStreamer uses bulk Insert to load data from the external tables to the corresponding target tables.
- **false**. Select this option if you unload data to the output files or pipes. You can then load data from these files or pipes to the Netezza target by using the nzload utility.

Default value: false

#### netezza_error_log_directory

If you use DataStreamer to load data to a Netezza target, specifies the directory in which DataStreamer creates the following Netezza log files:

- *table_name*.*schema_name*.nzlog. This log file includes load statistics and diagnostic messages that the Netezza ODBC driver issues when loading data to the target tables from the corresponding external tables.
- *table_name*.*schema_name*.nzbad. This log file includes the external table rows that DataStreamer could not load to the corresponding target table.

Default value: The output directory

#### netezza_pipe_directory

If you use DataStreamer to load data to a Netezza target, specifies the directory in which Fast Clone creates named pipes that represent Netezza external tables.

On Windows, DataStreamer ignores this parameter because the named pipes are created in the named pipe directory that is mounted under the special path \\.\pipe\.

Default value: The output directory

#### netezza_quoted_value

If you use DataStreamer to load data to a Netezza target, specifies the value of the QuotedValue option for the Netezza external tables. DataStreamer requires this option to build correct Insert statements when loading data to the target tables from the corresponding external tables. Options are:

- **single**. Indicates that the data values in the external tables are enclosed with single quotation marks.
- **double**. Indicates that the data values in the external tables are enclosed with double quotation marks.
- **no**. Indicates that the data values in the external tables are not quoted.

Ensure that the quotation character that you specify for the load operation matches the quotation character that you specify for the unload operation. For unload jobs, Fast Clone uses the quotation character that you specify in the **Enclosed by** filed on the **Runtime Settings** tab > **Format Settings** view.

Default value: no

#### nvl_fixed_size_characters_by_space

For character columns that are unloaded in fixed-length format, indicates whether to replace null values with space characters. Specify true to replace nulls with spaces.

Default value: false

#### ocfs_direct_io

For Oracle sources in an Oracle RAC and Oracle Cluster File System (OCFS) environment on Linux or UNIX, indicates whether Fast Clone uses direct I/O mode to read Oracle data files. Direct I/O mode minimizes the effects of cache on reading data from Oracle data files. Set this parameter to true if the Oracle data files are located in an OCFS.

Default value: false

#### optionally_enclosed_by

Specifies the character or character sequence that Fast Clone uses to enclose character data in output data files. Specify a character or character sequence that does not appear in the source data. The target load utility cannot correctly parse the data files if the enclosure character is a source data value.

If you use the default value and the source columns contain double-quotation marks (") in the data, specify a different character or character string for this parameter or use the escape character in the escape_enclosure parameter to escape the double-quotation marks in the column data.

**Note:** If you use the character that is defined by the escape_enclosure parameter to escape the optionally_enclosed_by value in the source data, Fast Clone truncates the optionally_enclosed_by value to a single character.

Default value: Double-quotation mark (")

#### outdir

Specifies the directory in which Fast Clone creates the output files. The output directory can be on a local file system or on a network mapped share, such as NFS or Samba. For Hive targets, specifies the location of flat files that store table data on the HDFS.

Default value: The FAST_READER_HOME environment variable value.

#### output_file_extension_base

Specifies the extension that Fast Clone uses for output data files. Fast Clone creates a data file for each source table from which data is unloaded.

Default value: dat

#### output_file_name_base

Specifies the naming pattern that Fast Clone uses to create the base name of the output data files. In the naming pattern, you can include the following variables:

- %&schema&% - Represents a default schema name for the user that Fast Clone uses to connect to the source database.
- %&source_schema&% - Represents a source schema name.
- %&target_schema&% - Represents a target schema name.
- %&table&% - Represents a table name.
- %&table_counter&% - Represents a sequence number of a table.
- %&partition&% - Represents a table partition name. This partition variable is required if you specify partitions in the [SOURCE_TABLES] or [SOURCE_INDIRECT_TABLES] sections of the configuration file.
- %&counter&% - Represents a sequence number of a data file for a table. This counter variable is required if Fast Clone splits the output data files based on the mb_per_file and split_files_by_row_count parameters.
- %&yyyy&% - Represents a year.
- %&mm&% - Represents a month.
- %&dd&% - Represents a day.

