



EDB

EDB Migration Portal

Release 3.5.2

EDB

Mar 07, 2022

CONTENTS

1	Release notes	3
1.1	Version 3.5.2	3
1.2	Version 3.5.1	3
1.3	Version 3.5.0	4
2	Supported platforms	5
2.1	Supported browsers	5
3	Using Migration Portal	7
3.1	Overview of the Migration Portal home page	8
3.2	Overview of the Migration Portal Projects page	10
3.3	Migration Portal quick start	11
4	Migrating a database	13
4.1	Performing a schema extraction	13
4.1.1	Extracting schemas to an SQL dump file	13
4.1.2	Schemas and objects support	15
4.2	Performing a schema assessment	17
4.2.1	Perform a schema assessment	17
4.2.2	Generate an assessment report	21
4.3	Schema migration	21
4.3.1	Migrating to an existing on-premises EDB Postgres Advanced Server host	21
4.3.2	Migrating to a new on-premises EDB Postgres Advanced Server installation	22
4.3.3	Migrate to the cloud	23
4.4	Data migration	25
5	Advanced data migration	27

Migration Portal is a web-based service for migrating Oracle database schemas to the EDB Postgres Advanced Server platform. The Migration Portal assesses and analyzes Oracle database schemas and converts:- Types- Tables- Sequences- Constraints- Triggers- Views- Stored procedures- Packages- Database links (dblinks)- Materialized views- Indexes

The migration produces DDLs compatible with EDB Postgres Advanced Server.

The user-friendly portal interface simplifies the assessment and migration process. Log in to the portal and start the [migration process](#).

EnterpriseDB has helped companies migrate their existing database systems to Postgres for years. For more information, see the [EnterpriseDB website](#).

RELEASE NOTES

The Migration Portal documentation describes the latest version of Migration Portal 3.5.

Version	Release Date
[3.5.2](08_mp_3.5.2_rel_notes)	2022 Mar 04
[3.5.1](09_mp_3.5.1_rel_notes)	2022 Feb 24
[3.5.0](10_mp_3.5.0_rel_notes)	2022 Feb 03

1.1 Version 3.5.2

New features, enhancements, bug fixes, and other changes in Migration Portal 3.5.2 include the following:

Type	Description
Bug Fix	Fixed an issue that was causing frequent outages for the web-based version of Migration Portal.

1.2 Version 3.5.1

New features, enhancements, bug fixes, and other changes in Migration Portal 3.5.1 include the following:

Type	Description
Enhancement	Added a validation to prevent the use of DDL files generated by the DDL Extractor previously provided by EDB. The current version of the Migration Portal requires the DDLs uploaded into the Migration Portal to be generated using the Oracle Data Pump utilities as documented in the [Performing a schema extraction](../04_mp_migrating_database/01_mp_schema_extraction) instructions. Note that disabling the use of DDL extractor generated files is expected to be a temporary measure for this release until issues that were encountered after the 3.5.0 release are fully addressed.
Security Fix	Migration Portal uses the EDB JDBC driver for communicating with the underlying database. The driver has been updated to version 42.3.2.1, which covers recent security fixes to address CVE-2022-21724.
Bug Fix	Fixed different parsing related errors encountered after the 3.5.0 release.

1.3 Version 3.5.0

New features, enhancements, bug fixes, and other changes in Migration Portal 3.5.0 include the following:

Type	Description
Enhancement	Migration Portal is now certified to support migrations to EDB Postgres Advanced Server 14.
Enhancement	Migration Portal now uses the metadata information in a SQL dump file to assess the extracted schemas. You no longer should download the DDL extractor script to extract DDLs for your schemas.
Feature	Updated repair handler to remove OVERFLOW syntax from CREATE TABLE syntax to make it compatible with EDB Postgres Advanced Server compatible syntax. (ERH-2011: ORGANIZATION_INDEX_COMPRESS).
Feature	Added a repair handler to remove the NAME clause and associated label name from SET TRANSACTION statements. (ERH-3002: SET_TRANSACTION).
Enhancement	Improved the user interface for a better user experience.

SUPPORTED PLATFORMS

The Migration Portal supports assessment and migration from Oracle 11g, 12c, 18c, and 19c to EDB Postgres Advanced Server 10, 11, 12, 13, or 14. Migration Portal is supported on the following browsers.

2.1 Supported browsers

For the best user experience, we recommend using the Google Chrome browser. Migration Portal is also supported on the following browsers:

Browser	Supported version
Apple Safari on Macintosh OS	14 and above
Google Chrome	68 and above
Microsoft Edge	42 and above
Mozilla Firefox	60 and above

USING MIGRATION PORTAL

Migration Portal allows you to easily migrate your database schema from Oracle to EDB Postgres Advanced Server. You can upload schemas for assessment and get immediate feedback and suggestions. The portal allows you to download assessed DDLs for all objects and create your EDB Postgres database on premises or in the cloud.

