



EDB Postgres Migration Portal

Release 2.2.0

User's Guide

Sep 09, 2019

Contents

1	What's New	2
2	Supported Versions	5
3	Using the EDB Migration Portal	6
3.1	Overview of the Migration Portal Home Page	8
3.2	Overview of the Migration Portal Projects Page	9
3.3	Overview of the Migration Portal Wiki Page	11
4	Migrating a Database	12
4.1	Performing a Schema Extraction	13
4.1.1	Supported Object Types	15
4.2	Performing a Schema Assessment	16
4.2.1	Generating an Assessment Report	21
4.3	Schema Migration	23
4.3.1	Migrating a Schema to a CDS cluster	24
4.4	Data Migration	27
5	Advanced Data Migration	28
6	Conclusion	29
	Index	30

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, dblinks, 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/>

What's New

The following features are added to the EDB Postgres Migration Portal for this release:

- You can now choose to extract dependent objects while extracting the schemas. The dependent objects from the other schemas will also be extracted, which will help in improving the assessment results.
- New users will have an improved experience when getting started with the Migration Portal.
- As Advanced Server does not support the same name for tables and indexes, you may now choose to add an index prefix, 'idx', while creating a project for better assessment results.
- The assessment report shows the count of distinct repair handlers applied to the DDLs under the selected schema.

Repair Handlers Applied

The following table displays the number of times the unique repair handlers applied on schemas.

Code	Short description	Long description	Usage Count
ERH-2013	EDITIONABLE	deleted [EDITIONABLE clause]	700
ERH-2009	ENABLE	deleted [ENABLE]	618
ERH-2005	USING_INDEX_ENABLE	deleted [USING INDEX <CREATE INDEX CLAUSE> ENABLE]	320
ERH-2007	NOORDER_SEQUENCE	deleted [Un-supported sequence option (ORDER or NOORDER)]	153
ERH-2024	NOPARTITION_SEQUENCE	deleted [Un-supported sequence option N OPARTITION)]	153
ERH-2023	NOCYCLE_SEQUENCE	replaced [Un-supported sequence option NOCYCLE] with [NO CYCLE]	153
ERH-2019	SEQUENCE_OUT_OF_RANGE_ERROR	replaced [out of range value] with [+/-922 3372036854775807]	153

Fig. 1.1: *Applied repair handlers count*

- You can now filter system-repaired and manually-repaired objects on the Workspace page.

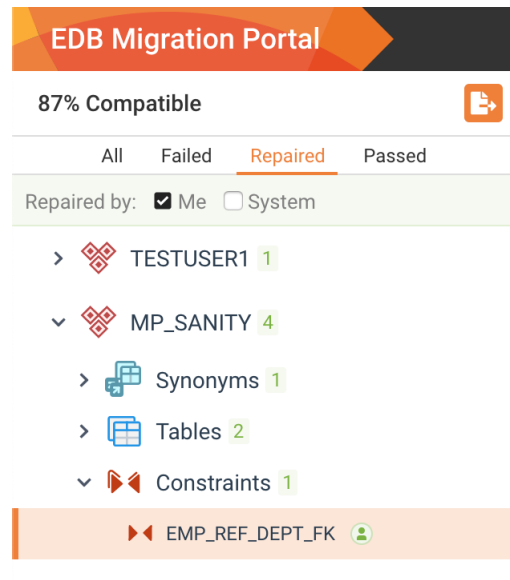


Fig. 1.2: Manually repaired objects

Note:

Starting this release, Migration Portal will accept DDLs extracted by the latest EDB DDL Extractor only.

New Repair Handlers

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

- **ERH 2073** - Removes `CONSTRAINT constraint_name NOT NULL DISABLE` clause from the source DDL for tables.

For example, `CREATE TABLE TEST (COL1 VARCHAR2(40) CONSTRAINT TEST_COL1_NN NOT NULL DISABLE);` would become `CREATE TABLE TEST (COL1 VARCHAR2(40));`

- **ERH 2074** - Removes `NOT NULL DISABLE` clause from the source DDL for tables.

For example, `CREATE TABLE TEST(COL1 VARCHAR2(40) NOT NULL DISABLE );` would become `CREATE TABLE TEST (COL1 VARCHAR2(40));`

- **ERH 2075** - Removes `CONSTRAINT constraint_name CHECK|UNIQUE|PRIMARY KEY parameters DISABLE` clause from the source DDL for tables.

For example, `CREATE TABLE TEST(DATE_COL1 DATE, CONSTRAINT TEST_DATE_COL1_CHK CHECK (DATE_COL1 > '01-JAN-2018') DISABLE);` would become `CREATE TABLE TEST (DATE_COL1 DATE);`

- **ERH 2076** - Removes `CHECK|UNIQUE|PRIMARY KEY parameters DISABLE` clause from the source DDL for tables.

