EXL DQLENS™ Administrator Guide

Table of Contents

1 INTRODUCTION

1.1 EXL DQLENS™ OVERVIEW

EXL DQLENS™ is comprised of four distinct and complimentary modules that accelerate key aspects of the Data Quality Management lifecycle: Data Analysis, Rule Implementation, and Quality Monitoring.

1.1.1 EXL DQLENS™ PROFILE

Analyzes data sets and reports basic statistics

EXL DQLENS™ Profile is a data profiling tool. When a data set is not well-understood, EXL DQLENS™ Profile can be used to produce summary-level statistics that characterize the data and may uncover potential data quality issues.

1.1.2 EXL DQLENS™ CLEANSE

Validates and standardizes data elements

EXL DQLENS™ Cleanse is a rules-based data cleansing tool. Once the characteristics of a data set are understood, EXL DQLENS™ Cleanse applies business rules that standardize and validate data to ensure consistency and fit for use.

1.1.3 EXL DQLENS™ RESOLVE

Identifies real-world entities across multiple data sets

EXL DQLENS™ Resolve is a matching (or "entity resolution") tool. When data is duplicated within or across data sets, this data must often be linked or harmonized to enable analytics and eliminate redundant and inconsistent data. EXL DQLENS™ Resolve identifies

The real-world entities (for example, Customers) represented within these data sets using advanced probabilistic and machine learning techniques.

1.1.4 EXL DQLENS™ MONITOR

Visualizes data quality metrics and trends over time

EXL DQLENS™ Monitor is a data quality dashboard. Once measurable data quality KPIs have been defined, and quality improvement rules implemented, EXL DQLENS™ Monitor provides the business direct insight into data quality metrics and trends over time.

Underpinning these functional modules is a layer of shared services that unify the platform. EXL DQLENS™ Core provides shared functionality and internal APIs that EXL DQLENS™ modules hook into to enable centralized administration and usage of the EXL DQLENS™ platform.

1.2 DOCUMENT PURPOSE

The purpose of this document is to provide an overview of the administrative aspects of the solution, including installation/uninstallation logging, job control & monitoring, and connection management. This document is not a comprehensive guide of all possible usage and configuration options.

1.3 SUPPORTED PLATFORMS

EXL DQLENS™ is currently able to connect to and profile tables from three database sources, Oracle, Microsoft SQL Server, and Snowflake.

EXL DQLENS™ has been tested on the versions that follow. The queries generated by the tool are quite straightforward and are likely to work on older versions and perhaps different platforms.

1.4 PREREQUISITES

Before using the tool, you must have a valid EXL DQLENS™ installation and a configured database connection.

DATABASE	VERSION
Oracle	20.2.0
SQL Server	18.6
Snowflake	4.42.2

2 EXL DQLENS™ ADMINISTRATION

2.1 INSTALLATION AND UNINSTALLATION

2.1.1 PREREQUISITES

EXL DQLENS™ is executed from command-line or terminal and will require the ability to connect to databases as needed and create local files and directories to store configuration and results.

2.1.2 INSTALLATION

EXL DQLENS™ is delivered in a self-extracting archive. When executing you will specify the install to a directory.

2.1.3 REMOVAL

To remove EXL DQLENS™ simply delete the installation directory.

Note that any configuration, logs and other files will remain in the EXL DQLENS™_HOME directory. This directory defaults to .EXL DQLens™ in the executing user's home directory.

2.2 CONNECTION MANGEMENT

EXL DQLENS™ manages connections to supported database platforms. These connections can then be referenced and used by EXL DQLENS™ modules to read data from, and write data to, the database.

2.2.1 JOB MONITORING AND TERMINATION

EXL DQLENS™ monitors the status of running processes and maintains a history of these jobs. You can view the job type, status, and any started tasks with their unique process ID and its status. If there is no process currently running, Core will return a message stating that no Core process is currently running?

Example of checking the status of a process:

```
> EXL DQLens™ core status

> Enter the filter criteria for Status:(full, job_type, status,thon -m EXL DQLens™ core
status   started) Status

Output:
EXL DQLENS™ Core process [1234] is running
```

2.2.2 TERMINATING

If you wish to terminate an existing process, you can cancel a running process by running the terminate command within Core. If a desired process exists, Core will terminate the process and remove any files associated with the process.

Example of terminating a process:

```
EXL DQLens™ core kill
Enter the module name: Profile
```

2.2.3 LOGGING

Translations recursively replace all instances of source values with the corresponding target value.