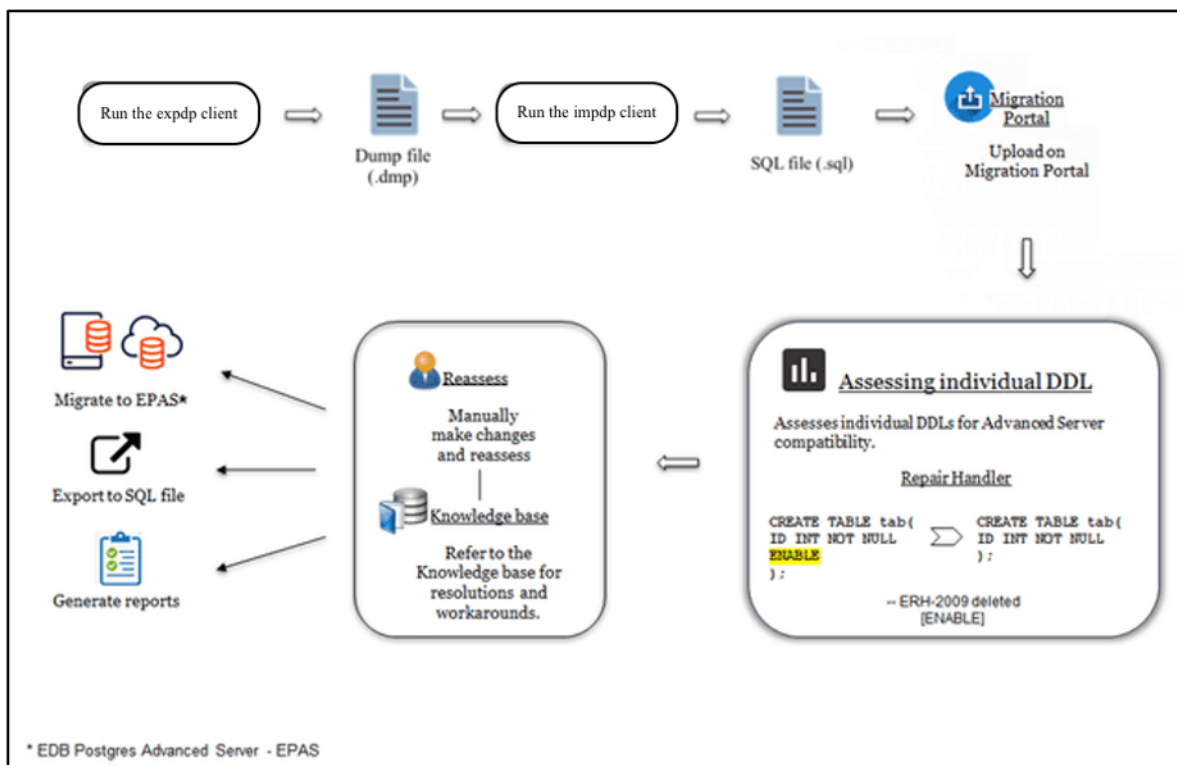


Fig. 1: The assessment and migration process

Note: Migration Portal currently doesn't have team collaboration features. We recommend collaborative meetings, screen sharing, and using the downloadable assessment report to share results.

To access the Migration Portal:

1. Open a browser and navigate to <https://www.enterprisedb.com/>.
2. On the EnterpriseDB home page, select **PostgreSQLSoftware>Migration**.
3. On the Migration Portal page, select **Findoutmore>OpenMigrationPortal**.

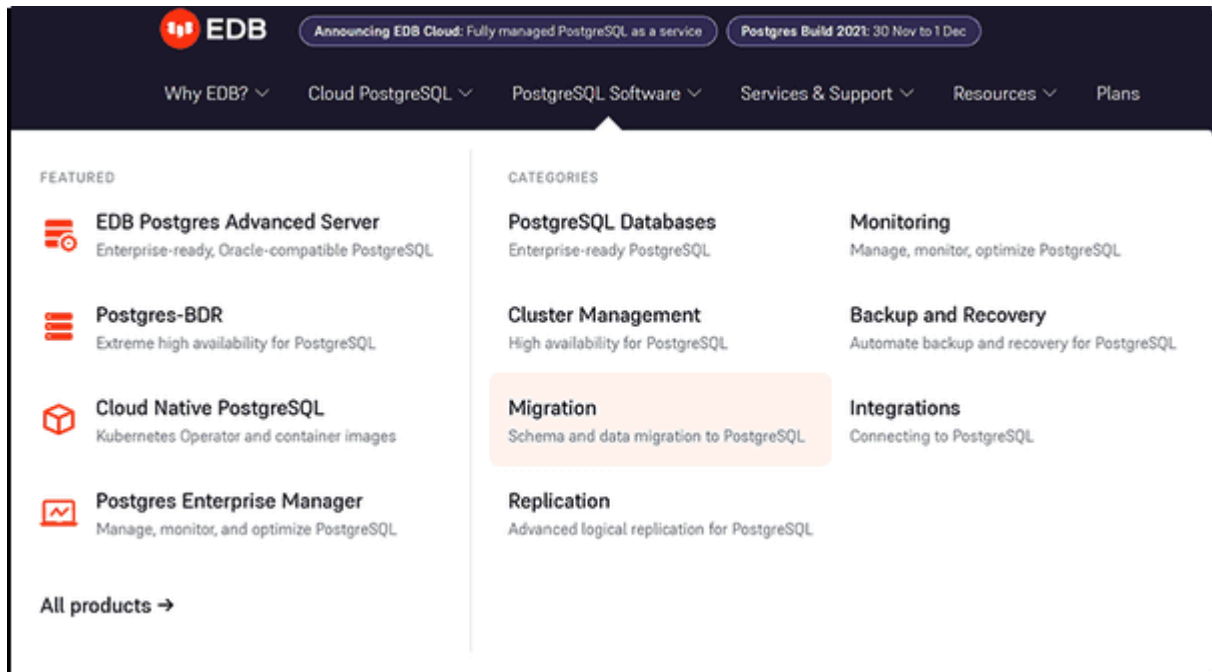


Fig. 2: Migration Portal on the EnterpriseDB home page

- Log in using your credentials. See [Migration Portal quick start](#) for an overview of using Migration Portal to migrate Oracle schemas.

3.1 Overview of the Migration Portal home page

The Migration Portal home page provides quick access to migration tools.

The Migration Portal home page allows access to the following Migration Portal features:

- Projects** : The Projects panel displays a **list** of assessed projects.
- Createproject** : Select **+New** to create a project.
- Searchprojects** : Use the **Search** box to search **for** projects.
- Overview** : The Overview panel provides details about the selected project **and** displays the compatibility percentage after schema assessment.
- Migrateto** : Select **Migrateto** either to download an EDB Postgres Advanced Server compatible `.sql`` file or migrate a schema to an EDB Postgres Advanced Server on premises or in the cloud. For more information, see [\[Schema migration\]\(./04_mp_migrating_database/03_mp_schema_migration/#mp_schema_migration\)](#).
- Report** : Select **Report** to view **and** download the schema assessment report.