For example, `CREATE TABLE TEST (ID VARCHAR2(40), PRIMARY KEY (ID) DISABLE);` would become `CREATE TABLE TEST (ID VARCHAR2(40));`

Knowledge Base

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

- **NVL function inside Index definition**

In Advanced Server, the `DATE` datatype from Oracle is converted into a `timestamp without time zone` datatype for compatibility. However, in Advanced Server v10, the `nvl()` function does not support the `timestamp without time zone` datatype in arguments. The following data types are supported as input arguments for the `nvl()` function in Advanced Server v10:

`character varying`

`Numeric`

`Date`

`timestamp with time zone`

Any other datatype used as an input value for the `nvl()` function inside the index definition will return an error.

Supported Versions

The Migration Portal supports assessment and migration from Oracle 11g and 12c to EDB Advanced Server 10 or 11. 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:

Browser	Supported Version
Apple Safari on Macintosh OS	11 and above
Google Chrome	68 and above
Microsoft Edge	42 and above
Mozilla Firefox	60 and above
Internet Explorer	11 and above

Supported Operating Systems

Operating Systems	Supported Version
Macintosh	OS X Sierra
Windows	10
Linux	CentOS 7

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.

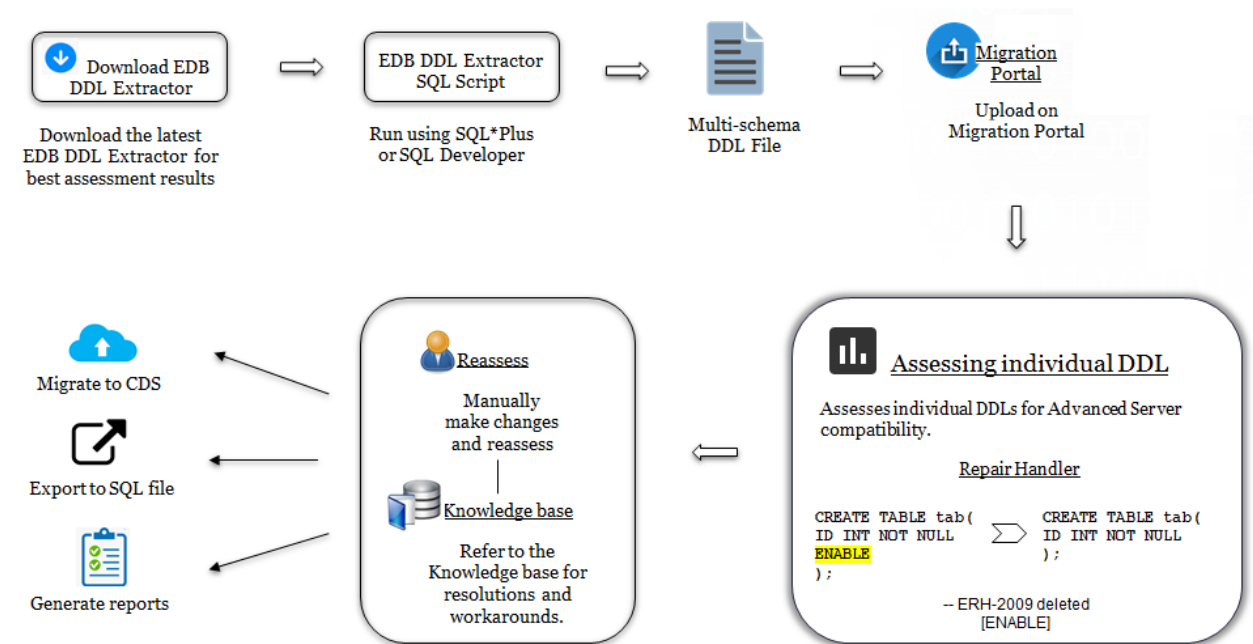


Fig. 3.1: The assessment and migration process

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.

