EC PL/SL Toolkit v25.6 - Installation Guide

Author: Philippe Debois (European Commission - DIGIT)

Table of Contents

1. Audience

This guide is intended for operations teams who will deploy the EC PL/SQL toolkit into one or several database schemas. It gives a short description of the various tools of the toolkit and explains how to quickly install or upgrade them using the Data Base Migration tool (DBM).

2. Introduction

The EC PL/SQL toolkit is a suite of PL/SQL tools or utilities that have been developed within the Commission, by database experts, for database developers. They have been developed over the years to fulfil concrete needs of real information system development projects.

Here follows the list of tools included in the toolkit together with a short description and link(s) to their documentation.

Core Utilities

Tool	Name	Short Description	Manual(s)
DBM	Data Base Migration Utility	Database migration tool similar to Liquibase and Flyway but more database developer oriented.	User's Guide
DOC	Mail Merge Utility	Produce documents based on templates and access data based on the DAPL language.	User's Guide
DPP	Data Pump Utility	Add-on to the Oracle Data Pump utility.	User's Guide
DS	Data Set Utility	Data sub-setting and transportation, Sensitive data discovery and masking, Synthetic data generator, Change data capture.	Data Masking / Data Subsetting / Synthetic Data Generation / Reference Manual / Demo Guide
GEN	Code Generator	PL/SQL code generator to boost developer's productivity and industrialise solutions.	User's Guide
QC	Quality Check Utility	Check database design best practices amongst which naming conventions of database objects.	User's Guide

Supporting Utilities

Tool	Name	Short Description	Manual(s)
ARM	Archive Management Utility	Utility to pack several files into a single one to ease data transportation (used by the DPP).	User's Guide

Tool	Name	Short Description	Manual(s)
DDL	DDL Utility	Miscellaneous useful DDL routines.	none
LOG	Logging Utility	Library to log error, warning, information, and debug messages.	none
MAIL	Mailing Utility	Send emails with attachments from within the database.	none
SEC	Security Utility	Solution to store password securely into the database.	User's Guide
SQL	SQL Utility	Miscellaneous useful SQL routines.	none
ZIP	Zip Utility	Read and write standard zip files.	none

3. Quick start

This section explains how to quickly install or upgrade the above described tools in a central schema of a database and how to expose them to other schemas using public synonyms and grants.

3.1. Pre-requisites

3.1.1. Database Platform

The EC PL/SQL toolkit requires an Oracle database version 19c or above. Note that it has not been tested on Oracle 21c and 23ai yet.

The database can be hosted in a Data Centre (DC), in a Cloud on Prem (CoP), or in the Amazon cloud (AWS).

All files are UTF-8 so NLS_LANG should be set accordingly via shell commands like `export NLS_LANG=.UTF8` on Linux or `set NLS_LANG=.UTF8` on Windows.

3.1.2. Installer Platform

The installer (the DBM tool) requires the following:

- A MS Windows OR a Linux machine (to execute Windows or Linux shell scripts).
- A file system to unzip the supplied archive or clone the git repository.
- SQL*Plus (any version compatible with the target database).
- A connection to the database via SQL*Net or JDBC.

3.1.3. System privileges

Depending on the tool and the database platform (DC, CoP, AWS), the schema where the toolkit is installed must have the following privileges or system privileges. Note that the DBM tool will check the grant of those privileges prior to proceeding with the installation.

