| 1  | What’s New | 2 |
| 2  | Supported Versions | 4 |
| 3  | Using the EDB Migration Portal | 5 |
| 3.1 | Overview of the Migration Portal Home Page | 7 |
| 3.2 | Overview of the Migration Portal Projects Page | 8 |
| 3.3 | Overview of the Migration Portal Wiki Page | 10 |
| 4  | Migrating a Database | 11 |
| 4.1 | Performing a Schema Extraction | 12 |
| 4.1.1 | Supported Object Types | 15 |
| 4.1.2 | Unsupported Object Types | 15 |
| 4.1.3 | Oracle System Schemas | 16 |
| 4.2 | Performing a Schema Assessment | 18 |
| 4.2.1 | Generating an Assessment Report | 23 |
| 4.3 | Schema Migration | 26 |
| 4.3.1 | Migrating a Schema to a CDS cluster | 27 |
| 4.4 | Data Migration | 29 |
| 5  | Advanced Data Migration | 30 |
| 6  | Conclusion | 31 |
| Index | 32 |
EDB Postgres Migration Portal (Migration Portal) is a web-based service for migrating Oracle database schemas to the EDB Postgres platform. The Migration Portal assesses and analyzes Oracle database schemas and converts types, tables, sequences, constraints, triggers, views, stored procedures, packages, dblink, materialized views, and indexes, producing DDLs that are compatible with EDB Postgres Advanced Server.

The user-friendly portal interface simplifies migration; log on to the portal and start the migration process.

The EDB Postgres Migration Portal guide provides a high-level description of the steps involved in the migration process. The guide also includes solutions to common migration problems and details of unsupported features and their potential workarounds.

EnterpriseDB has helped companies migrate their existing database systems to Postgres for years. For more information, visit the EnterpriseDB website at:

https://www.enterprisedb.com/
The following enhancements are added to the EDB Postgres Migration Portal for this release:

- **COMMENTS** on Columns, Tables, and Materialized Views are supported.
- All user schemas can be extracted from the current database.
- Project compatibility percentage and the total number of schemas for a project can be viewed with the project name. In addition, other UI changes are made to improve user experience.

**New Repair Handlers**

The following repair handlers are added to improve the Advance Server compatibility ratio:

- **ERH 2087** - Removes `RESULT_CACHE` from the DDL statement.
  
  For example:

  ```sql
  CREATE OR REPLACE FUNCTION get_complete_address RETURN VARCHAR2 RESULT_CACHE
  IS
      person_details VARCHAR2(130);
  BEGIN
      NULL;
  END get_complete_address;
  ```

  would become:

  ```sql
  CREATE OR REPLACE FUNCTION get_complete_address RETURN VARCHAR2
  IS
      person_details VARCHAR2(130);
  BEGIN
      NULL;
  END get_complete_address;
  ```

- **ERH 2088** - Replaces `VIRTUAL` keyword with `STORED` in TABLE definition.
  
  For example:
CREATE TABLE EMP1 (  
EID number,  
SAL number GENERATED ALWAYS AS (EID*(2+2)) VIRTUAL  
);  

would become;  

CREATE TABLE EMP (  
EID number,  
SAL number GENERATED ALWAYS AS (EID*(2+2)) STORED  
);  

• ERH 2089 - Removes empty spaces from operators i.e. <=, >=, ! = OR < >.  
For example:  

select 1 from dual where to_date('01/01/2020', 'dd/mm/yyyy') < = sysdate;  

would become;  

select 1 from dual where to_date('01/01/2020', 'dd/mm/yyyy') <= sysdate;  

Knowledge Base  
The following new knowledge base entries are added; refer to the Knowledge Base section on the Migration Portal for workaround details.  

• Foreign Key Constraints on Partitioned Tables:  
Oracle supports creating foreign key constraints on the partitioned tables. Advanced Server v11 onwards, creating foreign key constraints on the partitioned tables is supported; however, the earlier versions of Advanced Server do not support this functionality. So, we recommend assessing the schema on the latest Advanced Server version for better assessment results.  

