

Oracle® Business Intelligence Cloud Service
Using Oracle Business Intelligence Cloud Service Data Sync
E54616-01

April 2015

Explains how to load data using the Oracle Business Intelligence Cloud Service Data Sync.

Using Oracle Business Intelligence Cloud Service Data Sync

E54616-01

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Primary Author: Dan Hilldale

Contributors: Oracle Business Intelligence development, product management, and quality assurance teams.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate failsafe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

Contents

1 Oracle Business Intelligence Cloud Service Data Sync

Business Intelligence Cloud Service Data Sync Overview	1-1
Installing and Updating the Business Intelligence Cloud Service Data Sync	1-2
Security Guidelines and Requirements	1-3
Prerequisites, Supported Databases, and JDBC Requirements	1-3
Setting Up the Software	1-4
Setting Up Email Notifications	1-5
Configure Email Sender	1-6
Configure Email Recipients and Assign Them to Jobs	1-6
System Properties	1-6
Software Alerts	1-7
Updating the Software	1-7
Setting Up Connections	1-7
Data Loads From Relational Sources	1-9
Data From Tables	1-9
Data From SQL	1-10
Data Loads From Files	1-11
Importing Files	1-11
File Format	1-13
Error Handling and Logging	1-13
Load Strategies	1-14
Creating and Modifying Tables	1-16
Rolling Deletes	1-16
Jobs	1-16
Monitoring and Troubleshooting	1-17
Populating Multiple Cloud Targets	1-19
Multiple Sources	1-19
Scheduling	1-20
Integrating with Other Processes	1-21
Distributing Data Sync Metadata	1-22
Exporting Metadata	1-22
Importing Metadata	1-23
System Properties	1-24

Oracle Business Intelligence Cloud Service Data Sync

This chapter provides help file topics for the BI Cloud Service Data Sync.

This chapter contains the following topics:

- [Business Intelligence Cloud Service Data Sync Overview](#)
- [Installing and Updating the Business Intelligence Cloud Service Data Sync](#)
- [Setting Up Connections](#)
- [Data Loads From Relational Sources](#)
- [Data Loads From Files](#)
- [Load Strategies](#)
- [Creating and Modifying Tables](#)
- [Jobs](#)
- [Monitoring and Troubleshooting](#)
- [Multiple Sources](#)
- [Scheduling](#)
- [Integrating with Other Processes](#)
- [Distributing Data Sync Metadata](#)
- [System Properties](#)

Business Intelligence Cloud Service Data Sync Overview

The Business Intelligence Cloud Service Data Sync supports the load of on-premises data residing in one or more relational or comma-separated value file sources into the schema provisioned on the Oracle Business Intelligence Cloud Service.

To work with the BI Cloud Service Data Sync, you configure and work with the following:

- **Connections.** Source and target connection definitions.
- **Projects.** Defined metadata about target tables, relational and file sources, and target tables, columns, and indexes. Projects are workspaces that help organize and define what is uploaded.

For example, if you have human resources and finance systems on-premises from which you want to upload data, you can register tables from both sources in one

project called "My Project", or you can create one project for each called "My HR Project" and "My Finance Project". Such partitions may be helpful if there is more than one user working on each system.

- **Jobs.** Units of work that upload data from one or more relational tables or files as defined in the project. Jobs can be executed, scheduled, and monitored.

At a high level, you load data into the Oracle Business Intelligence Cloud Service schema using the following steps, depending on the source type.

Loading from Relational Structures

To load from relational structures:

1. Register one or more on-premises data sources.
2. Import the tables that need to be replicated.
3. Optionally, provide a SQL query to override how data is read from a table. This query can have any typical SQL constructs such as filters, joins, and sorts, anything that can be represented in SQL which is valid for the type of source database you are connecting to. Optionally, you can also sub-select the columns that need to be replicated.
4. Alternatively, when there is no particular source table, you can use a query whose results can be populated in a new table.
5. Specify the load strategy (replace data always, incrementally upload data, append data, insert only, update only).
6. Create a unique index, if necessary, when incremental load is desired.

Loading from Files

To load from a file:

1. Register a file.
2. Verify the mapping, choose a subset of attributes, and generate a table definition.
3. Optionally, specify whether to remove duplicates.
4. Optionally, load more than one table from a file (normalize the data).
5. Specify the load strategy (replace data always, incrementally upload data, append data, insert only, update only).
6. Create a unique index, if necessary, when incremental load is desired.

Running Jobs

Once you have defined the sources and load strategies, you can run and monitor jobs to load data.

Installing and Updating the Business Intelligence Cloud Service Data Sync

To install the Data Sync, you must meet the requirements and prerequisites, then unzip and run the application. Once installed, the Data Sync client notifies you of any available updates. This section includes the following topics:

- [Security Guidelines and Requirements](#)
- [Prerequisites, Supported Databases, and JDBC Requirements](#)

- [Setting Up the Software](#)
- [Setting Up Email Notifications](#)
- [System Properties](#)
- [Software Alerts](#)
- [Updating the Software](#)

Security Guidelines and Requirements

To perform data loads, the Data Sync tool requires Oracle Business Intelligence Cloud Service and on-premise source system user accounts with correct role assignments for use with its connections. In addition, there are important security guidelines for where the Data Sync tool should be installed.

User Accounts

To support data extraction and load, the following user accounts and roles must be created for use in Data Sync connections:

- Uploading data to a schema provisioned along with Oracle Business Intelligence Cloud Service requires the BI Data Load Author role. Create a user only for data loading purposes, and provide the user minimal access other than the required role.
- The connection to the on-premises source databases needs only to have a user with read privileges. It is recommended that such a user be created and used for the connections to any of your on-premises data sources.

Installation Guidelines

WARNING: The Data Sync tool stores sensitive information, including connection information to your on-premises databases. While the tool can run on any desktop environment, it is highly recommended that it be installed in production in a controlled environment where the operating system and file system privileges are tightly controlled.

Prerequisites, Supported Databases, and JDBC Requirements

Before installing, you must have the Java 1.7 or later Java Developer Kit (JDK). On an on-going basis, any critical updates specific to this version of Java should always be applied.

Note: The Java Developer Kit is required. The equivalent version of the Java Runtime Environment (JRE) is insufficient to support the installation.