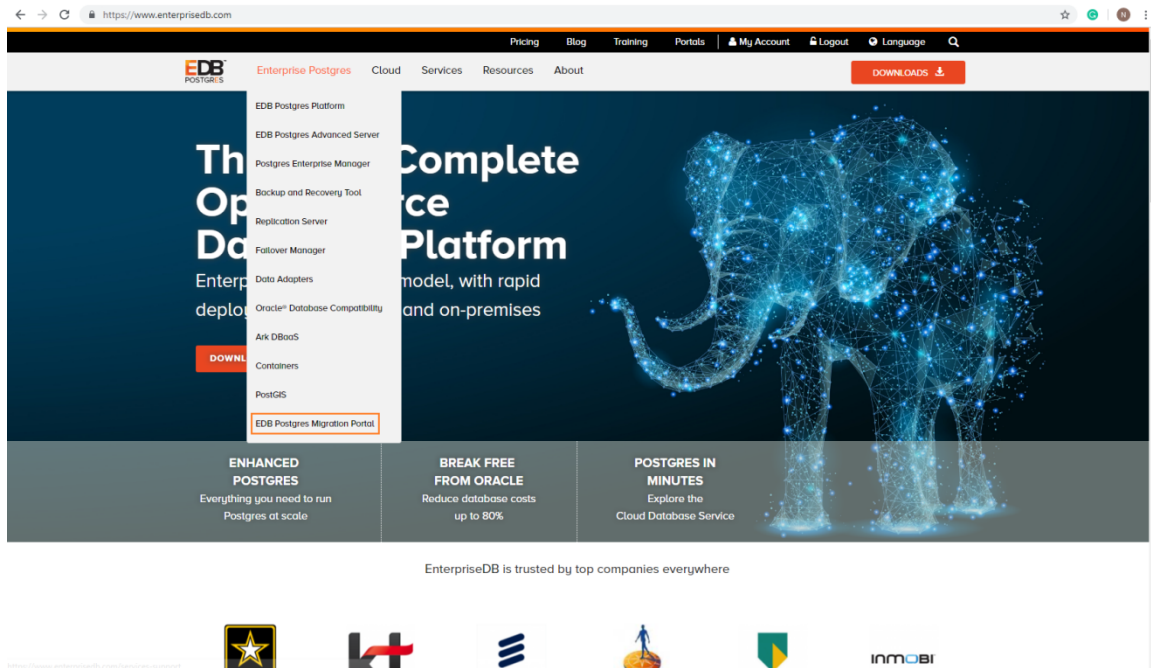


Fig. 3.2: Accessing the Migration Portal.

3.1 Overview of the Migration Portal Home Page

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

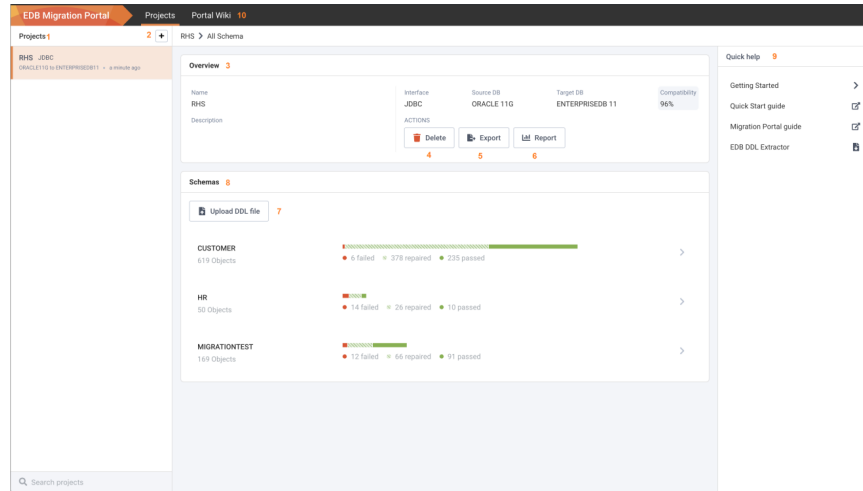


Fig. 3.3: The Migration Portal home page.

The page allows access to the following Migration Portal features:

1. **Projects:** The `Projects` panel displays a list of assessed projects.
2. **Create project:** Use the `Create project` icon (the plus icon 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. **Delete:** Use the `Delete` button to delete a selected project.
5. **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.
6. **Report:** Use the `Report` button to view and download the schema assessment report.
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` displays product information and help guides.

3.2 Overview of the Migration Portal Projects Page

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

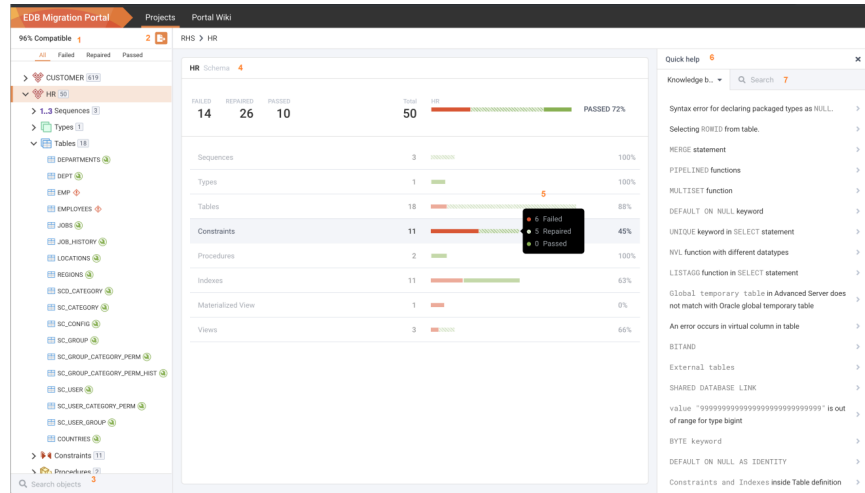


Fig. 3.4: The Migration Portal Projects Page overview.