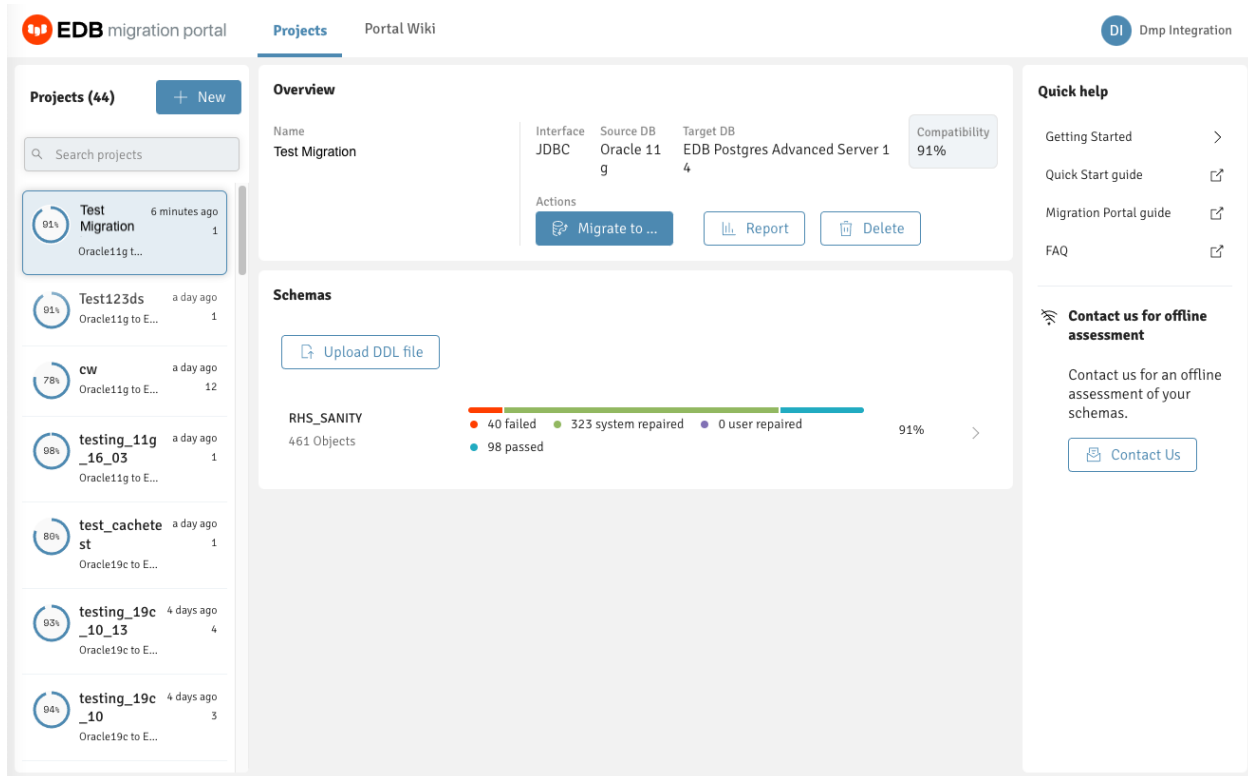


Fig. 3: The Migration Portal home page.

- **Delete** : Select **Delete** to delete a selected project.
- **Warningsign** : A warning message appears if a project or a schema is less than 70 % compatible or any DDL doesn't succeed after multiple attempts.
- **UploadDDLfile** : Select **Uploadnewschema** to upload a new DDL file.
- **Schemas** : The Schemas panel displays the assessment result from an uploaded DDL file.
- **Quickhelp** : The Quick help panel contains quick links to a few required resources.
- **PortalWiki** : Select **PortalWiki** to access links to product information and more help guides.

3.2 Overview of the Migration Portal Projects page

The Migration Portal Projects page provides detailed information about your migration.

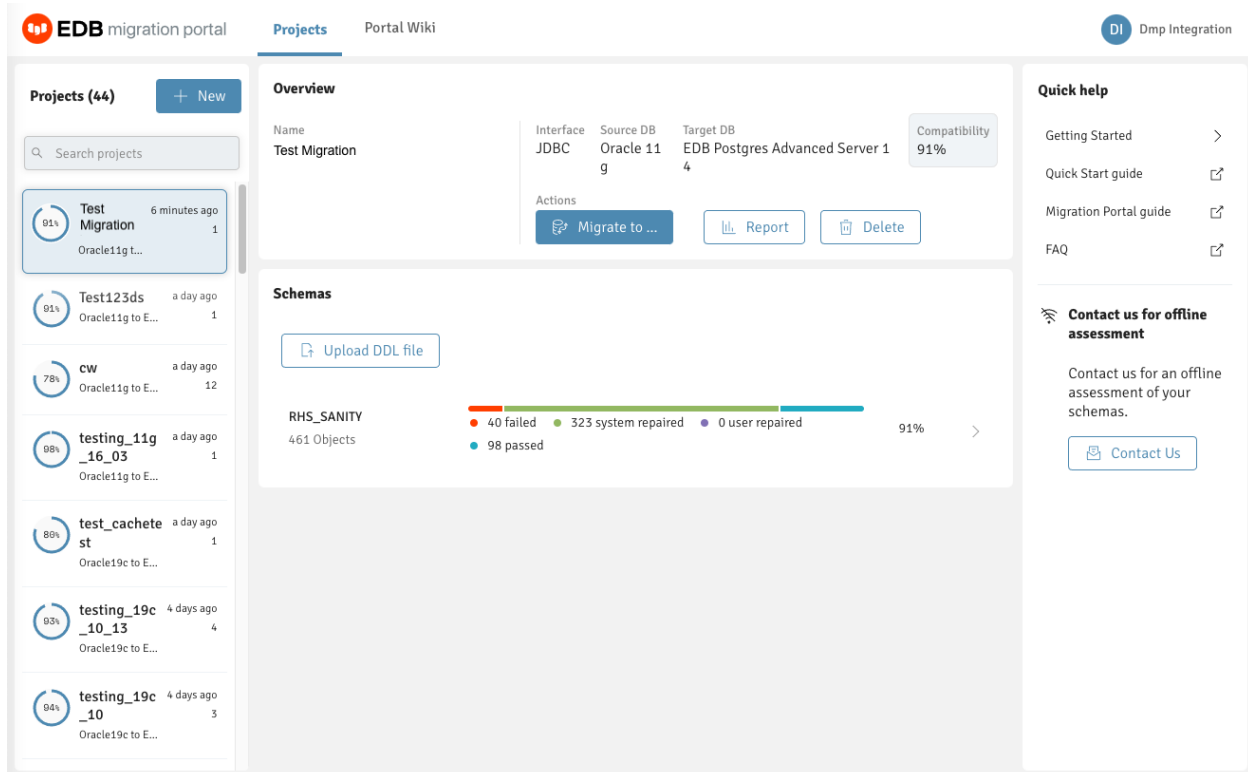


Fig. 4: The Migration Portal Projects Page overview.

Use the following resources to gather information about your migration projects:

- Compatible** : The **Compatible** gauge displays the color on the basis of the compatibility percentage of the assessed schema.
- SchemaCount** : Displays the number of schemas in a project.
- Migrateto** : Select **Migrateto** to migrate the schema to an EDB Postgres Advanced Server on premises or in the cloud.
- Searchobjects** : Use the **Search** box to search for objects.
- Filters** : From the left panel of the Projects page, you can filter the system repaired and manual repaired objects. You can select one or more filter combinations to refine the information.
- Objects** : Displays the objects for the selected schemas.
- CommonFailures** : Displays the reason for the failed objects for the selected schemas.

Note: You can download a `.csv` file for the common failures for the project.

- ****Schema**** : Displays the result of the assessment.
- ****Tooltip**** : Hover over a result `set` to display a tooltip `with` the number of `passed`, `failed`, `and` repaired objects.
- ****Quickhelp**** : The Quick help panel displays links to Knowledge Base articles `and` `repair handler documentation`.
- ****Search**** : Use the ****Search**** box to search the Knowledge Base entries `or` `repair handler documentation` `for` specific information.