Database Support

The Data Sync supports the following databases:

- Oracle (source and target)
- Microsoft SQL Server (source and target)
- DB2 (source and target)

- Teradata (source and target)
- MySQL (source only)
- Oracle TimesTen (source and target)

JDBC Drivers

The Data Sync is a pure java application and uses JDBC to retrieve data from databases. While Oracle JDBC version 11.2.x is packaged with the application, this version should be replaced with the JDBC version specific to the source database vendor and version. The appropriate database-specific JDBC drivers must be copied to the \lib directory after installation of the application. For example, if your Oracle version is any different, copy the JDBC driver from your local Oracle installation, which should match with the database version, and overwrite the provided JDBC driver.

Table 1–1 JDBC Drivers

Vendor	JDBC Driver Name
Oracle	ojdbc6.jar
MySQL	Mysql-connector-java*.jar
Microsoft SQL Server	sqljdbc.jar
DB2	db2java.zip
TimesTen	ttjdbc6.jar, orai18n.jar, timestenjmsxla.jar, jms.jar, javax.jms.jar
Teradata	terajdbc4.jar, log4j.jar, teradata.jar, tdgssjava.jar, tdgssconfig.jar

Setting Up the Software

To set up the software, copy the BICSDataSync.Zip file to an installation directory with no spaces in its name, and unzip the files.

WARNING: The Data Sync tool stores sensitive information, including connection information to your on-premises databases. While the tool can run on any desktop environment, it is highly recommended that it be installed in production in a controlled environment where the operating system and file system privileges are tightly controlled.

Setting the Java Home

Depending on your operating system, edit either the config.bat or config.sh file, modifying the line that sets the JAVA_HOME. Replace the @JAVA_HOME with the directory where the JDK is installed.

Note: If your directory has spaces in the name, then put double-quotes around it.

For example, on Windows:

```
set JAVA_HOME=D:\Java
```

For example, on UNIX:

```
JAVA_HOME=usr/java
```

To run the Oracle Business Intelligence Cloud Service Data Sync application, run `datasync.bat` (Windows) or `datasync.sh` (Linux/UNIX) in the extracted Oracle Business Intelligence Cloud Service Data Sync directory. To access the software easily, you can create a shortcut to the desktop for this file.

Running the Data Sync for the First Time

The first time that the process is run, it recognizes that this is the first time it has run and begins with a configuration setup wizard, which prompts you for the following:

1. A Logical Name for the repository. This name is used to distinguish the repository in multi-repository environments. For example, you could name the instance Development Environment or Production Environment, and that name is then displayed as a tooltip for the system tray Data Sync icon and also on the title bar of the client.
2. A password. Provide a password that can be used to access the client. You can also select to have the tool remember the password.

Note: Because the Data Sync client stores the transactional system passwords, the tool should only be installed in protected environments.

3. Create a project or select an existing one.
4. Set system properties and email configuration. For information about setting up email, see "[Setting Up Email Notifications](#)".

Running the Data Sync After the First Time

Run `datasync.bat` (Windows) or `datasync.sh` (Linux/UNIX) to restart the application. Upon successful launch of the application, it appears as a system tray icon, and the icon's tool tip displays the repository name. Subsequently, a login dialog box is displayed for the client, in which you can specify that the system remember the password if you did not specify this upon first run. If you close the client, you can restart by clicking the system tray icon and selecting **Start UI**. In the login screen, enter or confirm the password and click **Login**, then create a project or select an existing one.

Reconfiguring the Data Sync From the Beginning

To reset the Data Sync to its default state and redo the setup process, run `datasync.bat` (Windows) or `datasync.sh` (Linux/UNIX) in a command window with the `-clean` option.

Uninstalling the Data Sync

To uninstall the Data Sync, delete the install directory.

Setting Up Email Notifications

You can set up email notifications for data upload success or failure. Complete the steps in both of the following procedures:

- [Configure Email Sender](#)
- [Configure Email Recipients and Assign Them to Jobs](#)

Configure Email Sender

The Data Sync has a built-in login-authentication based email (SMTP) client, which connects to any SMTP login-authenticating server. Configure the email administrator account to enable recipients to receive email notifications.

To configure the email sender:

1. Select the **Tools** menu, then select **Email**, and then select **Setup**.
2. In the Setup dialog box's Email Configuration tab, enter the details for the email address that you want to receive information notifications.

Field	Description
User Name	User name for the email account.
Password	User password on the email server. (Only required if you select Needs Authentication.)
Email Server	Host name of the email server.
Email Server Port	Port where the email server listens.
Email Address	Email address of the user.
Needs Authentication	Specifies whether the corporate email server requires authentication.
Needs SSL	Specifies whether an SSL connection is required.

3. Click **Send Test Email** to test the connection.
4. Click **Save**.

Configure Email Recipients and Assign Them to Jobs

You can register one or more recipients to receive emailed updates about job status. Configured email recipients can then be assigned to specific jobs so that the person responsible for monitoring a job receives email notifications.

To configure email recipients:

1. Select the **Tools** menu, then select **Email**, and then select **Recipients**.
2. Click **New** in the toolbar and enter the following information:

Field	Description
Name	Name of the user to be notified.
Email Address	Email address where the notification is sent.
Inactive	Indicates whether the email notification for this recipient is active or inactive. When checked, the person is not included in the email broadcast.

3. Navigate to the Jobs tab in the Jobs view and assign an email recipient from those you have configured to each job.

System Properties

You can set properties to customize how the Data Sync works in your environment. For a detailed description of the properties, refer to "[System Properties](#)". To set properties, select the **Views** menu, then select **System Properties**.

During initial deployment of the Data Sync, two properties are worth configuring:

- **Data File Root Directory:** When set, the directory defaults to this location when registering a file to be uploaded.
- **Maximum number of run history entries to preserve:** Use this property to indicate how many run history entries should be preserved for your installation. As days progress, any entries older than the maximum are purged from the run history.

Software Alerts

If there is a new patch available or a new software package with additional functionality, the Data Sync client notifies you using the New Patch Alerts icon next to the Server Monitor icon. Prior to each load, the Data Sync performs a version check, sending its current version to be checked against the one on the cloud. Depending on whether the versions match, the following actions occur:

- If the versions match, the data load continues.
- If the minor version is changed on the cloud, indicating an available patch, an optional alert is created and an email sent prior to continuing with the load. The alert is sent only once.
- When the major version is changed, indicating a new software package, an alert is created and an email sent. The data load halts, while informing you that a new version of the patch is required prior to doing any further data loads.