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

1. **Compatible:** The Compatible panel displays a list of schemas and provides information (passed, failed, or repaired) about each object that has been assessed.
Note: You can also filter the system repaired and manual repaired objects from the left panel of the Project page.
2. **Export:** Use the Export icon (located to the right of the Compatible label) to either download the .sql file or to migrate the schema to an *EDB Cloud Database Service* cluster.
3. **Search:** Use the Search icon to search for a project.
4. **Schema:** The Schema panel displays the result of the assessment.
5. **Tooltip:** Hover over a result set to display a tooltip with the number of passed, failed, and repaired objects.
6. **Quick help:** The Quick help panel displays links to Knowledge base articles that can be used to fix failed objects.
7. **Search:** Use the Search box to search the Knowledge base entries for information about repairing failed objects.

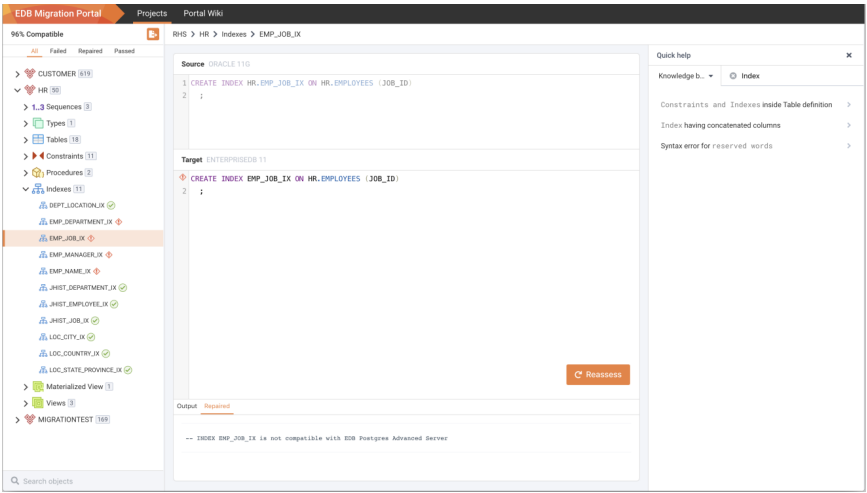


Fig. 3.5: Searching the Knowledge Base entry.

3.3 Overview of the Migration Portal Wiki Page

The Portal Wiki page provides quick access to information:

- the Migration Portal User's Guide
- the Migration Portal Quick Start Guide
- What's New information
- the Knowledge Base
- Repair handlers
- FAQs
- the Postgres Rocks Forum

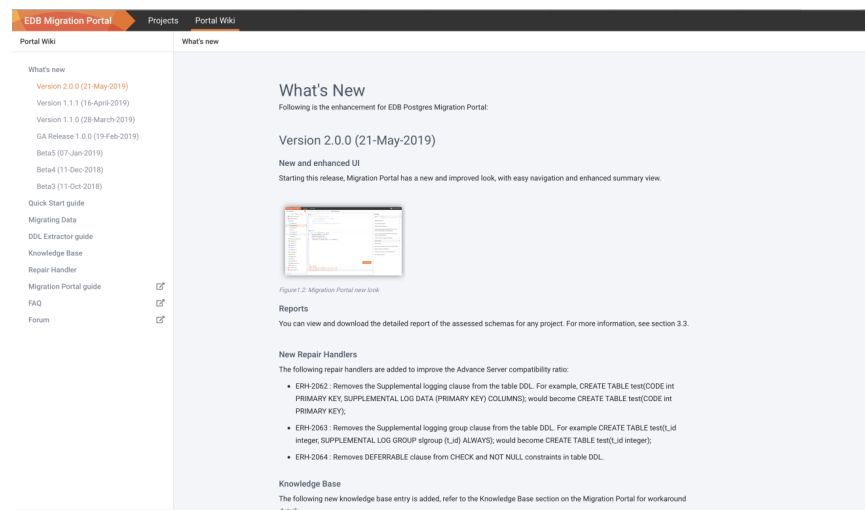


Fig. 3.6: The Migration Portal Wiki page.