3.3 Migration Portal quick start

To migrate Oracle schemas using Migration Portal:

1. Generate a `.sql` file with metadata information of your schemas using Oracle Data Pump's command line clients `expdp` and `impdp`. For more information, see [Performing a schema extraction](#).
2. Log in to [Migration Portal](#) with your EDB credentials.
3. Create a new project.
4. Upload the `.sql` file.

Note: Uploading large DDL files might freeze or crash the browser (depending on the browser and the system resources in use). In this scenario, try running Migration Portal in a different browser or extracting fewer schemas and uploading that file on Migration Portal.

1. Select **Create&assess** . Migration Portal reviews every construct, executes repair actions to improve compatibility with EDB Postgres Advanced Server, and flags any remaining errors that require manual intervention.
2. Verify the DDL objects; for example, TABLES that do not show a 100% success ratio.
3. Select the objects that are not compatible with EDB Postgres Advanced Server.
4. Refer to the Knowledge Base for checking the possible workarounds for the objects that are not compatible in EDB Postgres Advanced Server.
5. Make the required changes and select **Reassess** to rerun the analysis.
6. After resolving errors in your schemas, you can download the modified EDB compatible DDL file and use any client application such as pgAdmin or PSQL client and migrate the schema to EDB Postgres Advanced Server. You can also migrate to EDB Postgres Advanced Server on the cloud. For more information, see [Schema migration](#).

MIGRATING A DATABASE

To migrate a database, you must complete the following steps:

1. Perform a [schema extraction](#).
2. Perform a [schema assessment](#).
3. Perform a [schema migration](#).
4. Perform a [data migration](#).

4.1 Performing a schema extraction

You can use the Oracle's Data Pump utility to extract metadata from your Oracle database to an SQL dump file. Migration Portal requires this file to be uploaded to assess the extracted schemas.

Note: Migration Portal might become unresponsive for very large SQL files, depending on your system's and browser's resource availability. To resolve this, try extracting fewer schemas at a time to reduce the SQL file size.

4.1.1 Extracting schemas to an SQL dump file

Note that Migration Portal only requires the metadata in the SQL dump file to assess the extracted schemas.

Prerequisites

- If you plan on exporting schemas that are not your own, ensure that you are assigned the DATAPUMP_IMP_FULL_DATABASE role. Otherwise, you can only export your own schema.
- Ensure that you have sufficient tablespace quota to create objects in the tablespace.
- In SQL*Plus, create a directory object that points to a directory on your server file system. For example:

OnLinux :

```
SQL> create directory DMPDIR as '/sample/file/path';
```

OnWindows :

```
SQL> create directory DMPDIR as 'c:\sample\file\path\';
```

- In SQL*Plus, grant read, write permissions to the user running the Data Pump utility.

```
SQL> grant read,write on directory DMPDIR to <Username>;
```

Perform either of the following procedures:

- *To extract one or more schemas in a database*

- *To extract all schemas in a database*

To extract one or more schemas in a database

1. Before running the `expdp` command, create a file with a `.par` extension (for example, `export.par`) on your server. Add attributes and values to the file, as follows:

```
CONTENT=metadata_only

INCLUDE=SYNONYM, SEQUENCE, TYPE, TABLE, CONSTRAINT, PROCEDURE, FUNCTION, PACKAGE, INDEX,
↪ MATERIALIZED_VIEW, VIEW, TRIGGER, DB_LINK
```

The attributes and values in the above command specify the following options: + `CONTENT=metadata_only` extracts only the metadata of the schemas. + `INCLUDE=` specifies which database object types to include in the extraction. Append a comma-separated list of the *Supported object types* to only extract database object types that are supported by Migration Portal.

2. From the command line, run the export command to generate a **schemas_metadata.dump** file. For example, to extract metadata information for `<Schema_1>`, `<Schema_2>`, `<Schema_3>`, and so on, run:

```
expdp <Username>@<ConnectIdentifier> DIRECTORY=DMPDIR SCHEMAS=<Schema_1>,<Schema_2>,<Schema_3>
↪ DUMPFILE=schemas_metadata.dump parfile=export.par
```

3. To generate a SQL file from the dump file, run the import command.

For example, to generate **YourSchemas.SQL** file from the `schemas_metadata.dump` file, enter:

```
impdp <Username>@<ConnectIdentifier> DIRECTORY=DMPDIR TRANSFORM=OID:n_
↪ SQLFILE=YourSchemas.sql DUMPFILE=schemas_metadata.dump
```

To extract all schemas in a database

Note: Do not perform the following procedure from a user account that belongs to the excluded schemas list, see [Unsupported schemas](#unsupported-schemas). The `impdp` command fails if the user account running the command is in the excluded list of schemas.

1. Before running the `expdp` command, create a file with a `.par` extension (for example, `export.par`) on your server. Add attributes and values to the file, as follows:

```
CONTENT=metadata_only

FULL=Y

INCLUDE=SYNONYM, SEQUENCE, TYPE, TABLE, CONSTRAINT, PROCEDURE, FUNCTION, PACKAGE, INDEX,
↪ MATERIALIZED_VIEW, VIEW, TRIGGER, DB_LINK
```

The attributes and values in the above command specify the following options: + `CONTENT=metadata_only` extracts only the metadata of the schemas. + `INCLUDE=` specifies which database object types to include in the extraction. Append a comma-separated list of the *Supported object types* to only extract database object types that are supported by Migration Portal. + `FULL=Y` extracts all schemas in the database.

2. From the command line, run the export command to generate a **schemas_metadata.dump** file, as follows:

```
expdp <Username>@<ConnectIdentifier> DIRECTORY=DMPDIR DUMPFILE=schemas_metadata.dump_
↳parfile=export.par
```

- Before running the impdp command, create a parameter file with a .par extension (for example, import.par) on your server. Add attributes and values to the file, as follows:

```
EXCLUDE=SCHEMA:"IN(select username as schema_name from sys.dba_users where_
↳LOWER(username) LIKE 'pg_%' OR username IN
('ANONYMOUS','APEX_PUBLIC_USER','APEX_030200','APEX_040000','APEX_040200','APPQOSSYS',
↳'AUDSYS','CTXSYS','DMSYS','DBSNMP',
'DBSFWUSER','DEMO','DIP','DMSYS','DVF','DVSYS','EXFSYS','FLOWS_FILES','FLOWS_020100',
↳'FRANCK','GGSYS','GSMADMIN_INTERNAL',
'GSMCATUSER','GSMROOTUSER','GSMUSER','LBACSYS','MDDATA','MDSYS','MGMT_VIEW','OJVM SYS',
↳'OLAPSYS','ORDPLUGINS','ORDSYS',
'ORDDATA','OUTLN','ORACLE_OCM','OWBSYS','OWBYSS_AUDIT','PDBADMIN','RMAN','REMOTE_
↳SCHEDULER_AGENT','SI_INFORMTN_SCHEMA',
'SPATIAL_CSW_ADMIN_USR','SPATIAL_WFS_ADMIN_USR','SQLTXADMIN','SQLTXPLAIN','SYS$UMF','SYS
↳','SYSBACKUP','SYSDG','SYSKM',
'SYSRAC','SYSTEM','SYSMAN','TMSYS','WKPROXY','WKSYS','WK_TEST','WMSYS','XDB','XS$NULL'))
↳"
```

The attributes and values in the above command specify the following options: + **EXCLUDE** specifies which schemas to exclude. Append a list of the *Unsupported schemas* to ensure that the Data Pump utility *does not* extract any system schemas (including schemas beginning with **pg_**).