The number of unread alerts is indicated on the Alerts icon. Click the icon to view a list of alerts. If the icon is black there are no alerts, and it turns green when there are unread optional alerts, and red when there are unread mandatory alerts.

Updating the Software

New software updates are downloadable as compressed files that contain all content for a new installation of the software. To update the software, perform a full installation of the software as a new environment in a separate home directory, then migrate the environment from your existing installation. For information about setting up a new environment, refer to "[Setting Up the Software](#)."

Setting Up a New Environment with an Existing Environment Configuration

When you start the new environment after installing a patch or new version, when prompted for environment configuration, select **Copy existing environment configuration** and then specify the existing Data Sync environment's home directory. The Data Sync repository and configuration files are copied to the new environment. If the new installation requires metadata upgrade, perform any upgrade after the files are copied.

Setting Up Connections

This section provides instructions for specifying the connections used by the Data Sync. Upon initial setup, two default connections are set up:

- **Target:** This is the connection to the Oracle Business Intelligence Cloud Service. 2010, 2015,

- File Source: This is the connection to the file source. It does not require any configuration (Do not delete or rename this entry)

Note: Do not deleted or rename these entries.

To specify on-premises data source connections:

1. In the Connections view's Sources/Targets tab, you'll see the default Target and File Source connections.
2. For each record, enter the following information in the Edit subtab:

Field	Description
Name	Logical name for the data source. You should not change the names of the precreated records. When creating new sources, use a descriptive and environment-agnostic name such as CRM or HR. Avoid using instance-specific names such as host names, as the same connection can be configured against different databases in different environments such as development and production.
Connection Type	Database type.
Table Owner	Schema owner name. This is the user who owns the objects on the source schema.
Password	Password for the database user.
Service Name, TNS Name, Connection String, or Database Name, or ODBC Data Source (Note: Field name changes based on the Connection Type selection.)	Enter the value appropriate for your database. For an Oracle TNS Name, enter the TNS name that is specified in the tnsnames.ora file in \network\admin\.
Host	Machine name or IP address of the machine where the database resides.
Port	Port number where the database listens (for example, 1521 is the default for an Oracle database).
URL (Optional)	A JDBC URL for the data source connection. The value in this field must conform to the database specifications. Use this option to specify a unique URL for a particular data source. For example, this option can be useful if this physical data source uses Oracle RAC and other data sources use a different database type.

3. Click **Test Connection** to make sure the connection works.
4. Click **Save**.

Note: You have to ensure that the JDBC driver version matches that of the database. A version mismatch can lead to spurious errors during the data load process. Even using an Oracle database, if the version of the JDBC driver does not match that of the database, download the compatible version of the JDBC driver from Oracle's website and place it in the lib directory.

For Oracle, it is preferable to use Thin type of connection. In some cases, such as RAC configurations, you may have to use the OCI8 type connection. In this case, also make sure the client library installed on the local machine matches with the database's version.

Data Loads From Relational Sources

You can load data into the Oracle Business Intelligence Cloud Service schema directly from either a relational table, view or a SQL statement.

Data From Tables

You can import table definitions to load from using any of the defined relational connections. Supported data types include CHAR, VARCHAR, TIMESTAMP, DATE, NUMBER(n), NUMBER(m,n), CLOB, and BLOB. If a source table has columns with any other data type, they are imported with UNKNOWN data type, and the column will be marked as inactive, and will not participate in the data copy process.

To import one or more table definitions from a connection into the Oracle Business Intelligence Cloud Service schema:

1. In the Project view, select the Relational Data tab.
2. Click **Data From Table**.
3. In the Import Tables into [*Project*] dialog box, select the connection in the Data Sources drop-down list.
4. In the Table Filter field, type a table name or a table name filter, using wildcards to narrow the list of tables for import. The following examples filter the list of tables from a source.
 - CONTACT will show only the CONTACT table if it exists in the database with exactly the same name.
 - CONTACT* or CONTACT% lists all tables in the database whose name start with CONTACT.
 - *CONTACT* or %CONTACT% lists all tables in the database whose name contains CONTACT.
5. Click **Search Tables**.
6. In the Searching Tables confirmation dialog box, click **OK**. The Table List includes all tables from the source meeting the filter you applied, if any.
7. Select the **Import** check box for any tables in the list you want to replicate, then click **Import Tables** to register the source tables and create entries with the same name for target tables. All columns and indexes are also imported.

Note: Do not rename the tables. The Data Sync assumes that the source table name and target table name are the same. If you want to use a different target table name, consider using queries as a source.

8. If you chose to load data incrementally a unique index is suggested on the user/primary key columns. It is also recommended that you register additional indexes which support joining with other tables and can be used for filtering purposes while reporting.

By default, all table attributes are copied. If you want to preclude certain columns from being replicated to Oracle Business Intelligence Cloud Service because they are not needed for analysis or may contain sensitive information, select the table in the Target Tables tab, then select the Table Columns subtab and select the **Inactive** check box for the column or columns. If you deactivate a column, be sure to inspect the index definitions which may be referring to the inactive columns. Any index that refers to an inactive or deleted column definition gets dropped, but is not created. If you would like to deactivate the indexes that may refer to inactive or deleted columns, right-click the column and select the **Identify and inactivate invalid indexes** option. This marks any indexes that refer to inactive columns inactive as well.

Using a SQL Query Override to Refine a Data Load from a Table

By default, all data from a source table is copied to the Oracle Business Intelligence Cloud Service schema. To limit the data loaded, you can provide a SQL query override to refine the data that is read. For example, if you want to copy one year's worth of data on a table that includes a LAST_UPD date column, you could provide an additional query (Oracle Syntax) as:

```
SELECT * FROM TABLE_NAME WHERE LAST_UPD > SYSDATE - 365.
```

To add a SQL query to limit the data loaded:

1. In the Project view, select the source table in the Relational Data tab.
2. In the Edit subtab, click the button for the Query field.
3. In the Query dialog box, use the editor tools to enter the desired SQL query.
4. Click **OK**.

When you provide a SQL query override, it is validated against the database and, if there are errors in the SQL, you are prompted to correct it. If the SQL override includes new columns that are not present in the table definition, you are prompted to add them to the target table.