Tool	DB Platform	System privileges
ALL	ALL	CREATE PUBLIC SYNONYM, DROP PUBLIC SYNONYM
ARM	AWS	EXECUTE ON RDS_FILE_UTIL.
DPP	AWS	ADMINISTER DATABASE TRIGGER, ADVISOR, ALTER DATABASE LINK, ALTER PUBLIC DATABASE LINK, ALTER USER, CHANGE NOTIFICATION, CREATE ANY CONTEXT, CREATE ANY PROCEDURE, CREATE DATABASE LINK, CREATE JOB, CREATE PROCEDURE, CREATE SEQUENCE, CREATE SESSION, CREATE SYNONYM, CREATE TABLE, CREATE TRIGGER, CREATE VIEW, DEBUG CONNECT SESSION, DEQUEUE ANY QUEUE, DROP ANY CLUSTER, DROP ANY CONTEXT, DROP ANY DIMENSION, DROP ANY INDEX, DROP ANY INDEXTYPE, DROP ANY LIBRARY, DROP ANY MATERIALIZED VIEW, DROP ANY OPERATOR, DROP ANY PROCEDURE, DROP ANY SEQUENCE, DROP ANY SQL PROFILE, DROP ANY SYNONYM, DROP ANY TABLE, DROP ANY TRIGGER, DROP ANY TYPE, DROP ANY VIEW, ENQUEUE ANY QUEUE, EXECUTE ANY PROCEDURE, EXEMPT ACCESS POLICY, EXEMPT IDENTITY POLICY, EXEMPT REDACTION POLICY, FLASHBACK ANY TABLE, GRANT ANY OBJECT PRIVILEGE, MANAGE ANY QUEUE, MANAGE SCHEDULER, RESTRICTED SESSION, SELECT ANY DICTIONARY, SELECT ANY TABLE, UNLIMITED TABLESPACE.
	CoP	ADMINISTER DATABASE TRIGGER, ADVISOR, ALTER DATABASE LINK, ALTER PUBLIC DATABASE LINK, ALTER USER, CHANGE NOTIFICATION, CREATE ANY CONTEXT, CREATE ANY PROCEDURE, CREATE DATABASE LINK, CREATE JOB, CREATE PROCEDURE, CREATE SEQUENCE, CREATE SESSION, CREATE SYNONYM, CREATE TABLE, CREATE TRIGGER, CREATE VIEW, DEBUG CONNECT SESSION, DEQUEUE ANY QUEUE, DROP ANY CLUSTER, DROP ANY CONTEXT, DROP ANY DIMENSION, DROP ANY INDEX, DROP ANY INDEXTYPE, DROP ANY LIBRARY, DROP ANY MATERIALIZED VIEW, DROP ANY OPERATOR, DROP ANY

Tool	DB Platform	System privileges
		PROCEDURE, DROP ANY SEQUENCE, DROP ANY SQL PROFILE, DROP ANY SYNONYM, DROP ANY TABLE, DROP ANY TRIGGER, DROP ANY TYPE, DROP ANY VIEW, ENQUEUE ANY QUEUE, EXECUTE ANY PROCEDURE, EXEMPT ACCESS POLICY, EXEMPT IDENTITY POLICY, EXEMPT REDACTION POLICY, FLASHBACK ANY TABLE, GRANT ANY OBJECT PRIVILEGE, MANAGE ANY QUEUE, MANAGE SCHEDULER, RESTRICTED SESSION, SELECT ANY DICTIONARY, SELECT ANY TABLE.
	DC	ADVISOR, ALTER DATABASE LINK, CREATE ANY CONTEXT, CREATE ANY PROCEDURE, CREATE DATABASE LINK, CREATE JOB, CREATE PROCEDURE, CREATE SEQUENCE, CREATE SESSION, CREATE SYNONYM, CREATE TABLE, CREATE TRIGGER, CREATE VIEW, DEBUG CONNECT SESSION, DEQUEUE ANY QUEUE, DROP ANY CLUSTER, DROP ANY CONTEXT, DROP ANY DIMENSION, DROP ANY INDEX, DROP ANY MATERIALIZED VIEW, DROP ANY OPERATOR, DROP ANY PROCEDURE, DROP ANY SEQUENCE, DROP ANY SQL PROFILE, DROP ANY SYNONYM, DROP ANY TABLE, DROP ANY TRIGGER, DROP ANY TYPE, DROP ANY VIEW, ENQUEUE ANY QUEUE, EXECUTE ANY PROCEDURE, MANAGE ANY QUEUE, MANAGE SCHEDULER, RESTRICTED SESSION, SELECT ANY DICTIONARY, SELECT ANY TABLE.
SEC	ALL	EXECUTE ON DBMS_CRYPTO, UTL_ENCODE, UTL_I18N, UTL_RAW.
MAIL	ALL	EXECUTE ON UTL_SMTP, UTL_MAIL, UTL_TCP, UTL_RAW.

3.2. Installation procedure

3.2.1. Create the target schema

Create the central schema where the toolkit will be installed and grant it all the privileges described above. The tables and indexes of the DBM tool will be created in the default tablespace. If necessary, modify the tablespace settings accordingly.

3.2.2. Unzip the archive or clone the repository

Unzip the distributed archive (`ec-plsql-toolkit.zip`) into the folder of your choice. This action will create a sub-folder named `ec-plsql-toolkit` .

Alternatively, clone the git repository via the following commands:

```
mkdir ec_plsql_toolkit
cd ec_plsql_toolkit
git clone "https://<path-to-toolkit-repo>.git"
```

3.2.3. Set environment variables

Define the following environment variables as appropriate.