• SYS_GUID() workaround:  
Oracle supports SYS_GUID() function, which generates and returns a globally unique identifier (RAW value). Advanced Server v12 onwards, SYS_GUID() function is supported; however, the earlier versions of Advanced Server do not support SYS_GUID() function. The workaround for the earlier versions is available on the Portal Wiki.  

• Count(*) with ORDER BY clause:  
Oracle supports the ORDER BY clause along with simple count(*) operation on a table. However, Advanced Server does not support the ORDER BY clause along with simple count(*). Advanced Server expects a column that appeared in the ORDER BY clause to be part of the GROUP BY clause as well.  

• Indexes on String literal:  
Advanced Server does not support index creation on the string literal, however; it allows index creation on columns.
The Migration Portal supports assessment and migration from Oracle 11g and 12c to EDB Advanced Server 10, 11, or 12. Migration Portal is supported on the following browsers and operating systems:

**Supported Browsers**

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

<table>
<thead>
<tr>
<th>Browser</th>
<th>Supported Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apple Safari on Macintosh OS</td>
<td>11 and above</td>
</tr>
<tr>
<td>Google Chrome</td>
<td>68 and above</td>
</tr>
<tr>
<td>Microsoft Edge</td>
<td>42 and above</td>
</tr>
<tr>
<td>Mozilla Firefox</td>
<td>60 and above</td>
</tr>
<tr>
<td>Internet Explorer</td>
<td>11 and above</td>
</tr>
</tbody>
</table>

**Supported Operating Systems**

<table>
<thead>
<tr>
<th>Operating Systems</th>
<th>Supported Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Macintosh</td>
<td>OS X Sierra</td>
</tr>
<tr>
<td>Windows</td>
<td>10</td>
</tr>
<tr>
<td>Linux</td>
<td>CentOS 7</td>
</tr>
</tbody>
</table>
CHAPTER 3

Using the EDB Migration Portal

The Migration Portal allows you to easily migrate your database schema from Oracle to 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. 3.1: The assessment and migration process
To access the migration portal:

1. Open a browser and navigate to https://www.enterprisedb.com/
2. On the EnterpriseDB home page, click Enterprise Postgres > EDB Postgres Migration Portal.

Fig. 3.2: Accessing the Migration Portal.
3.1 Overview of the Migration Portal Home Page

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

![Fig. 3.3: The Migration Portal home page.](image)

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

1. **Projects**: The Projects panel displays a list of assessed projects.
2. **Create project**: Click + New (the button located to the right of the Projects label) to create a new project.
3. **Overview**: The Overview panel provides details about the selected project and displays the compatibility percentage after schema assessment.
4. **Export**: Use the Export button to either download an Advanced Server compatible .sql file or to migrate a schema to an EDB Cloud Database Services cluster.
5. **Report**: Use the Report button to view and download the schema assessment report.
6. **Delete**: Use the Delete button to delete a selected project.
7. **Upload DDL file**: Use the Upload new schema button to upload a new or additional DDL file.
8. **Schemas**: The Schemas panel displays the assessment result from an uploaded DDL file.
9. **Quick help**: The Quick help panel contains all the help guides.
10. **Portal Wiki**: The Portal Wiki has links to product information and different help guides.
3.2 Overview of the Migration Portal Projects Page

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

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

1. **Compatible**: The Compatible gauge displays the compatibility percentage of the assessed schema.
2. **Schema Count**: Displays the number of schemas in a project.
3. **Export**: Use the Export icon to either download the .sql file or to migrate the schema to an EDB Cloud Database Service cluster.
4. **Search objects**: Use the Search box to search for objects.
5. **Filters**: You can filter the system repaired and manual repaired objects from the left panel of the Project page. In addition, you can select one or more filter combinations to refine the information.
6. **Schema**: The Schema panel displays the result of the assessment.
7. **Tooltip**: Hover over a result set to display a tooltip with the number of passed, failed, and repaired objects.
8. **Quick help**: The Quick help panel displays links to Knowledge base articles and repair handler documentation.
9. **Search**: Use the Search box to search the Knowledge base entries or repair handler documentation for specific information.