- To generate a SQL file from the dump file, run the import command.

For example, to generate **YourSchemas.SQL** file from the schemas_metadata.dump file, enter:

```
impdp <Username>@<ConnectIdentifier> DIRECTORY=DMPDIR TRANSFORM=OID:n_
↳SQLFILE=YourSchemas.sql DUMPFILE=schemas_metadata.dump parfile=import.par
```

4.1.2 Schemas and objects support

See the following sections for supported and unsupported schemas and objects.

See the following table for a list of Oracle systems schemas you must exclude while generating the SQL dump file.

1	2	3	4
ANONYMOUS	APEX_PUBLIC_USER	APEX_030200	APEX_040000
APPQOSSYS	AUDSYS	CTXSYS	DMSYS
DBSFUSER	DEMO	DIP	DMSYS
DVSYS	EXFSYS	FLOWS_FILES	FLOWS_020100
GGSYS	GSMADMIN_INTERNAL	GSMCATUSER	GSMROOTUSER
LBACSYS	MDDATA	MDSYS	MGMT_VIEW
OLAPSYS	ORDPLUGINS	ORDSYS	ORDDATA
ORACLE_OCM	OWBSYS	OWBYSS_AUDIT	PDBADMIN
REMOTE_SCHEDULER_AGENT	SI_INFORMTN_SCHEMA	SPATIAL_CSW_ADMIN_USR	SPATIAL_WFS_ADMIN_USR
SQLTXPLAIN	SYS\$UMF	SYS	SYSBACKUP
SYSKM	SYSRAC	SYSTEM	SYSMAN
WKPROXY	WKSYS	WK_TEST	WMSYS
XS\$NULL			

Note: EDB Postgres Advanced Server does not support schema names starting with **pg_**, which should also be excluded while extracting from Oracle.

-
-
-
-
-
-
-
-
-
-
-
-
-

Note: COMMENTS on columns, tables, and materialized views are also supported.

-
-
-
-
-
-

- Role **and** object grants

- Tablespaces

- Directories

- Users

- RLS policy

- Queues

4.2 Performing a schema assessment

You can assess an Oracle database schema for compatibility with EDB Postgres Advanced Server. You can also generate a report about the assessment.

4.2.1 Perform a schema assessment

1. Navigate to the [Migration Portal](#).
2. Enter your EDB credentials.
3. Select **New** to create a project.
4. In the dialog box, enter the project name in the **Projectname** field.
5. Specify project details:
 - Select the application interface. - Select a source DB and version for the source. - Select the target DB and Version for the target.
6. To upload the .SQL file generated by the Data Pump utility, select **Choosefile**.

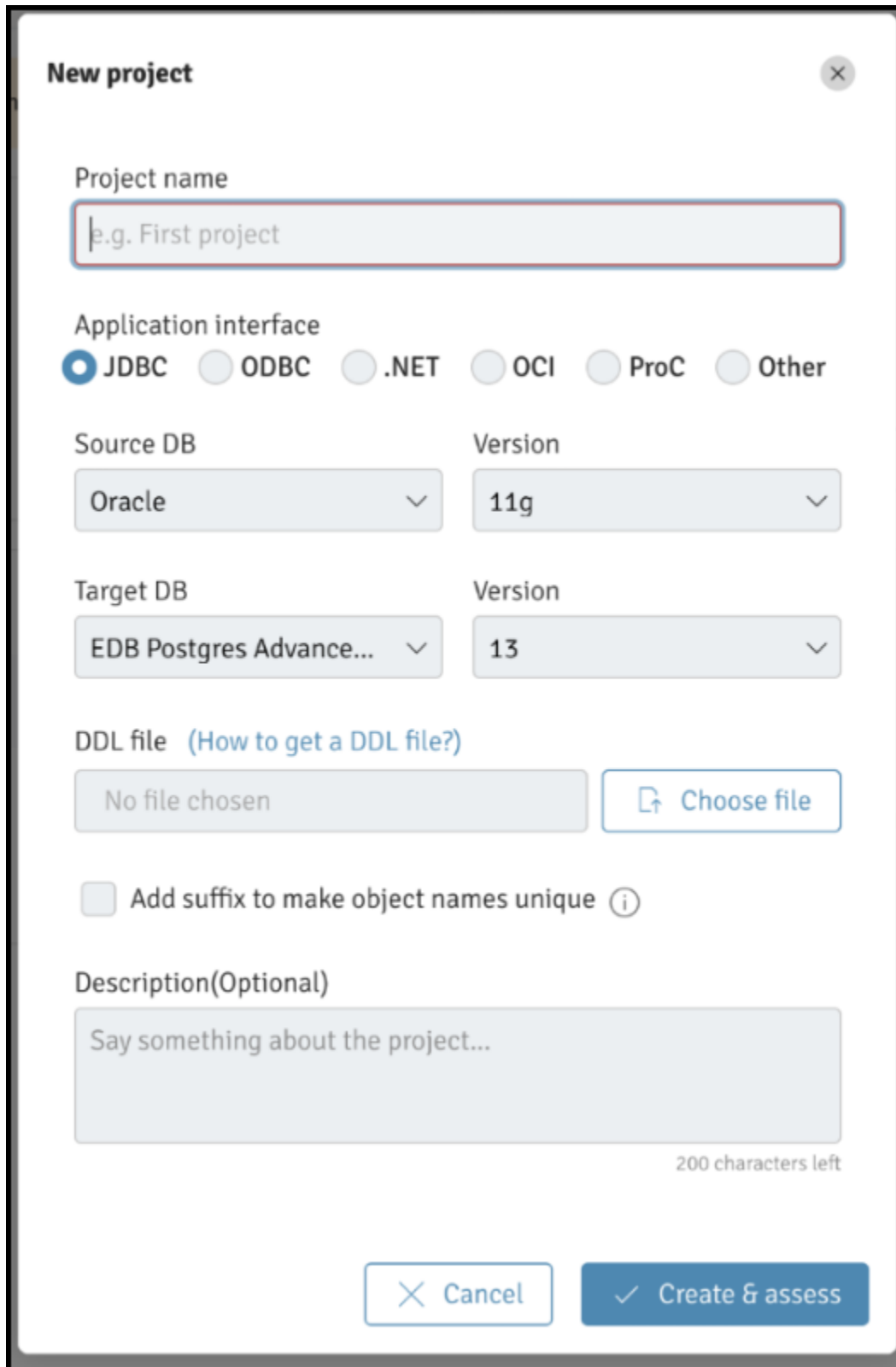
Note:

- Don't modify the .SQL file.
-

7. To specify adding a suffix to indexes (**_IDX**) and constraints (**_CON**) when creating a project, select the **Addsuffixtomakeobjectnamesunique** checkbox. This selection ensures better assessment results, as EDB Postgres Advanced Server doesn't support the same name for tables and indexes, whereas, Oracle does.
8. Select **Create&assess** to create a new project and to assess the schemas.