Name	Description	Mandatory?
DBM_USERNAME	Schema username	Y
DBM_PASSWORD	Schema password	Y
DBM_DATABASE	Schema database	Y
DBA_USERNAME	CoP admin username	N
DBA_PASSWORD	CoP admin password	N
DBA_DATABASE	CoP admin database	N
DBM_CONF_PATH	Path of the configuration file	N
DBM_APPS_DIR	Alternate location for apps dir	N
DBM_LOGS_DIR	Alternate location for logs dir	N
DBM_TMP_DIR	Alternate location for tmp dir	N

The first three environment variables that must always be defined allow the installer to connect to the schema in which the toolkit must be installed. The DBM tool will connect to the schema using the following connect string: `"$DBM_USERNAME/$DBM_PASSWORD@$DBM_DATABASE"` .

The following three DBA environment variables are used by the installer to connect to the admin account of a Cloud on Prem database for the purpose of creating and dropping restore points, as well as for flash-backing the database in the event of a migration failure. This optional feature is exclusively accessible in Cloud on Prem databases.

Other environment variables allow you to define an alternate location for the configuration file (`conf/dbm_utility.conf` by default), the application folder (`apps` by default), the folder where logs are stored (`logs` by default), and the folder where temporary files are created (`tmp` by default).

Environment variables can be defined using the following syntax:

- Under Windows: `C:> set VARIABLE=value`
- Under Linux: `$ export VARIABLE=value`

3.2.4. Install or upgrade the DBM tool

The DBM tool, which is the installer of the EC PL/SQL toolkit, must be first installed and/or upgraded by executing the `migrate-dbm` script while being located in the `ec_plsql_installer` folder.

This script will install and/or upgrade the DBM tool to its latest available version. Tables, views, and packages of the DBM tool created in the schema can be easily identified by their name, all prefixed with `DBM_`.

Due to the dependencies between its shell scripts and its database objects, this upgrade is mandatory before launching the DBM client. It is recommended to do it each time a new archive of the toolkit is deployed.

3.2.5. Interactive install or upgrade of the toolkit

To install or upgrade the toolkit, execute the `dbm-cli` shell script while being located in the `ec-plsql-toolkit` folder. The `DBM-CLI>` prompt indicates that the tool is ready to receive commands.

First, execute the `migrate` command to install and/or upgrade all tools of the toolkit. This single command installs the tools that are not installed yet (e.g., upon first invocation), and upgrades those that are possibly already present.

Then execute the `expose` command to expose the tools to other schemas i.e., to create public synonyms and grant necessary privileges. When done, type `exit` to leave SQL*Plus.

```
DBM-CLI> @dbm-cli migrate
DBM-CLI> @dbm-cli expose
DBM-CLI> exit
```

Note: all dbm-cli commands must be passed to the `dbm-cli.sql` SQL script using the `@`, `@@`, or `start` command of SQL*PLus. If you want to expose only a single tool, pass its name as a parameter of the `expose` command (e.g., `expose dpp_utility`). To expose several tools, repeat this command as necessary.

See the DBM Utility User's guide for a more detailed description of the various available commands.

3.2.6. Automated install or upgrade of the toolkit

To install or upgrade the toolkit from a pipeline, simply pass the command to execute as a parameter of the `dbm-cli` shell script:

```
dbm-cli migrate
dbm-cli expose
```

You can even execute both operations with a single command line:

```
dbm-cli migrate / expose
```

Upon completion, the exit code of the `dbm-cli` shell script will be set to zero in case of success and non-zero in case of failure. You can check the following variables depending on your operating system:

- For Windows: `%ERRORLEVEL%`
- For Linux: `$?`

3.2.7. Managing Migration Failure

For Cloud on Prem databases, the DBM tool provides a built-in feature for managing migration failures through the "flashback database to restore point" functionality. To utilise this feature:

1. Before executing the `migrate` command, run the `create-restore-point` command to create a restore point.
2. In the event of a migration failure, execute the `flashback-database` command to rollback the database to the previously created restore point.
3. If necessary, restore points can be dropped using the `drop-restore-point` command.

Alternatively, it is recommended to export the schema containing the toolkit before proceeding with its migration. This allows for the restoration of the schema in case of failure. It's important to note that uninstalling and re-installing the tools could result in the loss of end-users' data stored in internal tables of each tool.

In future releases, rollback scripts might also be provided for the migration of each tool, allowing for the rollback of failed migrations without any risk of data loss. However, such scripts are not available at this time.

3.2.8. Log files

The output of each command displayed on the screen is also sent to a log file stored in the `logs` sub-folder (or in the folder specified by the `DBM_LOGS_DIR` environment variable).