Migrating a Database

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

1. Perform a *Schema Extraction*.
2. Review the *Schema Assessment*.
3. Perform a *Schema Migration*.
4. Populate the database with 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.
3. If you want to extract dependant 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:

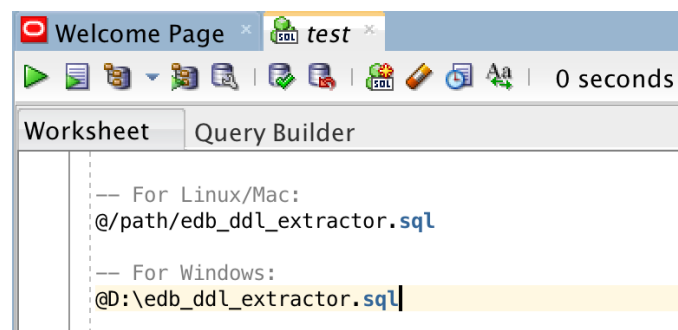


Fig. 4.1: Enter the path for Linux or Windows.

2. Enter a comma separated list of schemas:

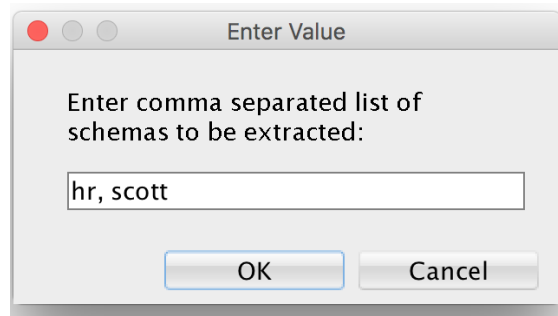


Fig. 4.2: Provide a list of schemas.

3. Enter file path for the output file:

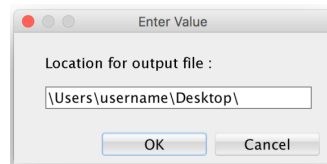


Fig. 4.3: Specify the output file path.

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

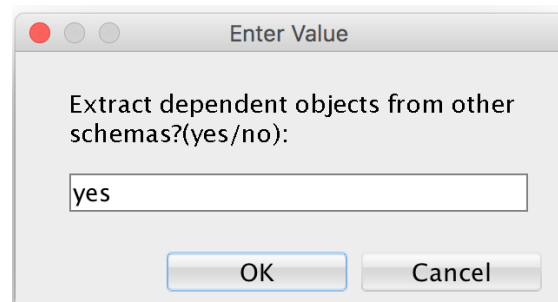


Fig. 4.4: Extracting dependent objects.

Note: You can also enter a single schema name in both 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 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

4.2 Performing a Schema Assessment

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

1. Use SQL*Plus or SQL Developer to connect to your Oracle database. You must connect with a database role that has `SELECT_CATALOG_ROLE` or `SELECT ANY DICTIONARY` privileges.
2. Invoke the DDL Extractor with the command:

```
SQL>@edb_ddl_extractor.sql.
```

The EDB DDL Extractor uses the SQL*Plus `ACCEPT` command to retrieve a list containing one or more schema names; the list should use a comma (,) delimiter. You can enter a maximum of 240 bytes. The maximum length of a schema name is 30 characters and you can provide 7 to 8 schema names.

3. Enter the path where the extraction file will be created.

On Linux:

Enter the schema name to extract the DDLs:

```
schema_name
```

Enter the path to store DDL file:

```
/home/oracle/extracted_ddls/
```

On Windows:

Enter schema name to extract the DDLs:

```
HR
```

Enter the path to store DDL file:

```
C:/Extracted_DDL/
```

4. Navigate to the [Migration Portal](#).
5. Enter your EDB credentials.
6. Click the `CREATE PROJECT` icon to create a new project.

Fig. 4.5: The Migration Portal New project dialog.

7. On the New project dialog, enter the project name in the Project name field.
 8. 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.
 9. 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.
10. 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.
 11. Click Create & assess to create new project and to assess the schemas.

