Oracle® Enterprise Data Quality for Product Data

Command Line Interface Guide

Release 5.6.2

E25662-01

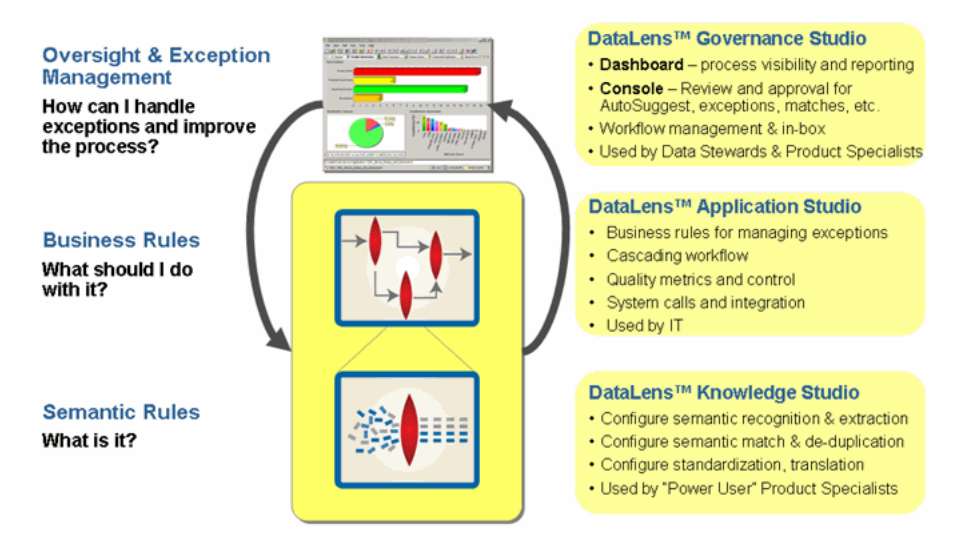
November 2011

This document describes the Oracle Enterprise Data Quality for Product Data Command Line Interface (CLI) and contains the following:

Overview

Oracle DataLens Server is built on industry-leading DataLens™ Technology to standardize, match, enrich, and correct product data from different sources and systems. The core DataLens Technology uses patented semantic technology designed from the ground up to tackle the extreme variability typical of product data.

Oracle Enterprise Data Quality for Product Data uses three core DataLens Technology modules: Governance Studio, Knowledge Studio, and Application Studio. The following figure illustrates the process flow of these modules.



TheEnterprise DQ for Product (EDQP) CLI provides an interface that allows you to run Data Service Applications (DSAs) jobs on a client system by accessing a remote Oracle DataLens Server. Using the CLI to run a DSA job is the same as if the job were run from the Governance Studio, Services for Excel, or the Oracle DataLens Server Administration Web Pages; you should use these methods rather than the CLI whenever possible.

The CLI can be used to:

- Run jobs from an external program, such as Informatica.
- Run jobs synchronously or asynchronously.
- Bypass the Oracle DataLens Administration Server by using the named output steps of a DSA.
- Schedule jobs using external schedulers or utilities like `cron`.
- Retrieve data files from the client system, process them by the Oracle DataLens Servers, and then returned to the client system. This does not require Oracle DataLens Server relative paths.

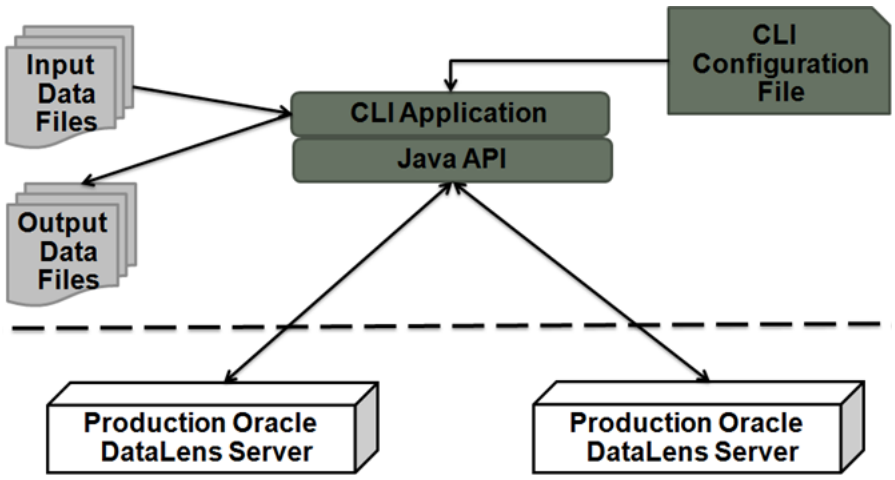
Additional CLI features include:

- Supported on both Linux and Windows.
- Support for secure (HTTPS) Oracle DataLens Servers.
- Tracing for SOAP request and response calls to facilitate debugging.
- The ability to set DSA job priority. For example, the priority should be set to 'low' for jobs that process large quantities of data.
- Supports processing discrete segments of data, which is known as 'chunking'.