For example:

```
output_file_name_base=%&source_schema&%.%&table&%.%&partition&%
```

If you do not specify the output_file_name_base parameter, Fast Clone uses the table name as the name for output data files.

#### output_ocfs_direct_io

For Oracle sources in an Oracle RAC and Cluster File System (OCFS) environment on Linux or UNIX, indicates whether Fast Clone uses direct I/O mode to write output files. Direct I/O mode minimizes the effects of cache on writing the output files. Set this parameter to true if you configured Fast Clone to generate output files in an OCFS.

Default value: false

**print_column_names**

For flat file targets, indicates whether Fast Clone writes column names in the first row of the output data files. Specify true to write column names in the first row.

Default value: false

**Note:** This parameter is available for all target types when the unload_data_only parameter is set to true.

**raw_device_offset**

The size of a single offset block, in bytes, in a raw device. Fast Clone skips the offset block if you set the calc_raw_device_offset parameter to true. To get the total number of bytes that Fast Clone skips on DF_LGDSK raw devices, multiply the raw_device_offset value by the big_raw_device_offset_mul parameter value.

Default value: 0 for Linux, 4096 for other operating systems

**record_separator**

A character or character sequence that Fast Clone uses to separate table rows in the output data files. Specify a character or character sequence that does not appear in the source data. Otherwise, the target load utility cannot correctly parse the data files. For more information, see "Selecting Column and Row Separator and Enclosure Character to Use for Data Loading" on page 50.

You can specify a row separator that uses the \x*hh* escape sequence in hexadecimal notation.

Also, you can use the following non-printable ASCII characters:

- \a (alert)
- \b (backspace)
- \n (new line)
- \r (carriage return)
- \t (horizontal tab)
- \v (vertical tab)

If you use non-printable ASCII characters in row separators, ensure that the target load utility supports these separators. For more information, see target database documentation.

Default value: \n

**replace_new_line_with_space**

Indicates whether to replace new line (\n) character in unloaded data with space characters. Specify true if you use the new line character as the row separator in output files and the source character data includes new line characters. However, with a value of true, unload processing performance might be degraded.

Default value: false

**right_pad_columns**

Indicates whether to pad columns on the right with spaces when the length of a column value is less than the full length of the column. Specify true to enable padding.

Default value: false

**show_increments_rows**

The number of rows that Fast Clone unloads with the direct path unload method before writing updated progress information for the unload job to the system console or command prompt window.

Valid values: 10 through 10000000

Default value: 50000

### skip_all_out_file_ops

For Greenplum and Teradata targets, indicates whether Fast Clone skips the writing of output files. Set this parameter to true if you use DataStreamer or if you need to troubleshoot jobs that unload a large volume of data. Valid values are true and false.

Default value: The value that is specified in the greenplum_direct_data_stream or teradata_direct_data_streamparameter.

### skip_unreadable_block

Indicates whether Fast Clone skips blocks that it cannot read from Oracle files when unloading data with the direct path unload method. Valid values are:

- **true**. Fast Clone skips the unreadable blocks and continues unload processing.
- **false**. Fast Clone ends with an error.

Default value: true

### split_files_by_row_count

The maximum number of rows that Fast Clone writes to an output data file. If Fast Clone needs to unload a greater number of rows from the source table, Fast Clone creates multiple output data files for the table. The file names have the following format:

```
table_name_file_number.dat
```

Specify this parameter only if you use the direct path unload method.

**Note:** For parallel unload processing, splitting output into multiple data files based on a number of rows is slower than splitting output based on a chunk size. This performance degradation occurs because of locks between the parallel threads.

Valid values: -1 through 1000000000

Default value: -1. With this default value, Fast Clone does not create multiple output data files based on row count.

### suppress_trailing_nullcols

Indicates whether Fast Clone includes trailing columns that contain null values in the output data files. Valid values are:

- **true**. Do not include trailing columns that contain null values.
- **false**. Include trailing columns that contain null values.

Default value: true

Fast Clone forces this parameter to false if the trailing_column_separator parameter is set to false.

**Important:** If you use DataStreamer to load data to a Netezza target, set this parameter to false.

### teradata_direct_data_stream

For Teradata targets, indicates whether DataStreamer sends the source data directly to Teradata Parallel Transporter (TPT) for loading to the target database. Valid values are:

- **true**. Do not create output data files. Send the source data directly to the TPT utility.
- **false**. Unload the source data to the output data files.