For example, take a case where a CONTACT table is imported. By default, the Data Sync will issue `SELECT * FROM CONTACT`. You may want to add a column named `UPLOADED_DT` to the table to record when the data is uploaded. To do this, you could provide SQL query override such as the following:

```
SELECT CONTACT.*, SYSDATE AS UPLOADED_DT FROM CONTACT
```

In this case, the Data Sync recognizes that the new column `UPLOADED_DT` does not exist on the target and offers to add it to the table definition.

Data From SQL

Another approach to load data into the Oracle Business Intelligence Cloud Service schema is to use a SQL statement whose results you want to persist. For example, instead of loading detail data, you may want to use an aggregate SQL to store

compressed data on the cloud. This aggregate SQL may join multiple tables, and use any of the SQL functions your database supports such as GROUP BY, filters, and joins.

To create a Oracle Business Intelligence Cloud Service target table from the results of a SQL query:

1. In the Project view, select the **Relational Data** tab.
2. Click **Data From SQL**.
3. In the New Query dialog box, enter a logical name for the query in the Name field. The name should not contain spaces.
4. Specify an existing target table or create a new one and provide a name for the table. If the query defines a new table, the column definitions are inferred from the SQL structure. If you use an existing table, any new columns from the SQL can be added to the list of columns.
5. Select a connection in the Connection drop-down list.
6. Enter the SQL query in the Query window.
7. Click **OK**.
8. If you chose to load data incrementally a unique index is suggested on the user/primary key columns. It is also recommended that you register additional indexes which support joining with other tables and can be used for filtering purposes while reporting.

Data Loads From Files

In the system properties, you can define the root directory of data files for import by using the "Data File Root Directory" property. This specification allows the tool to open the directory as the default directory for choosing files, and ensures portability across different operating systems. It is recommended that all of the files are kept in this directory, as this property change helps in easily configuring the repository when moving the metadata to another environment without having to adjust the properties for each file entry.

Importing Files

When you import files, the Data Sync estimates the data types and other characteristics of the data and allows you to edit the values before creating a corresponding Oracle Business Intelligence Cloud Service data source in the Target Tables tab and registering the columns to load the data into. By default, up to 10,000 rows are taken as a sample to identify the data types. You can specify that a higher or lower number of rows be sampled, or choose -1 to read the whole file, though it may take longer to parse the whole file.

During import, the tool estimates the length of string attributes in the file so that the corresponding target table definition can accommodate the data. If it's a string representation, then the length is computed in increments of 50. If the length is more than 50% of the computed length, the next increment is used. The following table illustrates how the length is computed.

Table 1–2 String length computation

Maximum length of string in sample data	Computed length
5	50

Table 1–2 (Cont.) String length computation

Maximum length of string in sample data	Computed length
27	100
55	100

To register files for import:

1. In the Project view, select the **File Data** tab.
2. Click **New**.
3. In the New Source File: Select File dialog box, click the button in the Select a File field and navigate in the Open dialog box to select the file to load from.
4. Click **Open**. The File field is populated with the name of the file. You can then change the file name to include patterns using wildcards. For example:
 - D:/csvFiles/AIRLINE_TRAFFIC.csv - To load a specific file.
 - D:/csvFiles/AIRLINE_TRAFFIC*.csv - To load any file that starts with a name AIRLINE_TRAFFIC.
 - D:/csvFiles/*.csv - To load all files that end with a .csv extension.
5. Enter a descriptive name in the Logical Name field.
6. In the Target Table option section of the dialog box, select whether to load an existing table or create a new one. When you choose to create a new table, a name for the table is required.
7. Enter the number of lines to skip in the file, if any. Use this option if your file has a header and the actual data starts from a specific line. If no lines should be skipped, leave the default, 0.
8. Select a delimiter for the file from the drop-down list. Options include Comma, Space, or Tab.
9. Select a timestamp format from the available options in the drop-down menu.
10. Select the First line contains headers checkbox if your file contains column names in a header. In this case, the tool parses the first line to generate column name suggestions. The suggested column name labels are upper cased, and the spaces replaced with underscore (_) characters. When the data file does not contain headers, then column names appear as "COLUMN_1, COLUMN_2", and so on, equal to the number of attributes in a row.

Note: Any special characters are removed from the column header, and are upper cased. If the column name length is more than 30 characters, it gets shortened.

11. Select the **Remove Duplicates** checkbox if the file contains duplicate records and you want the process to select a distinct set of rows based on a certain attribute or set of attributes that you identify as user key columns that can enforce uniqueness and de-duplicate the records. A unique index is suggested by the tool and used to enforce uniqueness and detect duplicates. The index also improves the performance of lookup operations.

12. Click **Next**. In the New Source File: Map Columns dialog box, the tool identifies if an attribute is a timestamp, integer, decimal or string type and creates a default column mapping that you can edit.

By default, up to 10,000 rows are taken as a sample to identify the data types. You can specify that a higher or lower number of rows be sampled, or choose -1 to read the whole file, though it may take longer to parse the whole file. If an column is a number, a length of 38 is assumed. If it is a decimal value, a length of 38 and a precision of 10 is assumed.

Note: It is important to verify the data type suggestions made by the tool and make corrections as needed.

13. In the Targets subtab, verify the column mapping by selecting the appropriate table and clicking **Column Mapping**. The Columns dialog box shows the mapping between the file's fields and the corresponding target table.
14. If you want to load more than one table from the same file, you can click **New** in the Targets tab for the file and use the above process to populate another table with a subset of attributes from the file.
15. The file can have empty lines, which will be ignored.

File Format

File format definitions and requirements include the following:

- You can specify a single character as delimiter. It can be a comma (","), or space (" ") or a tab ("\t" or "[tab]").
- When an attribute contains the delimiter as part of the value, the whole value needs to be double quoted.
- An attribute can contain new line characters, in which case it also needs to be double quoted.
- A file can have a banner, which can be ignored during load. However, the transition from header to data should predictably start from a certain line number.
- A file can have timestamp and date strings. However, only one format per file can be used. For example, if there is birth date and hire date, both need to be formatted in the same way. As an example, "yyyy-MM-dd" can be used to interpret timestamp "1968-01-01" as birth date, and "2010-06-31" as hire-date.
-

Refer to

<http://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html> for various formatting possibilities.