The analysis tool reviews every construct and executes repair actions to improve compatibility with EDB Postgres Advanced Server. It flags any remaining errors that require manual intervention.

9. Verify the DDL objects (e.g., TABLES) that don't show a 100% success ratio.
10. Select the objects that aren't compatible with EDB Postgres Advanced Server and view the details. At the bottom of the middle panel, you can view repair action details.

A screenshot of the 'New project' dialog box in the EDB Migration Portal. The dialog has a title bar with a close button. It contains several sections: 'Project name' with a text input field containing 'e.g. First project'; 'Application interface' with radio buttons for 'JDBC' (selected), 'ODBC', '.NET', 'OCI', 'ProC', and 'Other'; 'Source DB' and 'Version' dropdowns with 'Oracle' and '11g' selected; 'Target DB' and 'Version' dropdowns with 'EDB Postgres Advance...' and '13' selected; 'DDL file' with a text input showing 'No file chosen' and a 'Choose file' button; a checkbox for 'Add suffix to make object names unique' with an information icon; and a 'Description(Optional)' text area with the placeholder 'Say something about the project...' and a '200 characters left' indicator. At the bottom are 'Cancel' and 'Create & assess' buttons.

New project

Project name
e.g. First project

Application interface
☒ JDBC ☐ ODBC ☐ .NET ☐ OCI ☐ ProC ☐ Other

Source DB
Oracle

Version
11g

Target DB
EDB Postgres Advance...

Version
13

DDL file [\(How to get a DDL file?\)](#)
No file chosen [Choose file](#)

☐ Add suffix to make object names unique ⓘ

Description(Optional)
Say something about the project...
200 characters left

[Cancel](#) [Create & assess](#)

Fig. 1: The Migration Portal New project dialog.

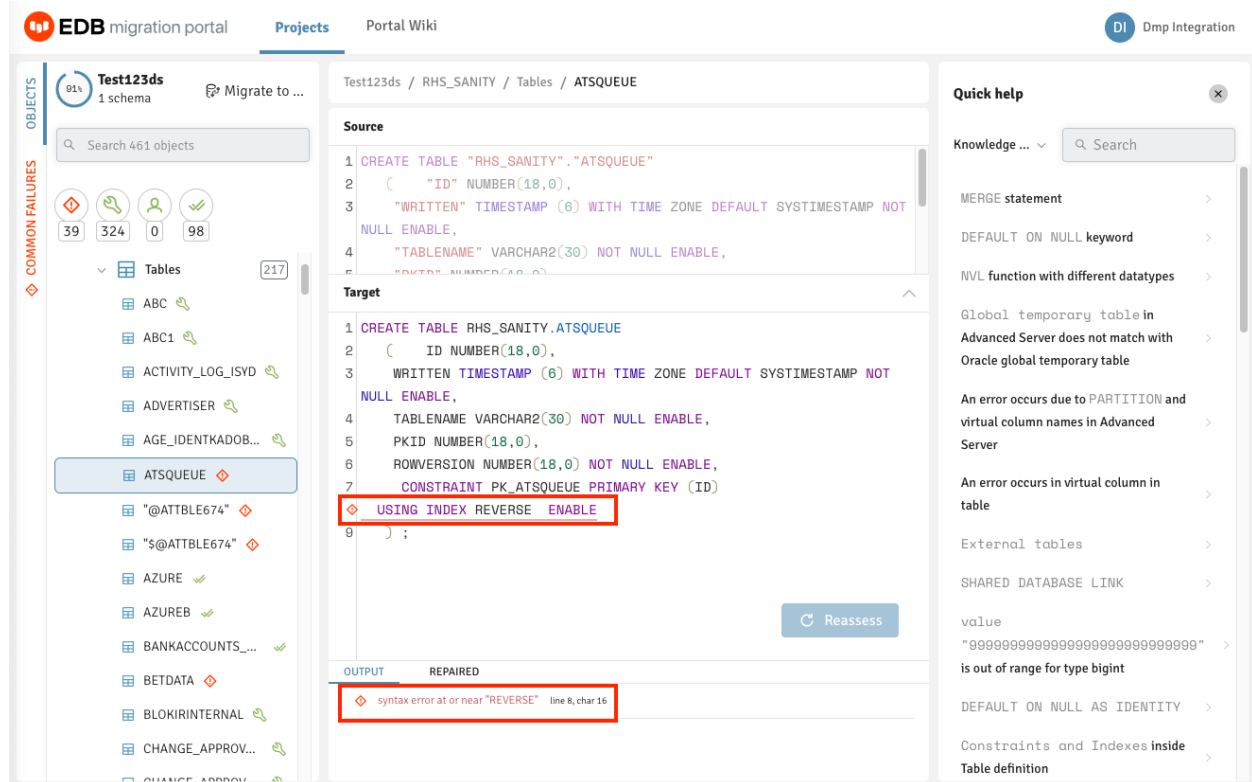


Fig. 2: Incompatible objects are identified.

11. Refer to the Knowledge Base information in the right panel to locate the possible workarounds for the objects that aren't immediately compatible with EDB Postgres Advanced Server. You can also view the Knowledge Base information on the Portal wiki page.
12. On the **KnowledgeBase** tab, you can enter the error message for the incompatible objects with EDB Postgres Advanced Server and select **Search**.

The object detail panel displays the workaround or the resolution for the failed object. You can manually make the changes on the **Assessment** tab for that object and select **Reassess**.

Note: If any failed object passes while reassessing, the dependent objects for that object are also reassessed.

Similarly, you can make all incompatible objects compatible.

Note: If the information or workaround for incompatible objects isn't available in the Knowledge Base, contact the support team for help.

When you have finished working with the DDL, you can either download the modified EDB compatible DDL as a .sql file or migrate the schemas. See [Schema migration](#).