Default value: The value that is specified by the direct_data_stream parameter.

### teradata_downgrade_to_multiload_if_table_not_empty

For Teradata targets, indicates whether DataStreamer tries to use the MultiLoad utility if the FastLoad utility fails to load source data to target tables that are not empty. Valid values are:

- **true**. DataStreamer tries to use the MultiLoad utility if FastLoad returns an error.
- **false**. DataStreamer does not try to use the MultiLoad utility if FastLoad returns an error.

Default value: true

### teradata_drop_rebuild_tables

For Teradata targets, indicates whether DataStreamer drops and re-creates tables in the target database before it loads the source data. Valid values are true and false.

Default: false

### teradata_load_sessions

For Teradata targets, specifies the number of sessions that the Teradata FastLoad or MultiLoad utility can open to load the source data to each target table.

Default value: 2

### teradata_loader_to_use

For Teradata targets, the loader that Fast Clone uses to load the source data to the target. Valid values are:

- **teradata_fast_load**
- **teradata_multi_load**
- **teradata_parallel_dpump**

Default value: teradata_fast_load

**Note:** Deprecated parameter. Use the destination_loader_to_use parameter.

### threads_num

Specifies the maximum number of tables or table partitions that Fast Clone can unload in parallel with the direct path unload method.

Valid values: 1 through 64

Default value: 1

### threads_per_segment_num

Specifies the maximum number of threads that Fast Clone can use to unload data from each table or table partition with the direct path unload method.

**Note:** Fast Clone might use fewer threads. Fast Clone does not create a thread to unload a chunk of less than 100 data blocks.

Valid values: 1 through 64

Default value: 1

### timestamp_format

Specifies the timestamp format in the output. For more information on datetime format elements, see the Oracle documentation. For example:

- YYYY-MM-DD HH24:MI:SS.FF6
- YYYY-MM-DD HH24:MI:SS

- DDMMYYYY
- YYYY-MM-DD
- HH24:MI:SS

**Note:** For the conventional path unload method, Fast Clone supports all timestamp formats.

Default value: YYYY-MM-DD HH24:MI:SS.FF6

#### trailing_column_separator

Indicates whether to add a column separator after the last column value in each row in the output. Valid values are true and false.

**Note:** Fast Clone forces this parameter to false for Oracle targets if the unloaded tables contain LOB columns and the export_binary_to_separate_file parameter is set to true.

Default value: true

**Important:** If you use DataStreamer to load data to a Netezza target, set this parameter to false.

#### truncate_numeric_precision

Specifies the number of significant digits to use for rounding down the source numeric values in the output. This parameter truncates only the fractional part of a number.

Valid values: -1 through 40

Default value: **-1**. Fast Clone does not round down the source numeric values.

#### unload_data_only

Indicates whether to unload source data only and skip the creation of control files and the load script. Valid values are:

- **true**. Do not create control files and a data load script when Fast Clone unloads source data to .dat files.
- **false**. Create control files and a data load script when Fast Clone unloads source data to .dat files.

Default value: false

For flat file, Hadoop, and Hive targets, Fast Clone always uses the parameter value of true.

#### use_asm_global_cache

For Oracle ASM sources, indicates whether Fast Clone caches ASM AUNs in memory. Valid values are true and false.

Default value: true

#### use_dbms_diskgroup_instead_of_direct

Indicates whether Fast Clone uses the DBMS_DISKGROUP package to read data from ASM sources. Typically, use of the DBMS_DISKGROUP package is significantly slower than reading data directly from ASM sources because the package has a PL/SQL limit of 32,600 bytes per read.

However, if you use the conventional path method and set this parameter to true, Fast Clone unloads data from ASM by using multiple threads. Fast Clone can use multiple threads from a remote computer, with an even data distribution across the threads.

Valid values are true and false.

Default value: false

### use_ip_instead_of_file

Indicates whether Fast Clone sends the source data to the Fast Clone Server that is defined by the ip_instead_of_file parameter. Valid values are true and false.

Default value: false

### use_sid_instead_of_service

Indicates whether to use the Oracle SID instead of the service name when you cannot connect to the source instance by using the instance name. Valid values are true and false.

Default value: false

### use_teradata_parallel_transporter

For Teradata targets, indicates whether Fast Clone uses Teradata Parallel Transporter to load the source data to the target database. Valid values are true and false.