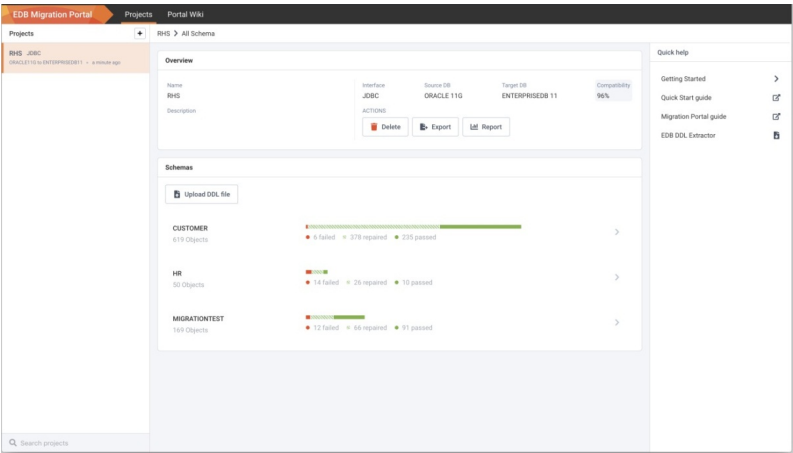


Fig. 4.6: The Schema analysis result.

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.

11. Verify the DDL objects (e.g., TABLES) that do not show a 100% success ratio.

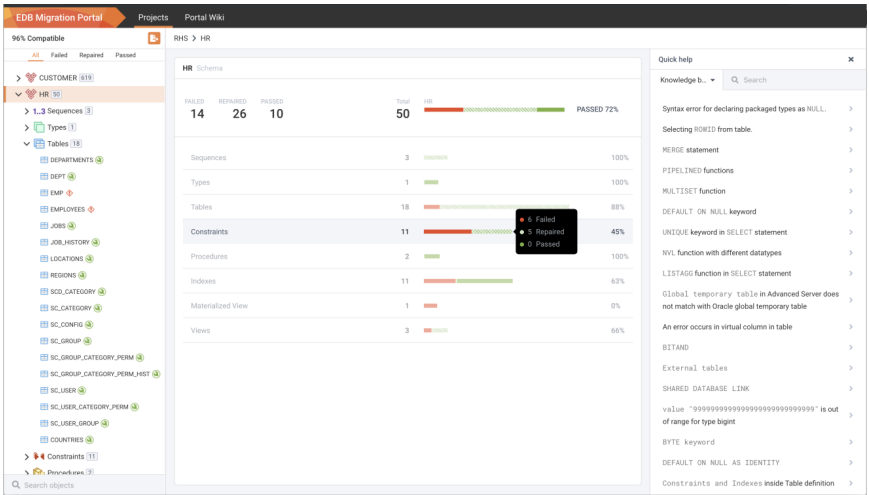


Fig. 4.7: Verifying the DDL objects.

12. 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.

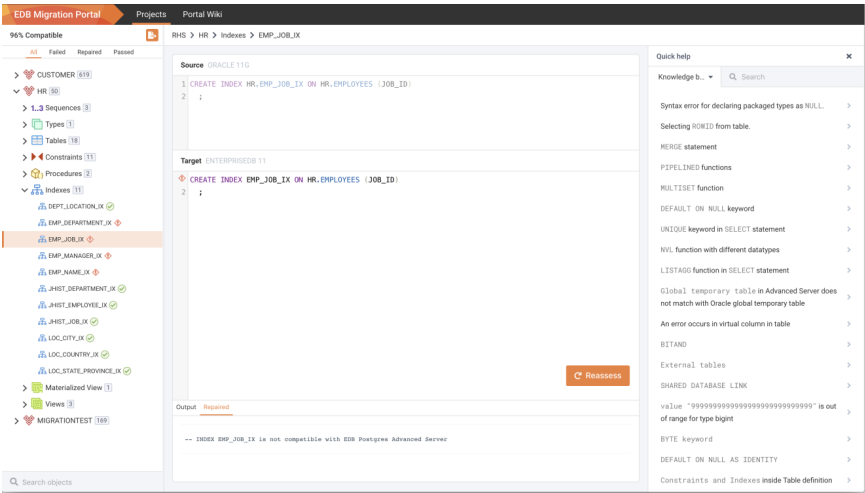


Fig. 4.8: Incompatible objects are identified.

13. 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.

Note: You can also view the Knowledge Base information on the Portal Wiki page.