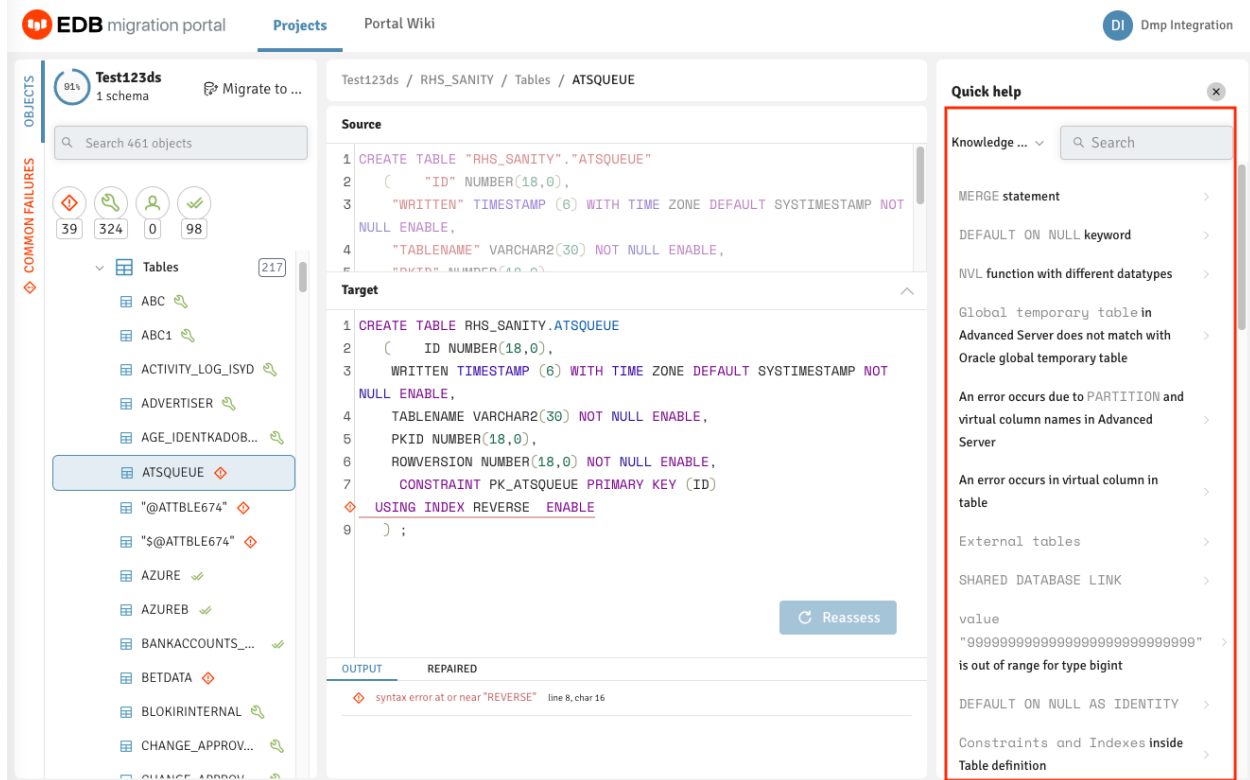


Fig. 3: Incompatible objects are identified.

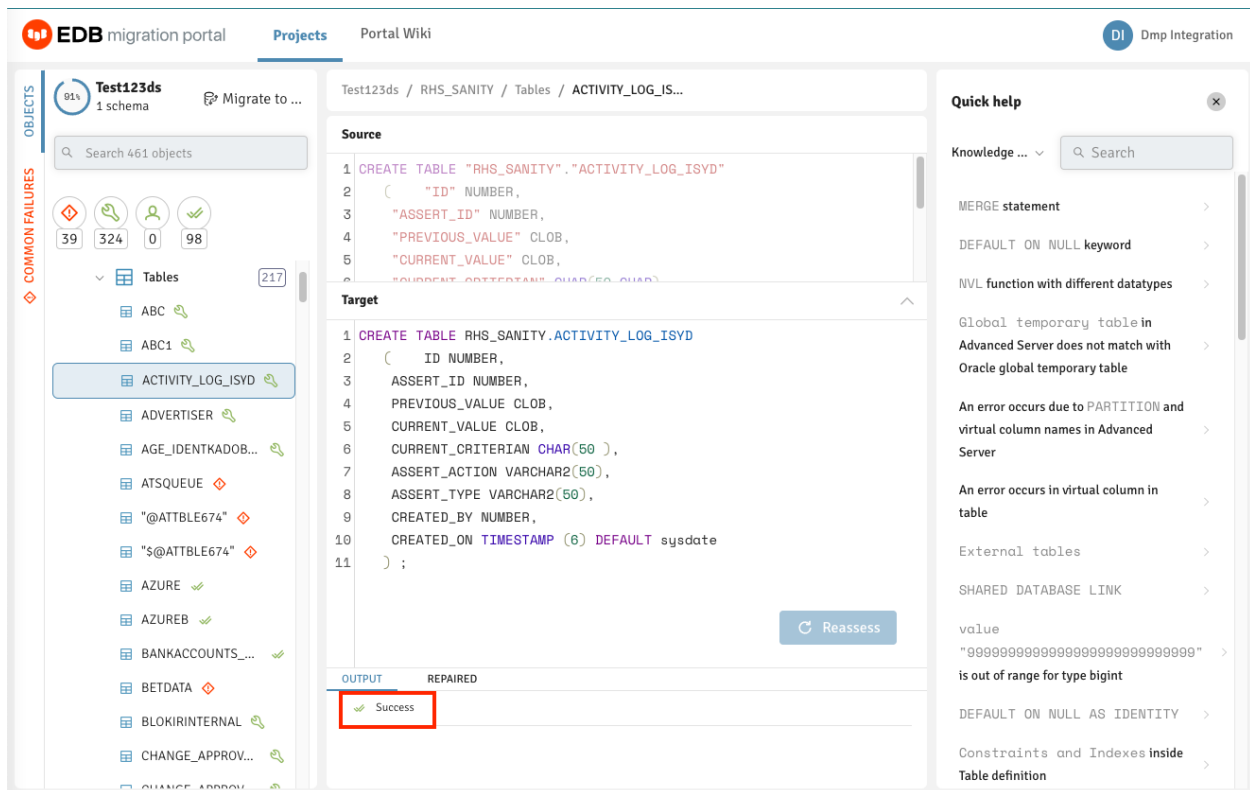


Fig. 4: Workaround or resolution for incompatible objects.

4.2.2 Generate an assessment report

Migration Portal reports provides a high-level assessment summary of the schemas assessed for your project. In addition, the report provides details about the failed objects and the cause of failure.

1. Select **Report** .
2. Select the schemas to include in the report.
3. To generate the onscreen report, select **Generate** .