Default value: false

### validate_block_scn

Determines whether to write a warning message to a log file when the SCN value of an unloaded data block is greater than the Start SCN value. The Start SCN value corresponds to point immediately after the checkpoint when Fast Clone started unloading data. Fast Clone compares these values to detect possible data inconsistencies in the unloaded data. Valid values are:

- **true** . Write a warning message to a log file when a block SCN value is greater than the Start SCN value.
- **false**. Do not write the warning message.

Default value: false

**Note:** To use the parameter, you must manually enter it in the cloning configuration file. You cannot enter it at the command line or in the Fast Clone Console.

In the [INDIRECT] section of the configuration file, you can enter the following runtime parameters for the conventional path method:

### blob_fetch_chunk_size_for_indirect_kb

Specifies the amount of data that Fast Clone can unload from LOB columns at one time.

Valid values: 1 through 4096

Default value: 128

### enforce_oci9_limitations

Indicates whether Fast Clone unloads a single row at a time when a source table contains RAW columns. Valid values are:

- **true**. Force the fetch_array_size_for_indirect parameter to 1 if a source table contains RAW columns.
- **false**. Always use the fetch_array_size_for_indirect parameter.

Default value: true

### fetch_array_size_for_indirect

Specifies the number of rows that Fast Clone can unload from the Oracle source database at one time.

Valid values: 1 through 50000

Default value: 500

**fetch_blob_as_single_chunk_for_indirect**

Indicates whether to unload a single chunk of LOB data that has the size specified by the fetch_blob_as_single_chunk_for_indirect_kb parameter. Valid values are:

- **true**. Unload only a single chunk of LOB data.
- **false**. Unload all LOB data in multiple chunks.

Default value: false

**fetch_blob_as_single_chunk_for_indirect_kb**

Specifies the maximum amount of data that Fast Clone unloads from a LOB column if the fetch_blob_as_single_chunk_for_indirect parameter is set to true.

Valid values: 64 through 2097151

Default value: 128

**oracle_use_direct_file_access_to_fetch_extent_map**

Indicates whether to read extent information for parallel unload processing directly from the data file. Specify true to read extent information from the data file.

Default value: false

**Note:** In the Fast Clone Console, the corresponding **Use direct file access to fetch extent map** option is selected (true) by default.

**statement_unload_threads_num**

Specifies the maximum number of tables or table partitions that Fast Clone can unload in parallel with the conventional path unload method.

Valid values: 1 through 64

Default value: 1

**statement_unload_threads_per_segment_num**

Specifies the maximum number of threads that Fast Clone can use to unload data from each table or partition with the conventional path method.

**Note:** If you use multiple threads for a single table, CPU usage might increase. Usually, multiple threads are used to unload data from large tables. Fast Clone might use fewer threads than this parameter defines because there no reason to create a thread to export a chunk of less than 100 data blocks.

Valid values: 1 through 64

Default value: 1

# Character Replacement Parameter

In the [CHARACTER_REPLACEMENT] section of the configuration file, specify character replacement rules.

By default the [CHARACTER_REPLACEMENT] section is empty. You can use the following parameter to define a replacement rule:

***replacement_rule_id***

Ensure that each *replacement_rule_id* value in the [CHARACTER_REPLACEMENT] section is unique. For the value, use the following syntax: `character_to_be_replaced_new_character`

For example:

```
replace1="a_b"
```

# Reporting Parameters

In the [REPORTING] section of the configuration file, specify reporting parameters.

The following parameters are supported:

**create_run_report**

Indicates whether to create an XML report file in the output directory. The report file contains the number of unloaded rows and unload failures for each table. Valid values are true and false.

Default value: false

**job_name**

Specifies the job name that Fast Clone uses as a base for the XML report file name. Fast Clone also uses the job name in the SNMP trap text.

Default value: FastReader_Extract

**Note:** The Fast Clone Console sets this parameter to an empty string.

**manual_report_file_name**

Indicates whether to use the path and file name that are specified by the report_file parameter to generate the XML report file. Valid values are:

- **true**. Use the path and file name that are specified by the report_file parameter.
- **false**. Use the file name base that is specified by the job_name parameter.

Default value: false

**report_file**

Specifies the path and file name to use for the XML report file.

**report_on_begin**

Indicates whether to send an SNMP trap to an SNMP manager when Fast Clone starts unloading the source data. Valid values are true and false.