Fig. 3.4: The Migration Portal Projects Page overview.
Fig. 3.5: Searching the Knowledge Base entry.
3.3 Overview of the Migration Portal Wiki Page

The Portal Wiki page provides quick access to information:

- What’s New information
- Quick Start guide
- Migration Data guide
- DDL Extractor guide
- Knowledge Base
- Repair handlers
- Migration Portal User’s Guide
- FAQs
- Postgres Rocks Forum

Fig. 3.6: The Migration Portal Wiki page.
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*.

The following sections provide detailed information about each step in the migration process.
4.1 Performing a Schema Extraction

Prerequisites

Before extracting a schema, you must download the latest EDB DDL Extractor script from the Migration Portal Projects page or from the link provided in the DDL Extractor guide in the Portal Wiki. The script can be run in SQL Developer or SQL*Plus. It uses Oracle’s DBMS_METADATA built-in package to extract DDLs for different objects under schemas (specified while running the script). The EDB DDL extractor creates the DDL file that will be uploaded to the portal and analyzed for EDB Postgres compatibility.

Note: You must have SELECT CATALOG ROLE and SELECT ANY DICTIONARY privileges in the Oracle database.

For SQL*Plus

1. Connect to SQL*Plus and run the command:
   
   ```
   SQL>@edb_ddl_extractor.sql
   ```

2. Provide the schema name and the path directory in which the extractor will store the extracted DDL. When extracting multiple schemas, use a comma (',') as a delimiter.

Note: If you want to extract all the user schemas from the current database, do not mention any schema names while extracting. However, it is recommended to mention the schema names that you would like to extract.

3. If you want to extract dependent objects from other schemas, enter yes or no.
   
   For example, on Linux:
   
   Enter SCHEMA NAME[S] to extract DDLs:
   
   HR, SCOTT, FINANCE
   
   Enter the PATH to store DDL file:
   
   /home/oracle/extracted_ddls/
   
   Extract dependent objects from other schemas? (yes/no): yes
   
   On Windows:
   
   Enter SCHEMA NAME[S] to extract DDLs:
   
   HR, SCOTT, FINANCE
   
   Enter the PATH to store DDL file:
   
   C:\Users\Example\Desktop\
   
   Extract dependent objects from other schemas? (yes/no): yes

For SQL Developer

1. Connect to the SQL server and run the following command:
2. Enter a comma-separated list of schemas:

![Image of Enter Value window with hr, scott entered]

Fig. 4.2: Provide a list of schemas.

3. Enter file path for the output file:

![Image of Enter Value window with \Users\username\Desktop entered]

Fig. 4.3: Specify the output file path.

4. Extract dependent objects from other schemas?(yes/no): yes

![Image of Enter Value window with yes entered]

Fig. 4.4: Extracting dependent objects.
Note: You can also enter a single schema name in both the SQL*Plus and SQL Developer tools.

5. The script iterates through the object types in the source database and once the task is completed, the .SQL output is stored at the entered location, i.e., c:\Users\Example\Desktop\.

Additional Notes

- The EDB DDL Extractor does not extract objects that have names like:

  BIN$b54+4XIEYwPgUAB/AQBWwA= =$0

  To extract these objects, you must change the name of the objects and re-run the extraction process.

- DDL Extractor extracts nologging tables as normal tables. Once these tables are migrated to Advanced Server, WAL log files will be created.
4.1.1 Supported Object Types

The migration portal supports the migration of the following object types:

- Synonyms
- DB Links
- Types and Type Body
- Sequences
- Tables
- Constraints
- Indexes (Except LOB indexes and indexes on materialized views)
- Views
- Materialized Views
- Triggers
- Functions
- Procedures
- Packages

Note: COMMENTS on Columns, Tables, and Materialized Views are supported.