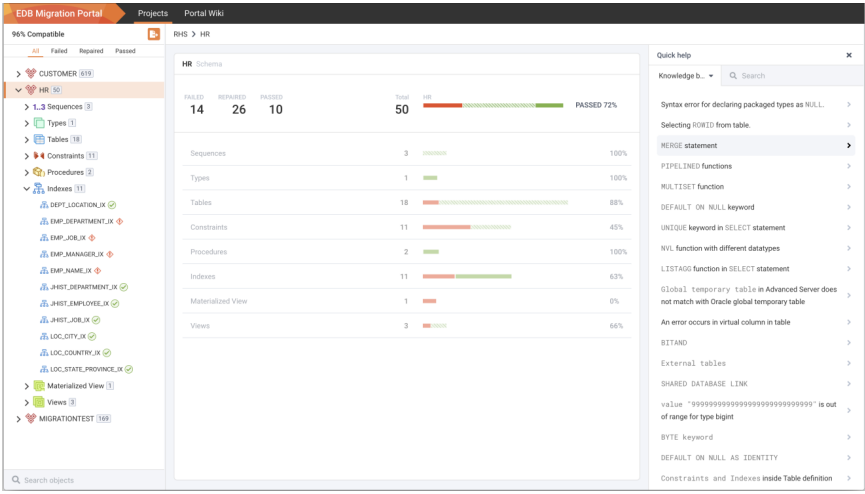


Fig. 4.9: Assessment result with errors.

14. 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.

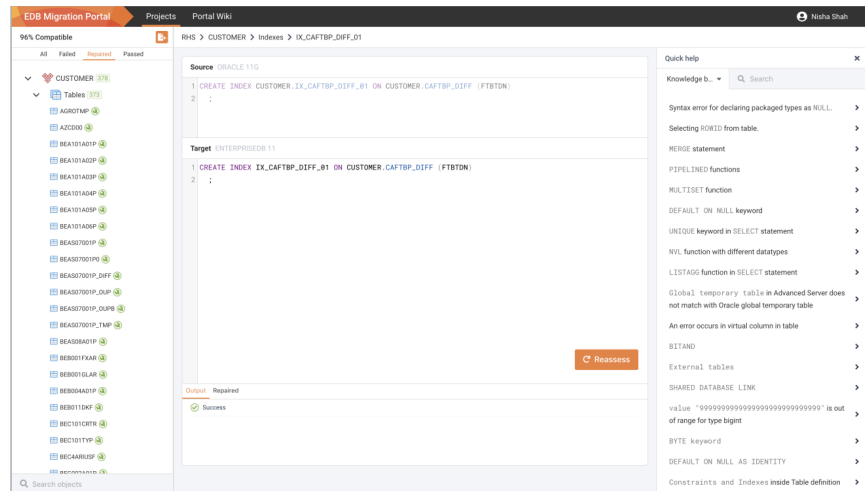


Fig. 4.10: Workaround or resolution for incompatible object.

Similarly, you can make all the incompatible objects compatible.

Note: If the object is 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.

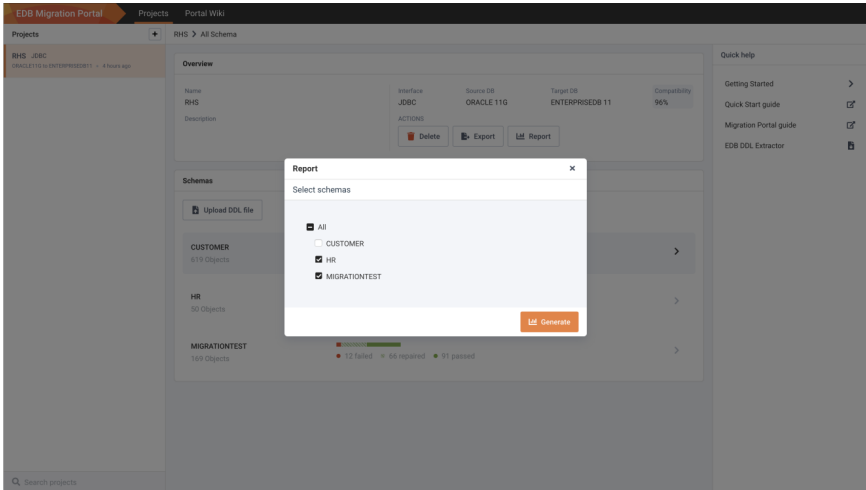


Fig. 4.11: Select schemas for reports.

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.

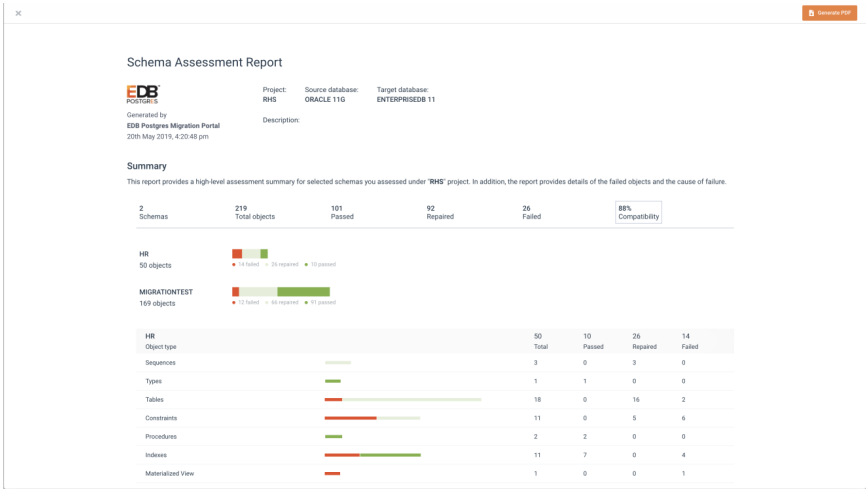


Fig. 4.12: The Schema Assessment Report.