Default value: false

**report_on_end**

Indicates whether to send an SNMP trap to an SNMP manager when Fast Clone finishes unloading the source data. Valid values are true and false.

Default value: false

**send_snmp_trap**

Indicates whether to send SNMP traps. Valid values are true and false.

Default value: false

**snmp_community**

Specifies the Fast Clone SNMP community to which Fast Clone sends SNMP traps.

Default value: Informatica

**snmp_enterprise_oid**

Specifies the Fast Clone enterprise object identifier (OID).

Default value: 1.3.6.1.2.1.1.1.2.0.1

**snmp_monitor_ip**

Specifies the SNMP manager IP address.

Default value: 127.0.0.1

**snmp_trap_begin_detail_oid**

Specifies the Fast Clone SNMP trap detail OID for the trap that Fast Clone sends when it starts unloading the source data.

Default value: 1.3.6.1.2.1.1.1.2.0.3

**snmp_trap_end_detail_oid**

Specifies the Fast Clone SNMP trap detail OID for the trap that Fast Clone sends when it finishes unloading the source data.

Default value: 1.3.6.1.2.1.1.1.2.0.4

**snmp_trap_oid**

Specifies the Fast Clone SNMP trap OID.

Default value: 1.3.6.1.2.1.1.1.2.0.2

# Transparent Data Encryption Parameters

In the [MKEY_ID] and [MKEY_VALUES] sections of the configuration file, specify master key IDs and values for Oracle sources that use Transparent Data Encryption (TDE).

**Important:** Fast Clone uses these parameters only with the direct path unload method. For the conventional path unload method, ensure that the Oracle wallet is open before you start unloading the source data.

## Master Key IDs

The [MKEY_ID] section contains a list of master key IDs from the Oracle wallet. Use the following parameter to specify a master key ID:

***master_key_id***

A master key ID. Each *master_key_id* value in the configuration file must be a unique key.

Add all of the master key IDs, except for ORACLE.SECURITY.DB.ENCRYPTION.MASTERKEY, to the [MKEY_ID] section. For example:

```
KEY1=AX66JYiXI0/Kv0BpzPZ31PsAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
KEY2=BVRVfkdd3wD3uMD23F8L2tECAwAAAAAAAAAAAAAAAAAAAAAAAAAA
```

**Tip:** To get a list of master key IDs from an Oracle wallet, run the following command in the console:

```
orapki wallet display -wallet wallet_file_name
```

The Oracle orapki utility is located in the *ORACLE_HOME*/bin directory.

## Master Key Values

The [MKEY_VALUES] section contains a list of master key values from the Oracle wallet. Use the following parameter to specify a master key value:

***master_key_id***

The master key value of the master key ID that is defined by the matching *master_key_id* key in the [MKEY_ID] section.

Add master key values for all master key IDs that are specified in the [MKEY_ID] section. For example:

```
KEY1=AEMAASAATegG3c24WbhSe7nCMuv8qObLm9F+QKpgNGfvqj8ewdIDEAAlV
+AR6hr9rRxzIBe7JZYsBQcAeHALAQ8VDQ==
KEY2=AEMAASAAhe+zE9L3EmcIxfTq35K9fyvFZUdyWaYu/fhl0hgt/DgDEAAmyAKGUoutcRMHnfl/
xGdABQcAeHALAQ8VDQ==
```

**Tip:** To get a master key value from an Oracle wallet, run the following command in the console:

```
mkstore -wrl wallet_file_name -viewEntry master_key_ID
```

The Oracle mkstore utility is located in the *ORACLE_HOME*/bin directory.

# APPENDIX C

# Glossary

### cloning configuration file

A text file that contains all of the parameters for a data-cloning job. Informatica recommends that you use the Fast Clone Console to create and manage cloning configuration files to reduce the risk of error. Alternatively, you can create and edit configuration files in a text editor.

### conventional path unload method

Fast Clone unload method that uses the OCI to retrieve metadata and data from the Oracle source. This method usually provides an acceptable and efficient level of performance but is slower than the direct path unload method. With the conventional path unload method, you can run Fast Clone on a system that is remote from the source database server without any special permissions. You can also unload data from Oracle views, cluster tables, and index-organized tables (IOTs) and perform table JOIN operations. See also *direct path unload method* on page 146.