Error Handling and Logging

When a file is parsed and loaded, errors can result either while reading or writing. Read-related errors are mainly due to conversion of strings to an object of type integer, decimal, or timestamps. They can also result from bad formatting. For example, if an attribute contains the delimiter and it's not double quoted, or a line does not have as many attributes as the header. When there are bad records, the process fails.

Write-related errors can result from insufficient length or entering null into a not null attribute.

When such records are encountered, the errors are logged in a file in the log directory with the naming convention `CR_<Table/File Name>_<From Connection>_<To Connection>.<Timestamp>.bad`.

This log file contains information including the line number of the bad record, the record itself, and what problems occurred when parsing the file.

Once the bad records are identified, fix the original file and rerun the process. If you are unsure about how to fix a record in the file at the location specified in the .bad log file, you can comment it out by adding "--" in front of the line in the file.

Load Strategies

When loading data into a table, you define how the data is loaded. When you choose a load strategy that incrementally loads the data, the tool requires you to define a user key which can be used to uniquely identify a record on the target side, and a DATE/TIMESTAMP based column which can be used to identify the incremental data. If such an index did not get imported while importing the table definition, one is suggested.

For example, for a given table CONTACT_ID is the unique identifier for any record, and the table has a date column called LAST_UPD whose value is updated to current timestamp whenever a record is created or updated, you would choose CONTACT_ID for user key and LAST_UPD column as the Filter.

In this example, the first time the data is extracted, the tool issues a `SELECT * FROM CONTACT` statement. If the first load happened on January 1, 2014 at 10:00 AM, the subsequent load would issue the following SQL statement (Oracle syntax): `SELECT * FROM CONTACT WHERE LAST_UPD > TO_DATE('01-01-2014 10:00', 'MM-DD-YYYY HH24:MI')`. The record set then compares the CONTACT_ID value to the data already existing in the Oracle Business Intelligence Cloud Service schema. Any record without a match is inserted. Any records with a match are updated. It is important that the source system contains an index for the incremental filter column.

The following options are available.

Table 1–3 Load strategies

Value	Description
Replace data in table	<p>Delete any existing data and reload data always. Also applies to loads where a unique key is not available. Does not require a primary key or filter column.</p> <p>Table is truncated before data load on each run. Any indexes are dropped prior to data load and recreated after load. Table is analyzed at the end to update statistics.</p>
Append data to table	<p>New data is added to the table without checking for any prior existence of data. Does not require a primary key or filter column.</p> <p>Table is never truncated. If registered indexes do not exist on the provisioned schema, they are created after the data load.</p>

Table 1–3 (Cont.) Load strategies

Value	Description
Update table (Add new records)	<p>Requires a primary key or filter column. If the data with the user key is not available, then it is inserted, else the record is ignored.</p> <p>During initial run, the table is truncated before the first load. Indexes are dropped prior to data load, and recreated after load. The table is analyzed at the end to update statistics.</p> <p>During incremental runs, the table is not truncated. Data is applied incrementally, and any missing indexes are created. The table is analyzed at the end. If any index is marked as "Always drop and create", those indexes do get dropped and created even during incremental runs.</p>
Update table (Update existing records)	<p>Requires a primary key or filter column. If the data with the user key is available, then it is updated, else it is ignored.</p> <p>During initial run, the table is truncated before the first load. Indexes are dropped prior to data load, and recreated after load. The table is analyzed at the end to update statistics.</p> <p>During incremental runs, the table is not truncated. Data is applied incrementally, and any missing indexes are created. The table is analyzed at the end. If any index is marked as "Always drop and create", those indexes do get dropped and created even during incremental runs.</p>

Setting Load Strategies for Relational Data

To set load strategies for a relational source:

1. In the Project view, select the **Relational Data** tab.
2. Select the connection in the list and, either in the Load Strategy column or in the Load Strategy field in the Edit subtab, click the button.
3. In the Load Strategy dialog box, select the load strategy you prefer. If you select the Never delete data check box, the target table is never truncated. This target table-level attribute is particularly useful in cases where a table was created and populated by some other means, and you do not want Data Sync operations to truncate the table.
4. Click **OK**.

Setting Load Strategies for File Data

To set load strategies for a file data source:

1. In the Project view, select the **File Data** tab.
2. Select the file in the list and, in the Targets subtab, click the button for the Load Strategy field for the target table.
3. In the Load Strategy dialog box, select the load strategy you prefer. If you select the Never delete data check box, the target table is never truncated. This target table-level attribute is particularly useful in cases where a table was created and populated by some other means, and you do not want Data Sync operations to truncate the table.
4. Click **OK**.

Creating and Modifying Tables

When data load happens to a target of type "Oracle (Oracle Business Intelligence Cloud Service)" target tables are created on the Oracle Business Intelligence Cloud Service schema automatically prior to loading the data. For subsequent data load runs, the table definition from the local repository is compared against that in the Cloud schema, and altered dynamically. You can also create the tables prior to load manually by navigating to the Target Tables tab in the Project view, right-clicking, choosing **Drop/Create/Alter Tables**, and in the dialog box selecting the target BI Cloud Service target. This process causes errors if the table is not alterable. Examples of such situations include:

- Changing a null column into a not null column.
- Changing the datatype.
- Reducing the length or precision.

Under such circumstances, you can manually drop and re-create the table by right-clicking the target table in the Target Tables list in the Project view and selecting **Drop/Create/Alter tables**. You can also choose this option to drop/create/alter tables manually.

Note: Dropping and re-creating the table clears the refresh date for that table. This will force a full load on the table in the subsequent load.

Creating and Modifying Non-Oracle Business Intelligence Cloud Service Tables

For trial run purposes or testing, it is possible that you might use the Data Sync to populate a non-Oracle Business Intelligence Cloud Service or on-premises database. In this case, you should create and alter the tables manually prior to data load. This can be done by navigating to the Target Tables tab in the Project view, right-clicking, choosing **Drop/Create/Alter Tables**.

Rolling Deletes

If you wish to subset the data loaded into the Oracle Business Intelligence Cloud Service schema for the past 'n' number of days, you can control this behavior by providing a SQL Query Override. However, as time passes, the data in the Cloud schema continues to accumulate. If you want to limit data in the Cloud schema to a certain period and periodically purge older data, you can specify Rolling Delete Days at the target table level. For example, for the CONTACT table, if you want to purge all data greater than a year, in the Target Tables tab in the Project view, enter 365 for Rolling Delete Days for the table. You can set this in the table list or in the Edit subtab.