You can optionally select **GeneratePDF** to save the report in PDF format. You can also view the count of distinct repair handlers applied to the DDLs under the selected schemas.

4.3 Schema migration

After resolving errors in your schemas, you can use the schemas with a client application such as pgAdmin, Toad Edge, or the PSQl client. Or you can migrate the schema to an EDB Postgres Advanced Server.

You can migrate to any of these schemas:

- Existing on-premises EDB Postgres Advanced Server
- New on-premises EDB Postgres Advanced Server
- EDB Postgres Advanced Server **in** the cloud

4.3.1 Migrating to an existing on-premises EDB Postgres Advanced Server host

You can migrate schemas to an existing on-premises EDB Postgres Advanced Server on Windows or Linux platforms.

Migrate schemas on Windows

1. Select **Existingon-premisesEDBPostgresPostgresAdvancedServer** .
2. Select one or more schemas to migrate to EDB Postgres Advanced Server.

Note: If your schemas aren't 100% compatible, a banner appears. Complete the Contact Us form as needed.

3. To download the assessed schemas, select **DownloadSQLfile** .
4. Select **Windows** .
5. To import the schemas, run:

- On CLI:

```
\i c:\users\...\<project_name>.sql
```

- On cmd/shell:

```
edb-psql -f <project_name>.sql
```

Note: You can alternatively use the *pgAdmin* client for the import.

The converted schemas migrate to the target server.

Migrate schemas on Linux

You can migrate schemas to an existing on-premises EDB Postgres Advanced Server on Linux.

1. Select **Existingon-premisesEDBPostgresAdvancedServer** .
2. Select one or more schemas to migrate to EDB Postgres Advanced Server.
3. To download the assessed schemas, select **DownloadSQLfile** .
4. Select **Linux** .
5. To import the schemas, invoke the following `edb-psql` client commands:

```
sudo su - enterprisedb
edb-psql edb
create database <database_name>;
\\connect <database_name>
\\i <project_name>.sql
```

Note: You can alternatively use the *pgAdmin* client for the import.

The converted schemas migrate to the target server.

4.3.2 Migrating to a new on-premises EDB Postgres Advanced Server installation

You can install new EDB Postgres Advanced Server on premises on Windows or Linux platforms and migrate the schemas.

Migrate schemas on Windows

Migrate schemas to a new on-premises EDB Postgres Advanced Server on Windows.

1. Select **Newon-premisesEDBPostgresPostgresAdvancedServer** .
2. Select one or more schemas to migrate on EDB Postgres Advanced Server.
3. For the operating system, select **Windows** .
4. Select **DownloadWindowsinstaller** .
5. For installation steps, select **EDBPostgresAdvancedServerInstallationGuideforWindows** .
6. To download the assessed schemas, select **DownloadSQLfile** .
7. You can import schemas by running:
 - On CLI

```
\i c:\users\...\<project_name>.sql
```

```
- On cmd/shell
```

```
edb-psql -f <project_name>.sql
```

Note: You can alternatively use the *pgAdmin* client for the import.

Migrate schemas on Linux

Migrate schemas to an on-premises EDB Postgres Advanced Server on Linux.

1. Select **NewOn-premisesEDBPostgresAdvancedServer** .
2. Select one or more schemas to migrate on EDB Postgres Advanced Server.
3. For the operating system, select **Linux** .
4. You can select one of the following options to install the EDB Postgres Advanced Server:
 - Repository - More options
5. For information on the installation procedure, select **EDBPostgresAdvancedServerInstallationGuide** for Linux.
6. To download the assessed schemas, select **DownloadSQLfile** .
7. To import the schemas, run:

```
sudo su - enterprisedb
edb-psql edb
create database <database_name>;
\connect <database_name>
\i <project_name>.sql
```

Note: You can alternatively use the *pgAdmin* client for the import.

The converted schemas migrate to the target server.

4.3.3 Migrate to the cloud

Migrate schemas on EDB Postgres Advanced Server to the cloud.

1. Select **EDBPostgresAdvancedServeronCloud** .
2. Select one or more schemas to migrate to EDB Postgres Advanced Server.
3. Select the cloud platform, for example, **BigAnimal** .
4. To launch a new cluster, select **GotoBigAnimal** . Or, if you have an existing cluster running, select **Next** .

Note: See the [Big Animal](<https://www.enterprisedb.com/edb-cloud>) page for more information.

5. Enter the required connection details on the Connect page.

Migrate your schemas to the cloud

Progress: Select Schemas — Select Platform — Launch Cluster — **4 Connect** — 5 Deploy — 6 Finish

Project Name	Interface	Source DB	Target DB	Compatibility
TestMigration	JDBC	Oracle 11g	EDB Postgres Advanced Server 13	99%

Connect to BigAnimal cluster

Target Database:

Host Name/Address: Port:

Note: Ensure that your database server is accessible from Migration Portal.

Maintenance Database:

Username:

Password:

[Test Connection](#)

[Cancel](#) [← Previous](#) [→ Next](#)

Fig. 5: Connecting to the cloud cluster

7. To verify the connection details, select **TestConnection**.

Note: You can select **Edit Connection** to make changes to the connection details and retest the connection details.

8. Once the connection is successful, select **Next**.

The converted schemas migrate to the target server.

4.4 Data migration

After performing the schema migration, migrate the data.

1. Use Migration Toolkit to migrate the data. For detailed information about using Migration Toolkit, see [Migration Toolkit](#).
2. Configure the Migration Toolkit `toolkit.properties` file, ensuring that connection information for the source and target databases is available in the property file:

```
SRC_DB_URL = jdbc: oracle:thin:@localhost:1521:ORCL
SRC_DB_USER = user_name
SRC_DB_PASSWORD = password
TARGET_DB_URL= jdbc:edb://localhost:5444/migration
TARGET_DB_USER = enterprisedb
TARGET_DB-PASSWORD = password
```

For more information, see [Building the toolkit.properties File](#).

3. Invoke Migration Toolkit in `-dataOnly` mode. Include the `-truncLoad` keyword to resolve foreign key dependencies across tables.

For example, the following command migrates data to a schema named `hr`:

```
runMTK.sh -dataOnly -targetSchema hr -truncLoad HR
```

The command migrates the specified `source_schema` to `target_schema`. The data is loaded into the locally installed EDB Postgres Advanced Server instance with a database superuser named `enterprisedb` and the password of `password`.

Note: The tables are truncated before attempting the data load.

ADVANCED DATA MIGRATION

For larger databases that require a parallel data load, you can use one of the following methods:

- The EDB Postgres Advanced Server database link feature (**for** compatibility **with**  Oracle databases)
- A Dblink or a database link style migration (if your data contains `CLOB` data)

For more information, see the [Migration Toolkit](#).