### data cloning

The general process of copying source data to the target database. This process includes unloading source data to output files or pipes and loading the data from the output files to a target database by using the target load utility. If you use the DataStreamer component, this process includes unloading source data and streaming it directly to a target database by using the target load utility.

### data files

Files to which Fast Clone writes the unloaded source data. Fast Clone generates the data files in the output directory when you run unload jobs that write data to files. You can customize the format of the data files, specify row and column separators to use in the files, and an enclosure character for record fields. By default, Fast Clone names data files by using the source table or table partition name with the .dat extension. If you configured Fast Clone to unload LOB data to separate files, Fast Clone also generates separate data files for storing LOB data. Fast Clone names these data files based on the record identifier for the record that contains the LOB data.

### DataStreamer

An optional add-on component that you can purchase for faster loads of data to Greenplum, Netezza, and Teradata targets. DataStreamer streams the unloaded Oracle data directly to the Greenplum parallel file distribution server (gpfdist), to the Netezza external tables, or to the Teradata Parallel Data Pump, FastLoad, or MultiLoad utility for loading to the targets. You must use the direct path unload method. Because DataStreamer does not use Fast Clone data files, you do not need to reserve storage for these files.

### direct path unload method

Fast Clone unload method that reads data directly from Oracle data files. This method provides high-performance unloads of data without increasing overheard on the source system. However, you cannot use this method to unload data from a remote Oracle database server, perform SQL JOIN operations, or unload data from views, cluster tables, or index-organized tables (IOTs). See also *conventional path unload method* on page 146.

### Fast Clone Console

A graphical user interface from which you configure and manage data-cloning processes. From the Fast Clone Console, you can generate a cloning configuration file and run unload jobs on a local or remote system. You can run the Fast Clone Console on the Oracle source system, a target system, or a standalone system.

### Fast Clone executable

An executable file, FastReader.exe on Windows or FastReader on Linux or UNIX, that runs unload jobs. You can start the Fast Clone executable from the command line or from the Fast Clone Console. You can also use the Fast Clone executable with the RUN_AS_SERVICE=true option to run the Fast Clone Server.

### Fast Clone Server

An optional add-on component that you can purchase to enable network communication across systems in a distributed Fast Clone topology. The Fast Clone Server runs as a Windows service or as a daemon on Linux or UNIX. You can run the Fast Clone Server on an Oracle source system to initiate data unload jobs from a remote system. You can also run the Fast Clone Server on the target system to accept output files from the source system.

### Fast Clone Server configuration file

A configuration file for the Fast Clone Server. This file contains parameters that control how the Fast Clone Server runs. When you start the Fast Clone Server from the command line, you enter the SERVER_CONFIG parameter to specify the server configuration file. Fast Clone provides a predefined server configuration file, server.ini, in the top-level Fast Clone installation directory. You can edit this file to override the default Fast Clone Server settings.

### load script

A .cmd script on Windows or .sh script on Linux and UNIX that you run to load data to a target. Fast Clone generates the load script in the output directory when you unload data to the data files or pipes. The load script invokes the target load utility and specifies the data files or pipes from which to get data. Typically, you run this script on the target system to load data to the target database. You can also run the load script on another system where the target load utility is installed. By default, Fast Clone names the load script based on the source owner or schema name. See also *output files* on page 147.

### output files

Files that the Fast Clone executable generates in the output directory when you unload data to data files or pipes. These files include the load script and data files. Optionally, these files can also include control files, SQL scripts, and format files for the target load utility. When you run the load script, Fast Clone invokes the target load utility, and the load utility uses the other output files to load data to the target. See also *load script* on page 147 and *data files* on page 146.

### remote configuration management

An option in the Fast Clone Console that you can use to manage cloning configuration files that are stored on remote systems. After you enabled remote configuration management in the Fast Clone Console, you can open, edit, and save the cloning configuration files on a remote system by using FTP.

### server configuration file

See *Fast Clone Server configuration file* on page 147.

### unload job

The process of unloading data from an Oracle source. To unload the source data, Fast Clone either reads the Oracle data files directly or uses the OCI to retrieve data from an Oracle source.

### unload script

The unload.cmd script on Windows or unload.sh script on Linux and UNIX that Fast Clone provides in the top-level installation directory. This script accepts a path to the cloning configuration file as a command line parameter and invokes the Fast Clone executable with this configuration file for data unload processing.

# INDEX