It is important to define an index on the column used to identify the older records. This improves data load performance.

Jobs

A job is the unit of work you use to organize, schedule, execute, and monitor load processes. Every Project when created comes with a default job with the naming convention of <Project Name>-Job1. In the Jobs view, you can view the default job or create new jobs in the Jobs tab. When you select a job, the Ordered Tasks subtab lists the tasks which are run when the job is executed. A job is initially empty, with no

ordered tasks. Upon the first run of a job the tasks are automatically computed and executed in order.

Any changes you make to the project require re-computing the ordered task list. For example, if a run fails and a metadata change is made prior to restarting the run, then the failed run gets marked as completed, and a new run will be started to accommodate re-computation of the ordered tasks.

When there is more than one job in an incomplete status (such as Running or Failed or Stopped), the job execution fails. In this case, it is necessary to ensure all the incomplete runs are completed prior to starting a job. To mark a run as completed, right-click the run entry and select **Mark as Completed**. For information about monitoring job execution, see "[Monitoring and Troubleshooting](#)".

If there are mutually exclusive set of data flows, the order would be 0 for all of the tasks. If the same target table is populated by more than one data flow, then they will be staggered based on the load strategy.

Manually Computing a Job Definition

If you want to manually compute a job definition, right-click in any tab in the Project screen and select **Update Job**.

Restarting Jobs Automatically

Uploading data to the cloud could encounter failures that are network related, which otherwise could run just fine if one were to just reattempt the run. If you want the tool to automatically retry upon failure, you can set the number of retries at the job level.

Monitoring and Troubleshooting

Once you have defined all the tables, SQL, and files whose data you want to copy to the cloud schema, click **Run Job** on the toolbar to run a job.

You can monitor the status of a job on either the Current Jobs tab or the History tab. The record in the top pane shows the summary of the job. The Tasks tab below lists one item per user-defined data flow. The Task Details tab elaborates on the line items for the data flow. The tasks and the details show important metrics, including start and end timestamps, number of rows processed, read and write throughput (number of rows processed per minute).

The status description of the task detail contains detailed message in a field called status description. It also publishes the specific log file name to look at when there is an issue.

The run log files are stored in the log directory. One directory per run is created with a naming convention of `CR_<Table/File Name>_<From Connection>_<To Connection>.<Timestamp>.log`.

Task Details

The task details typically contain the following details:

- **Truncate Table:** When the load strategy is set to Replace data in table or when loading a table (without the never delete data option) for the first time, truncate table is executed.
- **Drop Index:** When a table is truncated, all the indexes on the table are dropped prior to the data load. The indexes whose 'Drop/Create Always' property is checked, always get dropped prior to the data load, even if the table is incrementally loaded.

- **Insert/Update/Append/Upsert:** Depending on the load strategy, an appropriate command is used to load the data.
- **Create Index:** When an index is dropped, it gets created after the data is loaded. Any new index that is registered which does not exist on the cloud also is created.
- **Analyze table:** After data load and index creation, the table is analyzed.

Table Analysis

Because table analysis, wherein table statistics are collected, can be an expensive and long-running process, the following methodology is enforced.

- Upon initial load, the table is analyzed.
- For subsequent loads, if a table is truncated or any index is dropped or created, the table is analyzed.
- For all other normal incremental runs, the client does not send the analyze request, but instead notifies you that the statistics will be collected as part of the nightly scheduled routines. The analyze table task detail is created, but the status description indicates that the analyze table request was not sent.

Abandoning a Failed Run

If for some reason you want to abandon a failed run, right-click its record in the Current Jobs tab and choose **Mark as Completed**, which updates the job status to Completed. A subsequent request starts a new run.

Performing a Complete Reload of Data

After having been in an incremental mode, if you wish to perform a complete refresh of the data in the cloud. To reload all tables, on the **Tools** menu, select **Re-extract and load all data**. Confirm that you want to do this in the confirmation dialog box by clicking **OK**. Upon confirmation, the next job run is performed in Full mode rather than Incremental mode, and reads and loads all data.

If you want to re-extract all data for a single table, in the Connections view, select the table and, in the Refresh Dates tab, click the **Re-Extract Data** button and, when prompted in the Re-Extract Data dialog box, choose **Selected record only** and click **OK**. In the Re-Extract Data dialog box, select the **All Data** option and click **OK**. The next run will extract all data from the source table and reload the table.

Reloading Data that Is Already Loaded

The tool remembers when a table is loaded into the database, which is typically the start time of the job. If your source database is in a different time zone, the value is adjusted accordingly. You can view this timestamp in the Refresh Dates tab of the Connections view.

In the case of an incremental load, the run looks for records whose filter column value has changed since the timestamp of the last load. For example, take a load run after a load performed on June 1st, 2014 10:00 PM, only changes after that time would be extracted and loaded. You can modify the date to set it to a previous timestamp by clicking the **Re-Extract Data** button and providing a new value in the Re-Extract Data dialog box. For example, if you were to set the timestamp to March 1st, 2014 10:00 PM, the next run would look for data that changed since March 1st rather than June 1st.

Populating Multiple Cloud Targets

In the case that you have multiple cloud targets, such as one for development and another for production, once you have tested your work against your development schema, you may want to populate the production one also. There are two ways to do this: you can use the same environment, but create a new job, or you can set up a mirror environment where you can import the data from the development environment.

Creating New jobs

To populate multiple cloud targets using a new job:

1. Determine the source and, if necessary, create a new data source in the Data Sync client. In the case you want to populate multiple cloud targets using a new job, there are two possibilities for the source database:
 - You want to extract from the same source and populate a production target on the cloud.
 - You want to extract from a different source and populate a production target. In this case, create another data source that points to the source you want to populate the production environment from.
2. For the target database, create another connection of type "Oracle (Oracle Business Intelligence Cloud Service)". For example, you could name it Production.
3. In the Jobs tab in the Jobs view, create a new job. In the New Job dialog box, give it a meaningful name, for example, "Production Job" and click **Next**.
4. The next page of the New Job dialog box displays the currently used connections. To remap these to the new source (if applicable) and the newly created target connection, select the new connections in the Override With column and click **Finish**.