4.1.2 Unsupported Object Types

- Editions
- Operators
- Schedulers
- LOB indexes and Indexes on Materialized Views
- XML Schemas
- Profiles
- Role and Object Grants
- Tablespaces
- Directories
- Users
- RLS Policy
- Queues
4.1.3 Oracle System Schemas

EDB DDL Extractor script will ignore the following system schemas while extracting from Oracle:

- ANONYMOUS
- APEX_PUBLIC_USER
- APEX_030200
- APEX_040000
- APEX_040200
- APPQOSSYS
- AUDSYS
- BI
- CTXSYS
- DMSYS
- DBSNMP
- DIP
- DVF
- DVSYS
- EXFSYS
- FLOWS_FILES
- FLOWS_020100
- GSMADMIN_INTERNAL
- GSMCATUSER
- GSMUSER
- IX
- LBACSYS
- MDDATA
- MDSYS
- MGMT_VIEW
- OE
- OJVMSYS
- OLAPSYS
- ORDPLUGINS
- ORDSYS
- ORDDATA
- OUTLN
- ORACLE_OCM
- OWBSYS
OWBYSS_AUDIT
PM
RMAN
SH
SI_INFORMTN_SCHEMA
SPATIAL_CSWS_ADMIN_USR
SPATIAL_WFS_ADMIN_USR
SYS
SYSBACKUP
SYSDG
SYSKM
SYSTEM
SYSDMAN
TSMSYS
WKPROXY
WKSYS
WK_TEST
WMSYS
XDB
XSSNULL
4.2 Performing a Schema Assessment

To assess an Oracle database schema for compatibility with Advanced Server, you must:

1. Navigate to the Migration Portal.
2. Enter your EDB credentials.
3. Click the Create project icon to create a new project.

4. On the New project dialog, enter the project name in the Project name field.

5. Specify project details:
   - Select the radio button next to the Application interface.
   - Select a Source DB and Version for the source.
   - Select the Target DB and Version for the target.

6. Click Choose file to upload the .SQL file generated by the latest EDB DDL Extractor for Oracle Database.

Note:
• You should not modify the .SQL file.
• Only the .SQL file generated by the latest EDB DDL Extractor can be uploaded.

For more information, refer to the Schema Extraction section.

7. Check the box next to Add Index Prefix to specify an index prefix (idx) when creating a project to ensure better assessment results, as Advanced Server does not support the same name for tables and indexes.

8. Click Create & assess to create a new project and to assess the schemas.

![Fig. 4.6: The Schema analysis result.](image)

The analysis tool will review every construct and execute repair actions to improve compatibility with Advanced Server, and flag any remaining errors that require manual intervention.

9. Verify the DDL objects (e.g., TABLES) that do not show a 100% success ratio.
10. Click the objects that are not compatible with EDB Postgres and view the details. At the bottom of the middle panel, you can view repair action details.

11. Refer to the Knowledge Base information in the right panel to locate the possible workarounds for the objects.
that are not immediately compatible with Advanced Server. You can also view the Knowledge Base information on the Portal Wiki page.

![Assessment result with errors.](image)

**Fig. 4.9: Assessment result with errors.**

12. On the Knowledge Base tab, you can enter the error message for the incompatible objects with Advanced Server and click 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 click Reassess.

**Note:** If any failed object passes while reassessing, the dependent objects for that object are also reassessed.
Fig. 4.10: Workaround or resolution for incompatible objects.

Similarly, you can make all the incompatible objects compatible.

**Note:** If the information or workaround for incompatible objects are not available in the Knowledge Base, please contact the support team for assistance.

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 to a CDS cluster.
4.2.1 Generating an Assessment Report

Migration Portal’s report functionality 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.

![Image of report selection dialog]

Fig. 4.11: Select schemas for reports.