When staging a translation, EXL DQLENS™ will automatically create a user-defined function (EXL DQLENS™_TX()) in the database that facilitates this.

2.3 LOGGING

All operations performed by EXL DQLENS™ are logged in files stored in the EXL DQLENS™_HOME/logs folder. EXL DQLENS™ uses native Python logging configured by 'log.conf' in the EXL DQLENS™_HOME/config folder. There are several different log messages that can occur during any of EXL DQLENS™'s modules:

2.3.1 DEBUG

Debug messages contain information that is used to help people diagnose or debug an error

2.3.2 CRITICAL

A critical message occurs when an error is forcing a shutdown of the service or application to prevent data loss (or further data loss).

2.3.3 ERROR

An error message indicates that an error which is fatal to the operation, but not the service or application has occurred.

WARNING

A warning message lists anything that can potentially cause application oddities, but for which I am automatically recovering.

2.3.4 INFO

Info messages are generally useful information to log (service start/stop, configuration assumptions, etc.)

Each EXL DQLENS™ module has its own log file within the logs folder and lists the messages in descending order, meaning the oldest message will appear first and the newest message will appear last. Each message is timestamped with the date and time that the message was sent out and which module the message came from. For more information about Python logging, please refer to https://docs.python.org/3/library/logging.html

Example of a log:

```
'12/09/2020 02:29:33 PM' - cleanse - INFO - EXL DQLENS™ cleanse process [15100] is running.
'12/30/2020 11:12:11 AM' - core - ERROR - Connection 'snowt' not found!
```

2.4 COMMAND LINE OPERATIONS

A command line interface (CLI) provides a way for a user to interact with a program running in a text-based shell interpreter. A CLI serves as alternative to running a python file from the file itself and allows the user to enter a single line of text instructing the program which file to run and what variables to input.

2.4.1 CORE COMMAND LINE ARGUMENTS

COMMAND LINE INPUT	DESCRIPTION
EXL DQLens™ core conn list	To view all configured connections available
EXL DQLens™ core conn add	To add a new connection
EXL DQLens™ core conn update <connection name>	To update an existing connection
EXL DQLens™ core conn delete <connection name>	To delete a connection
EXL DQLens™ core status	To view the job type, status, or when the proves was started
EXL DQLens™ core kill	To terminate a select process

2.4.2 CONNECTION ATTRIBUTE

Oracle

ATTRIBUTES	DESCRIPTION
type	Type of database
user_name	Oracle account username
password	Oracle account password
host	The host name or IP address of the computer that is running Oracle Database XE
port	The port number on which the Oracle Net listener is listening. If not specified, the default port number 1521 is assumed.
sid	The sid is the unique name of the instance (e.g. the oracle process running on the machine)

Microsoft SQL Server Connection Attributes

ATTRIBUTES	DESCRIPTION
type	Type of database
user_name	MSSQL account username

password	MSSQL account password
server_name	Name of the server that is being used
database	Name of the database that is being used
schema	Name of the schema being used
integrated_security	A True/False input where "True" means that the connection uses Windows authentication and "False" means that Windows authentication is not being used.

Snowflake Connection Attributes

ATTRIBUTE	DESCRIPTION
type	Type of database
user_name	Snowflake account username
password	Snowflake account password
account	Name of the account being used
warehouse	Name of the warehouse that is being used
database	Name of the database being used
schema	Name of the schema being used

EXL (NASDAQ: EXLS) is a leading operations management and analytics company that designs and enables agile, customer-centric operating models to help businesses enhance revenue growth and profitability. Our delivery model provides market-leading business outcomes using EXL's proprietary Digital EXLerator™ Framework, cutting-edge analytics, digital transformation and domain

expertise. At EXL, we look deeper to help companies improve global operations, enhance data- driven insights, increase customer satisfaction, and manage risk and compliance. EXL serves the insurance, healthcare, banking and financial services, utilities, travel, transportation and logistics industries. Headquartered in New York, New York, EXL has more than 29,000 professionals in locations throughout the United States, Europe, Asia (primarily India and Philippines), Latin America, Australia and South Africa.

This article is not intended to constitute legal, compliance, regulatory, privacy or similar professional advice, and EXL does not provide services to clients in those areas. The reader is advised to engage experts to provide professional advice on any legal, compliance, regulatory or privacy topics covered herein. For more information, see www.exlservice.com/legal-disclaimer