Setting Up a Different Environment

Setting up a different environment is preferable when there are a lot of updates happening on the development schema, or when the teams responsible for the development and production environments are different.

To set up a different environment for the first time, export the system and its logical metadata from the development environment and import it into the production environment by choosing to truncate the tables using the Export and Import commands on the Tools menu.

After initial setup, export only the logical metadata from the development environment and import into production by choosing to truncate the tables.

Multiple Sources

In the case that you have different kinds of sources in your environment, you may want to consolidate their data for analytic purposes. Multiple sources can be of three types: heterogenous, homogenous (of the same version), and homogenous (of different versions).

Heterogenous Sources

Heterogenous sources are of different types, for example CRM, Oracle EBS, Peoplesoft, JDEdwards, and so on. The sources typically contain mutually exclusive sets of tables. You can either choose to use the same project or different ones. A single project is

convenient when you have a single schedule to upload the data. Multiple projects, one per source are convenient when you need to schedule dataload at different times and/or multiple people work on them.

Homogenous Sources of the Same Version

Homogenous sources of the same version occur when you have the same kind of source, but multiple instances of it. For example, you could have two instances of CRM, one used for North America and the other for Europe. You can manage data extraction from both of these sources with a single project.

To manage data extraction for homogenous sources of the same version in the example described, consider the following tips and requirements:

- Create a connection for the database in North America and another for the one used for Europe.
- Use one of the databases for identifying the tables and queries to replicate.
- Test the load process from one source.
- Create a new job, where you can map the existing source (with which you defined metadata) to the new one.
- Schedule the jobs separately.

Homogenous Sources of Different Versions

Homogenous sources of different versions are very similar to the case of homogenous sources of the same version. The only difference is that there are tables that may have the same name but different structures and schema definitions. There are two ways of performing data upload: using a single project or using multiple projects, one per type of source.

To use a single project:

- Use one source as a primary source to import the table definitions.
- Use Query as a means of importing data from the others.
- If there are new attributes from the queries, add them to the target table definition.

To use multiple projects:

- Create individual projects, one per version of the source system.
- Import tables and queries from sources into their respective projects.
- Take care that the target tables to have similar datatypes. For example, if COLUMN1 for Table1 in source1 is of type VARCHAR, and the same table column has a type of DATE in the other, the data loads from one or the other source will fail.

Scheduling

In the Jobs view, you can use the scheduler to specify the periodicity of your scheduled jobs.

To schedule a job:

1. Select the **Job Schedules** tab to open the scheduler.
2. In the Edit tab, enter a name for the schedule and select the job you want to schedule from the drop-down list.

3. Use the scheduler's options to specify whether the job should run only once or on a recurring basis, whether it should be automatically restarted upon a failure, and to mark the schedule as inactive if it is not in use.
4. If the schedule is recurring, specify a recurrence pattern using the scheduler's options. You can have a scheduled job run on intervals of minutes, hours, specific days of the week, on specific days of each month or specified periods of one or more months, or on a certain day of a given month every year.
5. Specify a start date and whether you want the schedule to end by a certain date or never end.
6. Click **Save** to save the schedule.
7. In the top pane, you can view and select schedules and view their status and other properties. In the Last Schedule Status, you can mark a schedule as queued or completed. In the Next Run column for a schedule, click the button to open a Date dialog box, in which you can specify a new date to run the scheduled job on. Upon this change, the schedule resumes under the recurrence pattern you specify when creating the schedule.

Integrating with Other Processes

In some cases, you may want to trigger the upload process from an extraneous process. There are three ways of doing this: SQL-based, file-based, and command line based.

SQL-Based

SQL-based integration is convenient when an external process needs to dictate when a job can begin, if it can perform a SQL operation on a data source. To integrate using this method, you create a trigger table that can be polled to initiate a job.

To integrate using SQL:

1. In the Connections view, register the desired connection.
2. Navigate to the conf-shared directory and edit the on_demand_job.xml file. Edit the following properties:
 - The polling interval.
 - The datasource name that has the trigger table.
 - The time periods between which the Data Sync should poll the table.
3. Create a table on that datasource called JOB_TRIGGER containing the following four columns:

Column Name	Datatype	Length	Values	Unique
UNIQUE_KEY	VARCHAR	250		Yes
JOB_NAME	VARCHAR	250		No
EFFECTIVE_DT	DATE			No
INACTIVE_FLG	CHAR	1	Y/N	No

Below is an example of the Data Definition Language for a trigger table and index:

```
CREATE TABLE JOB_TRIGGER
(
```

```
UNIQUE_KEYVARCHAR(250) NOT NULL
,JOB_NAMEVARCHAR(250) NOT NULL
,EFFECTIVE_DTDATETIME NOT NULL
,INACTIVE_FLGCHAR(1) DEFAULT 'N' NOT NULL
)
;
CREATE INDEX JOB_TRIGGER_U1 ON JOB_TRIGGER(UNIQUE_KEY)
;
```

4. An entry in this table now triggers the job upload. As soon as the job request is started, the entry is deleted. Note that you can also specify during which period polling should occur.

File-Based

In the `on_demand_job.xml` file found in the `conf-shared` directory, you can specify a file which will trigger a certain job. The `TriggerFile job` parameter allows you to specify a job and a file whose existence triggers the job. For example:

```
<TriggerFile job="Job1" file="c:\abc.txt"/>
```

In the example above, the Data Sync polls for the presence of the `abc.txt` file, and when found it triggers `Job1` execution. Upon successful invocation of the job, the file is deleted. For this reason, it is important to create a file just as a semaphore, it should not contain anything of importance.

You can also specify time windows for polling, as in this example in which file polling happens between the 12:00 AM and 2:30 AM and between 6:00PM and 8:00PM every day:

```
<TriggerFile job="Job2" file="c:\xyz.txt">
<TimeWindow startTime="00:00" endTime="02:30"/>
<TimeWindow startTime="19:00" endTime="20:00"/>
</TriggerFile>
```

Command Line Based

There is a `datasyncCmdLine.bat/.sh` file, which can be used to start a job and to obtain the status of a running job. Edit the file to find the usage. The file `dac.properties` is configured with all that is needed to use the `dacmdline`.