To generate a report:

1. Click the Report button to access the schema selection dialog.
2. Select the schemas that you wish to include in the report.
3. Click Generate to generate the onscreen report.
4.2. Performing a Schema Assessment

You can optionally select Generate PDF 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.
Fig. 4.13: The saved pdf report.
4.3 Schema Migration

After resolving errors in your schemas, you can use the schemas with a client application such as pgAdmin, ToadEdge, or the PSQL client, or migrate the schema to an EDB Cloud Database Service (CDS) cluster.

**Please Note:** For more information about using Toad Edge with Advanced Server, see Toad Edge® for Postgres.

**Using PSQL or pgAdmin to Deploy a schema**

Ensure that the assessed schema is 100% compatible on Migration Portal.

Perform the following steps to deploy a schema using PSQL or pgAdmin:

1. On the **Project** Page, select the required project.
2. Click **Export** to download the assessed file.

![Fig. 4.14: Selecting download file option](image_url)

3. Click **Download**.

![Fig. 4.15: Downloading assessed file](image_url)

4. (For PSQL) Connect to desired Advanced Server using the psql/edb-psql client and run the following command:
   
   ```
   edb=# \i /exported_ddls_folder/hr_schema/ProjHR_hr.sql
   ```

5. (For pgAdmin) Connect to the desired database and click **Open**.

6. Upload the schema and click **Run**.

The Advanced Server instance must be installed in Oracle Mode to enable native compatibility with key Oracle capabilities.
4.3.1 Migrating a Schema to a CDS cluster

Perform the following steps to migrate your database to a CDS cluster:

1. On the Project page, click Export.
2. Select the Deploy to existing CDS cluster. Click here to launch new cluster option.

   For information about creating a new cluster, see Creating a Server Cluster.

   ![Fig. 4.16: Migrating database to CDS cluster.](image)

3. Click Next.

4. Select the schemas you wish to migrate.

   ![Fig. 4.17: Select schemas for migration.](image)

5. Click Next to continue.
6. Enter the required details in the **Connection Details** dialog box.

![Connection Details](image)

**Fig. 4.18: Provide connection details.**

7. Click **Test Connection** to verify the connection details.

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

**Fig. 4.19: Testing a successful connection.**

---

4.3. Schema Migration
8. Once the connection is successful, click Deploy.

9. You can view the deployment details on the Deploy dialog; click Download Summary to download the deployment log.

![Fig. 4.20: A successful deployment.](image)

10. Click Done to close the window.

### 4.4 Data Migration

After performing the schema migration, complete the following steps to migrate data:

1. Use EDB Migration Toolkit to migrate the data. For detailed information about using Migration Toolkit, see the EDB Postgres Migration Guide.

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:

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

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

   **Note:** The tables are truncated before attempting the data load.
For larger databases that require a parallel data load, you can use one of the following methods:

- Use the EDB Postgres Advanced Server database link feature (for compatibility with Oracle databases).
  
  Or
  
  - Use a Dblink or a database link style migration (if your data contains CLOB data).

For more information, see the EDB Postgres Migration Guide.
Conclusion
Index

A
Advanced Data Migration, 30

C
Conclusion, 31

D
Data Migration, 29

K
knowledge base, 8, 10

M
Migrating a Database, 11
Migration Toolkit, 29

O
Overview of the Migration Portal Home Page, 7
Overview of the Migration Portal Projects Page, 8
Overview of the Migration Portal Wiki Page, 10

P
Performing a Schema Assessment, 18
Performing a Schema Extraction, 12
pgAdmin, 26
psql, 26

R
repair handlers, 10
resources, 10

S
schema assessment report, 22, 23
schema connection details, 28
Schema Migration, 26
supported browsers, 4
supported databases, 4
supported object types, 14, 15
supported operating systems, 4
Supported Versions, 4

test schema connection, 28
Toad Edge, 26

U
Using the EDB Migration Portal, 5

W
What’s New, 2