CLI Architecture

The CLI overall architecture and how it interacts with the EDQP API framework is shown in Figure 1.

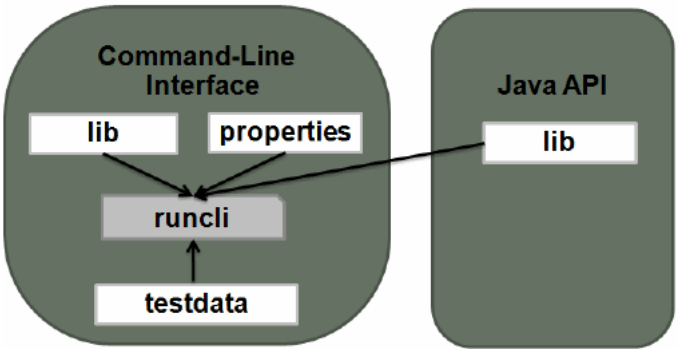
Figure 1 CLI Architecture



The CLI and Java API framework is used to process input data files by initiating a DSA job on a client system using the parameters defined in the configuration file, sending the job to an Oracle DataLens Server, and then returning the output data files to the client.

All DSA interaction with data lenses is supported by the CLI. It requires no programming because all parameters are passed to the DSA by the integrated CLI and Java API libraries using the customizable CLI properties files as in shown in Figure 2.

Figure 2 CLI Operation

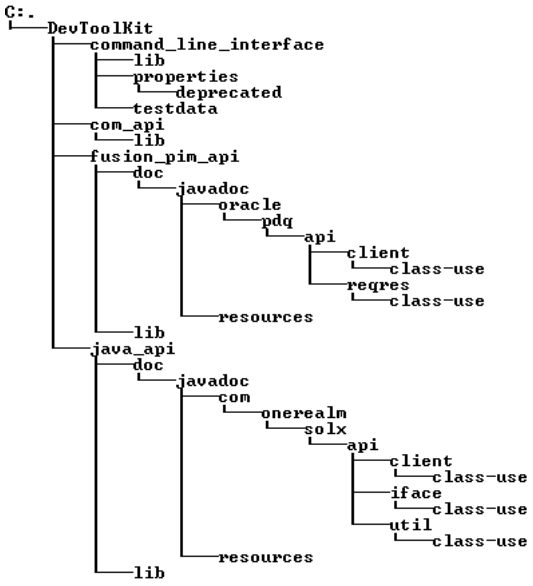


You can use the CLI with a single Oracle DataLens Transform Server, Administration Server, or the entire topology of Production Oracle DataLens Servers.

CLI Components

The CLI is packaged as part of the Developers Toolkit (`DevToolKit`), which also includes the COM, Java, and Fusion PIM APIs as shown in Figure 3. As such, it does not require any additional installation.

Figure 3 DevToolKit Contents



The CLI is delivered with the CLI library, sample DSAs, data lenses, and data files. It is in the `DevToolKit\command_line_interface` directory and contains the following components by directory:

Directory	Components
`command_line_interface`	All components needed to run the CLI organized in subdirectories and the following sample run scripts: `run.bat` - Batch file to run a DSA job on a Windows system that executes `runcli.bat`. This file is modified to run the property file that corresponds to the type of DSA job to be run. `runcli.bat` - Batch file to invoke the CLI Java executable. `runcli.sh` - Shell file to run a job and invoke the CLI Java executable on a Linux system.
`command_line_interface\lib`	The EDQP CLI Java Archive (JAR) file, `edqp-cli.jar`.
`command_line_interface\properties`	The customizable property files to configure the parameters passed to the DSA processing job by based on the type of input expected by the DSA: `db_test_dsa.properties` - Used to configure database input. `file_test_dsa.properties` - Used to configure text input. `xml_test_dsa.properties` - Used to configure XML input.
`command_line_interface\properties\deprecated`	Previous property files for assisting existing installations using the previous "WFG" property file format to upgrade to the new "DSA" format.

Directory	Components
command_line_interface\testdata	The sample DSAs, data lens, and input data files: create_resistors.sql - Creates sample database input and output database tables. order.xml - XML file that can be updated. samplePMapIdef.pmap - Text input DSA package including data lens. samplePMapIdefDbInput.pmap - Database input DSA package including data lens. samplePMapIdefXml.pmap - XML input DSA package including data lens. test_100.txt - Text input file.