Distributing Data Sync Metadata

You can use the Import/Export feature to make changes in one environment, then export the whole environment as XML files and then import the XML files into another environment.

This section includes the following topics:

- [Exporting Metadata](#)
- [Importing Metadata](#)

Exporting Metadata

You can use the Import/Export feature to export source system-specific Data Sync metadata out of the repository to migrate metadata from one environment to another, such as from a development environment to test or production environments.

Caution: When you export metadata, all files in the target folder are deleted.

Data Sync behavior relating to the target folder is as follows:

- If the target folder is empty, Data Sync exports the metadata without a warning.
- If the target folder contains Data Sync metadata, the client issues a warning and you must click OK to proceed with the export. The export process replaces all content in the target folder.
- If the target folder has non-Data Sync metadata as well as Data Sync metadata, the client issues a warning, and you must click OK to proceed with the export. The export process replaces all content in the folder. All non-Data Sync metadata is deleted.
- If the target folder has only non-Data Sync metadata, Data Sync cannot export into the specified target folder.

To export Data Sync metadata:

1. In the Data Sync, on the **Tools** menu, select **Export**.
2. Select the directory to which you want to export metadata, or accept the default directory.
3. Select the appropriate applications that you want to export metadata for.
4. Select the appropriate categories of metadata you want to export:
 - **Logical.** Exports all information contained in the Project view.
 - **System.** Exports all information contained in the Connections view, except passwords for servers and database connections.
 - **Run Time.** Exports information about jobs and schedules contained in the Jobs view.
 - **User Data.** (Applicable to Data Sync standalone authentication only) Exports users, roles, and passwords. **Note:** When importing roles and passwords, if the encryption key is different in the repository to which you are importing from that from which the metadata was exported, the roles and passwords will be unreadable.
5. Click **OK**.
6. Verify the export process by reviewing the log file <Domain_Home>\log\export.log. You can use the Overwrite Log File option to overwrite earlier export logs.

Importing Metadata

You can use the Import/Export feature to export source system-specific Data Sync metadata out of the repository to migrate metadata from one environment to another, such as from a development environment to test or production environments.

To import Data Sync metadata:

1. In the Data Sync, on the **Tools** menu, select **Import**.
2. Select the directory from which you want to import metadata, or accept the default directory.

3. Select the appropriate applications for which you want to import metadata.
4. Select the appropriate categories of metadata you want to import:
 - **Logical.** Exports all information contained in the Project view.
 - **System.** Exports all information contained in the Connections view, except passwords for servers and database connections.
 - **Run Time.** Exports information about jobs and schedules contained in the Jobs view.
5. If you are importing metadata into a blank repository or to completely replace selected categories of the current metadata in the repository, select **Truncate repository tables**. This option overwrites the content in the current repository. It also greatly increases the speed of the import process.
6. If you want to import new records and update existing records, select **Update existing records**. If you do not select this check box, Data Sync inserts only new records. This option is only available if you do not select the Truncate Repository Tables option.
7. (Optional) Select **Enable bulk mode** to insert the imported metadata into the repository as an array insert. You should elect this option only if you also selected the Truncate Repository Tables option.

This action increases the speed of the import process.
8. Click **OK**.
9. Verify the import process by reviewing the log file `<Domain_Home>\log\import.log`.

System Properties

System Properties allow you to customize your environment. To set system properties, click the **Views** menu and select **System Properties**, then set the value in the Value column of the System Properties dialog box. Click **Save** to save changes, then **Close** to close the dialog box.

Note: The Data File Root directory requires configuration, as this property has no default value.

Available system properties include:

- **Allow Clients To Remember User Password:** When set to true, the Data Sync client can remember the user's password while logging into repository. Passwords, though encrypted, are stored in the client machine's file system. While accessing critical environments, like Production environments, clients connecting to the repository may not be trustworthy. Under such circumstances, you can set this flag to 'false'. When set to false, even if on the login screen the Remember password option is selected while connecting to such a repository, the Data Sync client warns that the password cannot be remembered while logging into this repository, and the user is required to reenter the password upon each login.
- **Concurrency Level:** This property configures the maximum number of jobs that can run in parallel. When concurrency of 1 is set, ETL runs in the same process space with the Data Sync. When multiple jobs are allowed, they run as separate operating system processes. Any invalid value is interpreted as 1. Restart the server to effect changes to this property.

- **Data File Root Directory:** This property sets the Data File Root Directory for importing data files. When set, the directory defaults to this location when registering a file to be uploaded.
- **Maximum number of run history entries to preserve:** This property allows retention of a certain number of run history entries. As days progress, any older entries in the run history are purged. A value less than or equal to zero preserves everything. This is not recommended for Data Sync setup, where the repository is maintained in the Java DB.
- **Purge Run Log Files:** The server maintains a unique directory for each run for storing the log files under the server\log directory. The naming convention for the log directories is of the format <Job Name>.<Process Identifier>. You can set how many days' log files are preserved. For example, if you set this property to 30, all log directories with a creation date prior to 30 days are deleted. When set to -1, the log directories and files are never deleted. The lowest value for this property is 7 days. If any value is set below 7, will default this to 7. Restart the server to effect changes to this property.
- **Repository Name:** Descriptive name for repository. This name is displayed on the title bar. Restart the client to effect changes to this property.
- **Server Log Level:** This property sets the logging level. Note that this setting is case sensitive. The server has a logging facility that logs various events, error conditions, and information both during the process of executing various tasks and when idle. Logging includes, among other things, the results of various periodic self-diagnostics. The sum of all this information can be very extensive, and as a consequence, it may be difficult to find a specific piece of information. For this reason, the logging reported by the server is graduated into various levels. The available levels are:
 - FINEST
 - FINER
 - FINE
 - CONFIG
 - INFO
 - WARNING
 - SEVEREThese levels perform as gated thresholds. The level FINEST level is the most extensive amount of reporting. This may be a good setting for the server for the process of debugging and testing various execution plans before going into production. Each subsequent level in the list above represents less extensive levels of reporting. The lowest logging level, SEVERE, results in the server reporting only the most important messages, which are usually error conditions. When in production, a reasonable value for this property is INFO. Restart the server to effect changes to this property.
- **Test Run:** When this property is set to true, during an data load run if there are any errors they are ignored. Changes are effective for the next job. Restarting the server is not required for changes to this property.