Using the CLI to Run DSA Jobs

There are a variety of ways to use the CLI to run DSA jobs, which are dependent on your DSAs that are custom to your environment. This section provides the following examples:

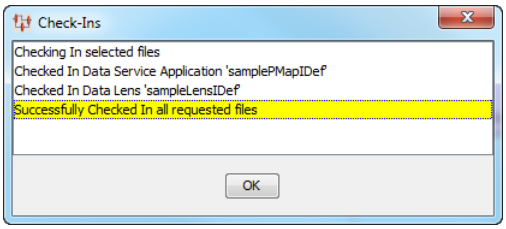
- "Running a Text Input DSA Job" on page -5
- "Running a Database Input DSA Job" on page -8
- "Running an XML Input DSA Job" on page -12

Running a Text Input DSA Job

Use the following steps to run a DSA job that expects text input using the CLI sample DSA:

1. On a client system, ensure that the DevTookKit package has been extracted and locate the DevToolKit\command_line_interface directory.
2. Start the EDQP Application Studio.
3. From the **File** menu, select **Import Package** and import the samplePMapIdef.pmap package from the DevToolKit\command_line_interface\testdata directory.

   For more information, see *Oracle Enterprise Data Quality for Product Data Application Studio Reference Guide*.
4. From the **DSA** menu, select **Check-In Package** to check in and deploy the DSA and data lens.



Note: Be sure to check in the entire package, not just the DSA, to avoid the following error when running the DSA job:

```
Error 1301. Data Lens 'sampleLensIDef' not loaded on server PROD1 -
Please check that the Lens is deployed to Development
```

5. Using a text editor, edit the `DevToolKit\command_line_interface\properties\file_test_dsa.properties` file.
6. Locate and edit the following parameters to match your environment:

Parameter	Modification
`IN_FILE=`	Change to the full path of the `DevToolKit\command_line_interface\testdata\test_100.txt` directory. For example, `C:\dls_install\DevToolKit\command_line_interface\testdata`.
`OUT_DIRECTORY=`	Change to the full path of the directory where you want the CLI output files stored. For example, `C:\dls_install\DevToolKit\command_line_interface`.
`DLS_SERVER_1=`	Change the `localhost:2229` default to the Production Oracle DataLens Server you want to process the job. This can be a Transform or Administration server.

7. Save and close the file.
8. Run the CLI:

   On Linux:

   a. Edit the `DevToolKit\command_line_interface\runcli.sh` file.
   b. Change the `SCS_BASE` parameter to the full path of where `DevToolKit\command_line_interface\` directory is located.
   c. Change the `JAVA_HOME` parameters to the full path of where Java JDK is installed.
   d. Save and close the file.
   e. Run `./runcli.sh` to use the CLI to run the DSA job.

   On Windows:

   1. Using a text editor, edit the `DevToolKit\command_line_interface\run.bat` file.
   2. Change the line to:

      ```
      ./runcli.bat properties.file_test_dsa
      ```

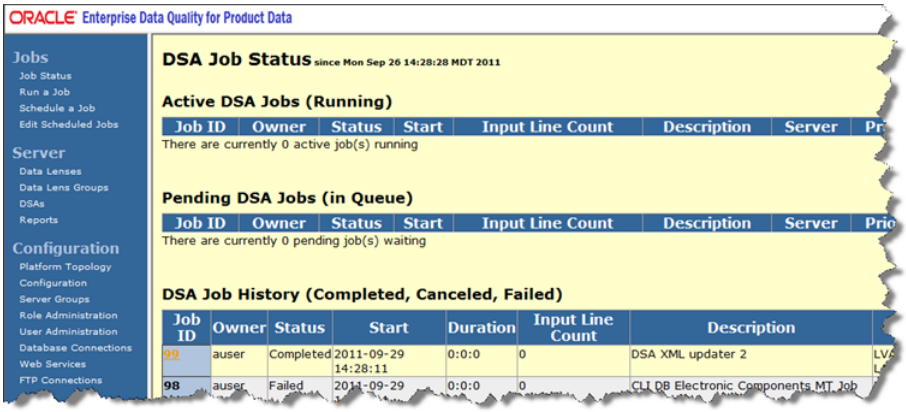
   3. Save and close the file.
   4. From the `DevToolKit\command_line_interface` directory, double-click `run.bat` to use the CLI to run the DSA job.

   A Command Prompt window is displayed while the job is running and is closed when the job is finished.

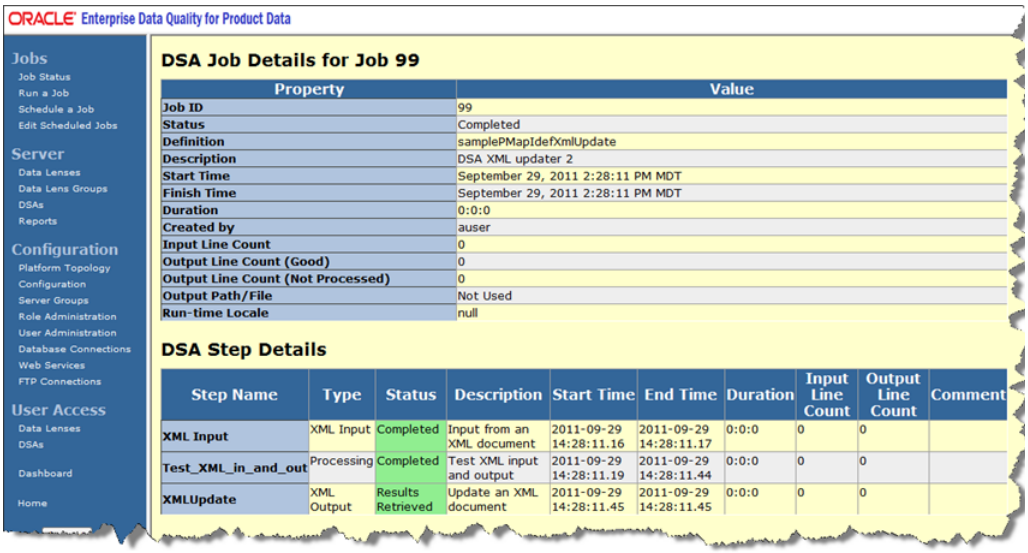
Tip: You can open a Command Prompt window using `cmd`, and then execute `run.bat` to view the runtime messages as they occur.

9. Log into your Oracle DataLens Administration Server Web Job Status Web page. For more information, see *Oracle Enterprise Data Quality for Product Data Oracle DataLens Server Administration Guide*.

10. Click **Job Status** to view the status of your DSA job run and ensure that it has completed.



11. Click the Job ID number of your DSA job to view the details of the job that has been run.



12. Optional - You can view the CLI output files, `samplePmapIdef_exceptions_##.txt` and `samplePmapIdef_output_##.txt`.

You can customize the `file_test_dsa.properties` file to run your DSA jobs using the following additional parameters and the process described in this section:

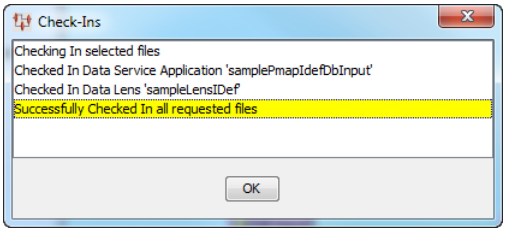
Parameter	Modification
TYPE=	Do not change the DSA default value.
USER=	Change to the user name that you want to appear in the Oracle DataLens Administration Server Job Status Web pages.
APPLICATION_NAME=	Change to the description that you want to appear in the Oracle DataLens Administration Server Job Status Web pages.
DLS_SERVER_2-4= or more	Uncomment and add these parameters and change the server name and port of as many Production Oracle DataLens Transform Servers as you want to process your data. All servers must be in the Production Server Group. This creates a multi-threaded processing environment.
CHUNK_SIZE=	Change to set the number of lines of data that you want to send to the server for processing. For example, if your input file contains 100 lines and you set this parameter to 25, then four chunks of input data are sent to the server for processing. When this value is larger than the lines of data, a single output file is produced. This value should be set to the same or smaller chunk size as on the processing server.
APPEND_JOB_ID=	*Only* set to false when a static file name is needed, such as regression tests because this removes the appending of the DSA Job ID from the file name.
JOB_PRIORITY=	Change the default priority from 3 (Low) to 2 (Medium) or 1 (High).
DSA_NAME=	Change to the name of your DSA.
LOCALE=	Change to your locale value. The default is en_US.
SEPARATOR_CHAR=	Uncomment to use any single character as a separator in your data rather than the default tab character. For example, a '|' could be used.
IS_RT_OUTPUT=	Set to false so that no real-time output is generated. For example, a database update job or to e-mail or FTP results.
SECS_TO_WAIT_FOR_ RETRY=	Change to the number of seconds that you want wait before polling the server for job completion. The default is 10 seconds.
WAIT_FOR_RESULTS=	Set to false, the default, this causes DSA jobs to be run asynchronously; set to true, jobs are run synchronously. Running jobs synchronously allows you to run the job without maintaining an open network connection or if you want the CLI to return control immediately though the job is still processing on the server.
USE_HTTPS=	Change to true if your Oracle DataLens Server topology uses secure HTTP (HTTPS).
IN_FILE_TYPE=	Uncomment to process a group of like files in the same directory and change to the type of files to process, text or xml.
IN_DIRECTORY=	Change to the full path of the directory that contains the group of input files you want to process. Only the files of the type specified by IN_FILE_TYPE= are processed.

Running a Database Input DSA Job

Use the following steps to run a DSA job that expects database input using the CLI sample DSA:

1. On a client system, ensure that the DevTookKit package has been extracted and locate the `DevToolKit\command_line_interface` directory.
2. Start the EDQP Application Studio.
3. From the **File** menu, select **Import Package** and import the `samplePmapIdefDbInput.pmap` package from the `DevToolKit\command_line_interface\testdata` directory.

   For more information, see *Oracle Enterprise Data Quality for Product Data Application Studio Reference Guide*.
4. From the **DSA** menu, select **Check-In Package** to check in and deploy the DSA and data lens.



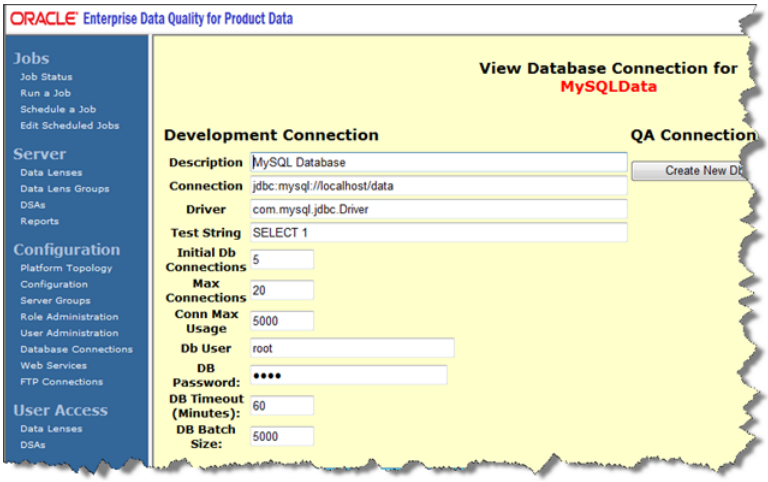
> **Note:** Be sure to check in the entire package, not just the DSA, to avoid the following error when running the DSA job:
>
> ```
> Error 1301. Data Lens 'samplePmapIdefDbInput' not loaded on server
> PROD1 -  Please check that the Lens is deployed to Development
> ```

5. Using a text editor, edit the `DevToolKit\command_line_interface\properties\db_test_dsa.properties` file.
6. Locate and edit the following parameters to match your environment:

Parameter	Modification
`DLS_SERVER_1=`	Change the `localhost:2229` default to the Production Oracle DataLens Server you want to process the job. This can be a Transform or Administration server.

7. Save and close the file.
8. Log into the Oracle DataLens Administration Server Web pages

   For more information, see *Oracle Enterprise Data Quality for Product Data* Oracle DataLens Server Administration Guide.
9. Click **Database Connection**, and then click **Create New Db Connection**.
10. Create and test a database connection named 'MySQLData' on your server as in the following example:



Note: The database connection must be named 'MySQLData' because the `samplePmapIdefDbInput` DSA is preconfigured to use this connection.

---

11. Use the `DevToolKit\command_line_interface\testdata\create_resistors.sql` file to create the necessary SQL database 'Resistors' table by connecting to your database and sourcing this file.
12. Run the CLI:

    On Linux:

    a. Edit the `DevToolKit\command_line_interface\runcli.sh` file.

    b. Change the `SCS_BASE` parameter to the full path of where `DevToolKit\command_line_interface\` directory is located.

    c. Change the `JAVA_HOME` parameters to the full path of where Java JDK is installed.

    d. Save and close the file.

    e. Run `./runcli.sh` to use the CLI to run the DSA job.

    On Windows:

    1. Using a text editor, edit the `DevToolKit\command_line_interface\run.bat` file.
    2. Change the line to:

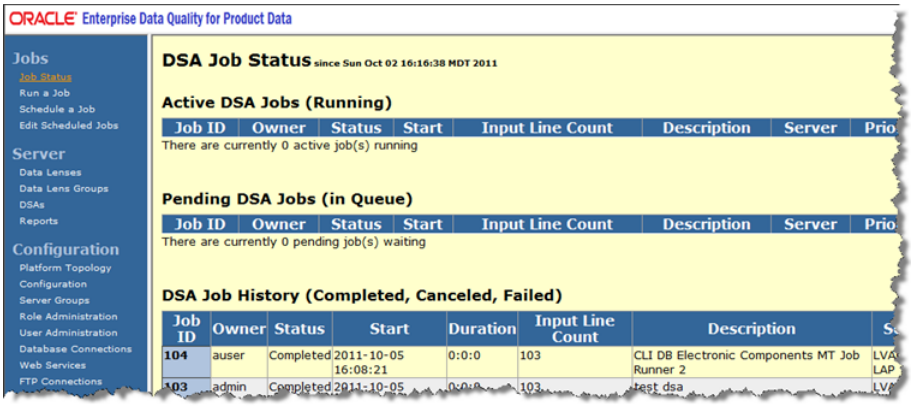
       ```
       ./runcli.bat properties.db_test_dsa
       ```

    3. Save and close the file.
    4. From the `DevToolKit\command_line_interface` directory, double-click `run.bat` to use the CLI to run the DSA job.

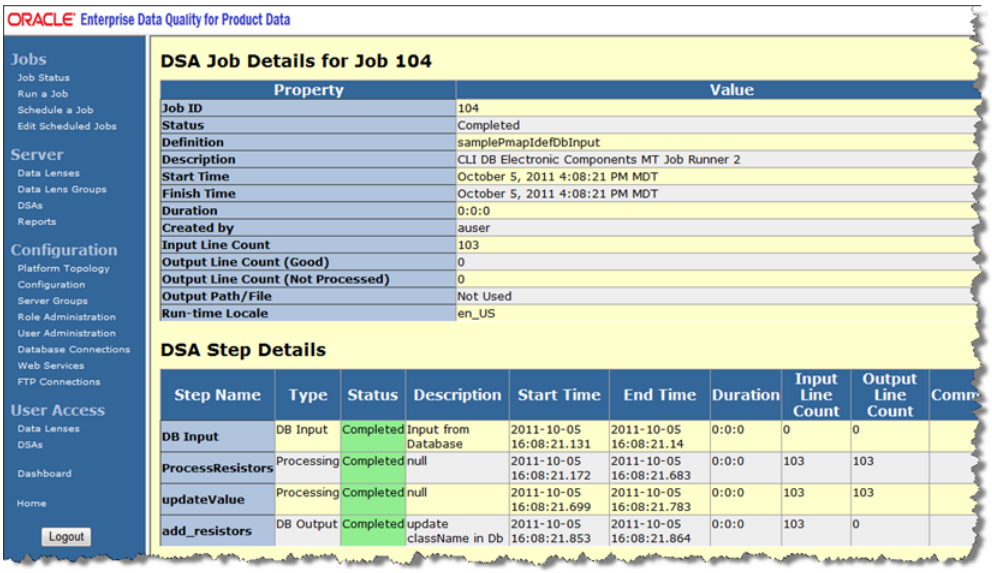
    A Command Prompt window is displayed while the job is run then closed.

Tip: You can open a Command Prompt window using `cmd`, and then execute `run.bat` to view the runtime messages as they occur.

13. Log into your Oracle DataLens Administration Server Web Job Status Web page. For more information, see *Oracle Enterprise Data Quality for Product Data Oracle DataLens Server Administration Guide*.
14. Click **Job Status** to view the status of your DSA job run and ensure that it completed.



15. Click the Job ID number of your DSA job to view the details of the job run.



You can customize the `db_test_dsa.properties` file to run your DSA jobs using the following additional parameters and the process described in this section:

Parameter	Modification
`TYPE=`	Do not change the `DSA` default value.

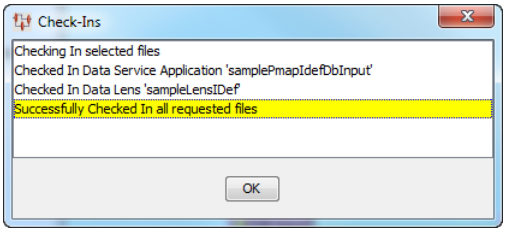
Parameter	Modification
USER=	Change to the user name that you want to appear in the Oracle DataLens Administration Server Job Status Web pages.
APPLICATION_NAME=	Change to the description that you want to appear in the Oracle DataLens Administration Server Job Status Web pages.
PROCESS_THRU_ERRORS=	
JOB_PRIORITY=	Change the default priority from 3 (Low) to 2 (Medium) or 1 (High).
DSA_NAME=	Change to the name of your DSA.
LOCALE=	Change to your locale value. The default is en_US.
DB_PARAM_1=	Optional database parameters for use with input database select statements.
SEPARATOR_CHAR=	Uncomment to use a Sheffer stroke as a separator in your data rather than the default tab character.
IS_RT_OUTPUT=	Set to false so that no RT output is generated. For example, a database update job or to e-mail or FTP results.
SECS_TO_WAIT_FOR_RETRY=	Change to the number of seconds that you want wait before polling the server for job completion. The default is 10 seconds.
WAIT_FOR_RESULTS=	Set to false, the default, causes DSA jobs to be run asynchronously; set to true jobs are run synchronously. Running jobs synchronously allows you to run the job without maintaining an open network connection or if you want the CLI to return control immediately though the job is still processing on the server.

Running an XML Input DSA Job

Use the following steps to run a DSA job that expects XML input using the CLI sample DSA:

1. On a client system, ensure that the DevTookKit package has been extracted and locate the DevToolKit\command_line_interface directory.
2. Start the EDQP Application Studio.
3. From the **File** menu, select **Import Package** and import the samplePMapIdefXmlUpdate.pmap package from the DevToolKit\command_line_interface\testdata directory.

   For more information, see *Oracle Enterprise Data Quality for Product Data Application Studio Reference Guide*.
4. From the **DSA** menu, select **Check-In Package** to check in and deploy the DSA and data lens.



Note: Be sure to check in the entire package, not just the DSA, to avoid the following error when running the DSA job:

```
Error 1301. Data Lens 'samplePMapIdefXmlUpdate' not loaded on
server PROD1 -  Please check that the Lens is deployed to
Development
```

5. Using a text editor, edit the `DevToolKit\command_line_interface\properties\xml_test_dsa.properties` file.
6. Locate and edit the following parameters to match your environment:

Parameter	Modification
`IN_FILE=`	Change to the full path of the `DevToolKit\command_line_interface\testdata\test_100.txt` directory. For example, `C:\dls_install\DevToolKit\command_line_interface\testdata`.
`OUT_DIRECTORY=`	Change to the full path of the directory where you want the CLI output files stored. For example, `C:\dls_install\DevToolKit\command_line_interface`.
`DLS_SERVER_1=`	Change the `localhost:2229` default to the Production Oracle DataLens Server you want to process the job. This can be a Transform or Administration server.

7. Save and close the file.
8. Run the CLI:

   On Linux:

   a. Edit the `DevToolKit\command_line_interface\runcli.sh` file.
   b. Change the `SCS_BASE` parameter to the full path of where `DevToolKit\command_line_interface\` directory is located.
   c. Change the `JAVA_HOME` parameters to the full path of where Java JDK is installed.
   d. Save and close the file.
   e. Run `./runcli.sh` to use the CLI to run the DSA job.

   On Windows:

   1. Using a text editor, edit the `DevToolKit\command_line_interface\run.bat` file.
   2. Change the line to:

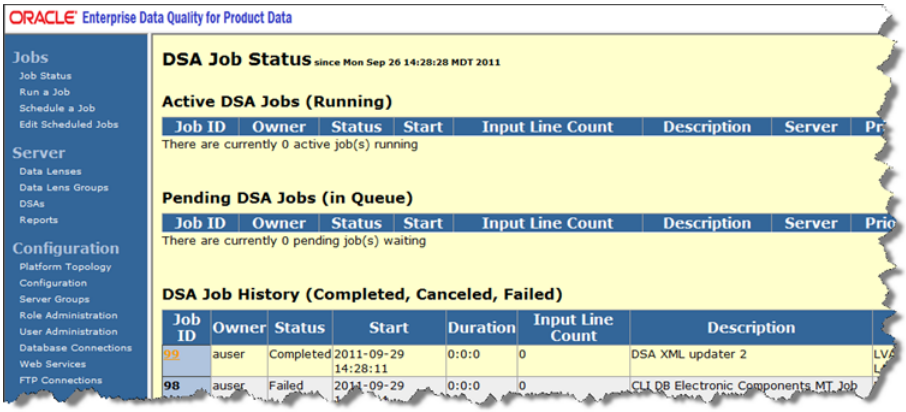
      ```
      ./runcli.bat properties.xml_test_dsa
      ```

   3. Save and close the file.
   4. From the `DevToolKit\command_line_interface` directory, double-click `run.bat` to use the CLI to run the DSA job.

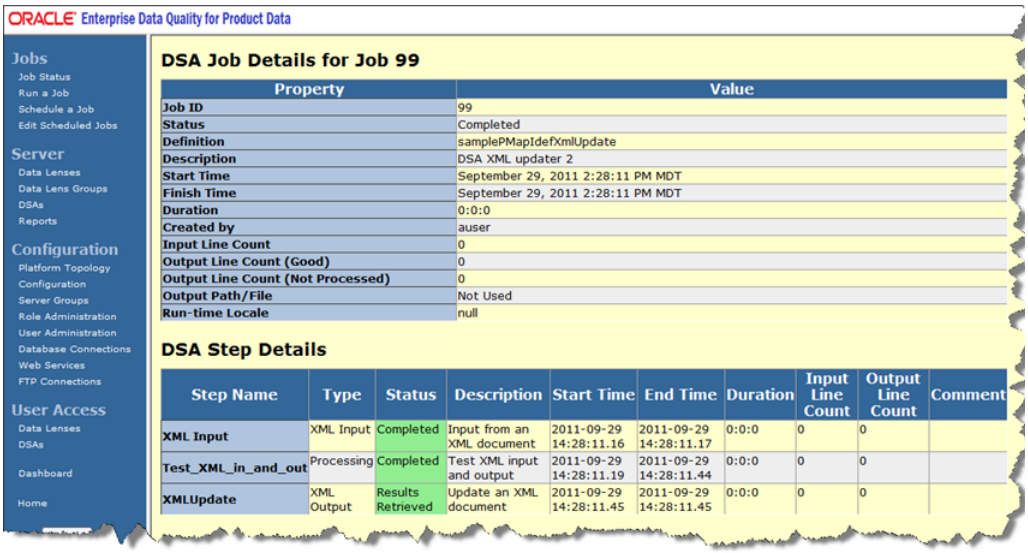
   A Command Prompt window is displayed while the job is run then closed.

Tip: You can open a Command Prompt window using `cmd`, and then execute `run.bat` to view the runtime messages as they occur.

9. Log into your Oracle DataLens Administration Server Web Job Status Web page. For more information, see *Oracle Enterprise Data Quality for Product Data Oracle DataLens Server Administration Guide*.
10. Click **Job Status** to view the status of your DSA job run and ensure that it completed.



11. Click the Job ID number of your DSA job to view the details of the job run.



12. Optional - You can view the CLI output files, `samplePMapIdefXmlUpdate_exceptions_##.txt` and `samplePMapIdefXmlUpdate_output_##.txt`.

You can customize the `xml_test_dsa.properties` file to run your DSA jobs using the following additional parameters and the process described in this section:

Parameter	Modification
TYPE=	Do not change the DSA default value.
USER=	Change to the user name that you want to appear in the Oracle DataLens Administration Server Job Status Web pages.
APPLICATION_NAME=	Change to the description that you want to appear in the Oracle DataLens Administration Server Job Status Web pages.
DSA_NAME=	Change to the name of your DSA.
LOCALE=	Change to your locale value. The default is en_US.
APPEND_JOB_ID=	*Only* set to false when a static file name is needed, such as regression tests because this removes the appending of the DSA Job ID from the file name.
SECS_TO_WAIT_FOR_ RETRY=	Change to the number of seconds that you want wait before polling the server for job completion. The default is 10 seconds.
WAIT_FOR_RESULTS=	Set to false, the default, causes DSA jobs to be run asynchronously; set to true jobs are run synchronously. Running jobs synchronously allows you to run the job without maintaining an open network connection or if you want the CLI to return control immediately though the job is still processing on the server.
IN_FILE_TYPE=	Do not change the xmlUpdate default value.

Related Documents

For more information, see the following documents in the documentation set:

- The *Oracle Enterprise Data Quality for Product Data R12 PIM Connector Installation Guide* provides installation and configuration of Enterprise DQ for Product R12 PIM Connector.
- The *Oracle Enterprise Data Quality for Product Data R12 PIM Connector User's Guide* provides highlights of the core process steps and features of Enterprise DQ for Product R12 PIM Connector.
- The *Oracle Enterprise Data Quality for Product Data Oracle DataLens Server Installation Guide* provides detailed Oracle DataLens Server installation instructions.
- The *Oracle Enterprise Data Quality for Product Data Oracle DataLens Server Administration Guide* provides information about installing and managing an Oracle DataLens Server.
- The *Oracle Enterprise Data Quality for Product Data COM Interface Guide* provides information about installing and using the Oracle DataLens Server COM APIs.
- The *Oracle Enterprise Data Quality for Product Data Java Interface Guide* provides information about installing and using the Oracle DataLens Server Java APIs.
- The *Oracle Enterprise Data Quality for Product Data Glossary* provides definitions to commonly used Enterprise DQ for Product technology terms.
- The *Oracle Enterprise Data Quality for Product Data Application Studio Reference Guide* provides information about creating and maintaining Data Service Applications (DSAs).

- The *Oracle Enterprise Data Quality for Product Data AutoBuild Reference Guide* provides information about creating an initial data lens based on existing product information and data lens knowledge.
- The *Oracle Enterprise Data Quality for Product Data Knowledge Studio Reference Guide* provides information about creating and maintaining data lenses.
- The *Oracle Enterprise Data Quality for Product Data Governance Studio Reference Guide* provides information about creating and maintaining Data Service Applications (DSAs).

See the latest version of this and all documents listed at the Oracle Enterprise Data Quality for Product Data Documentation Web site at:

http://download.oracle.com/docs/cd/E20593_01/index.htm

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Oracle Enterprise Data Quality for Product Data Command Line Interface Guide, Release 5.6.2
E25662-01