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.

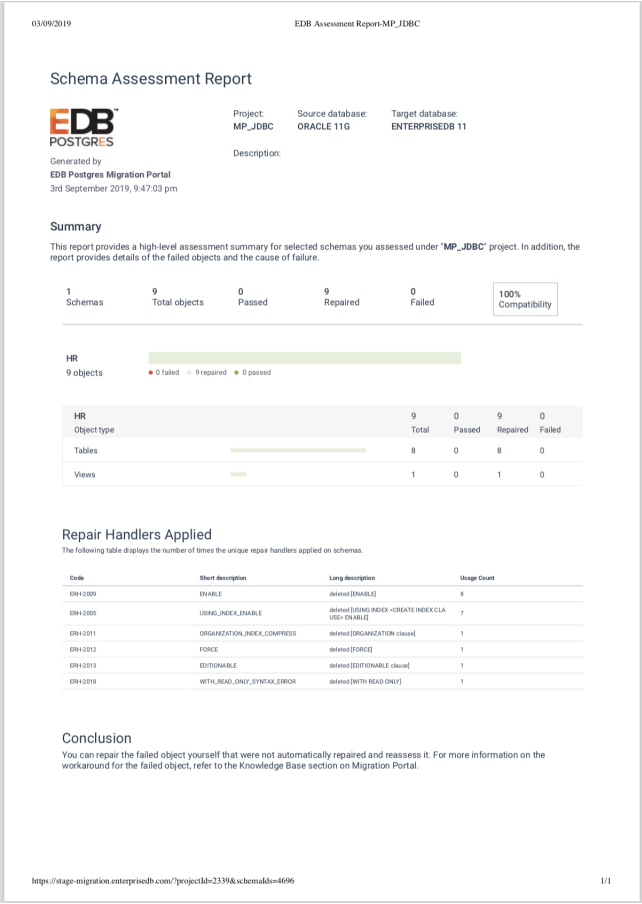


Fig. 4.13: The saved pdf report.

4.3 Schema Migration

After resolving errors in your schemas, you can use the schemas with a client application such as pgAdmin, ToadEdge, or 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](#).

Deploying schema using PSQL or pgAdmin

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

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

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

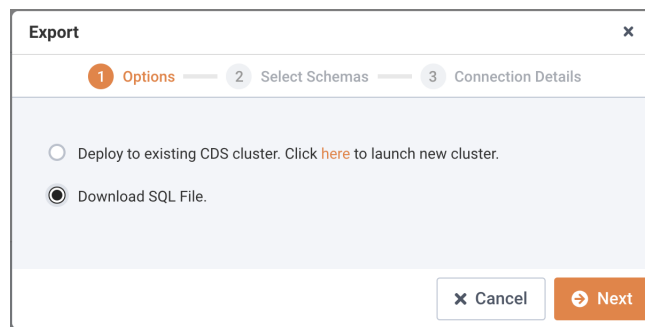


Fig. 4.14: Selecting download file option

3. Click **Download**.

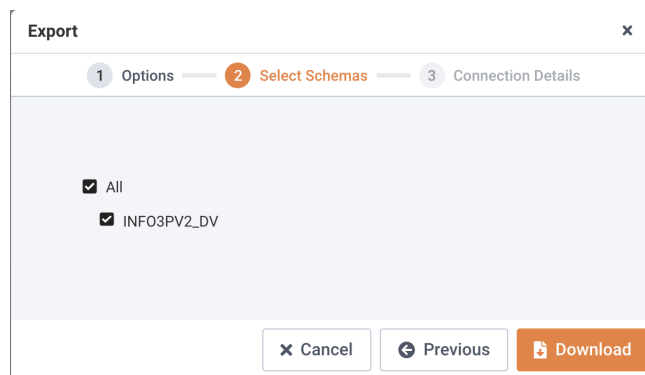


Fig. 4.15: Downloading assessed file

4. (For PSQL) Connect to desired Advanced Server using psql/edb-psql client and run the following command:

```
edb=# \i /exported_ddls_folder/hr_schema/ProjHR_hr.sql
```

5. (For pgAdmin) Connect to 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 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](#).

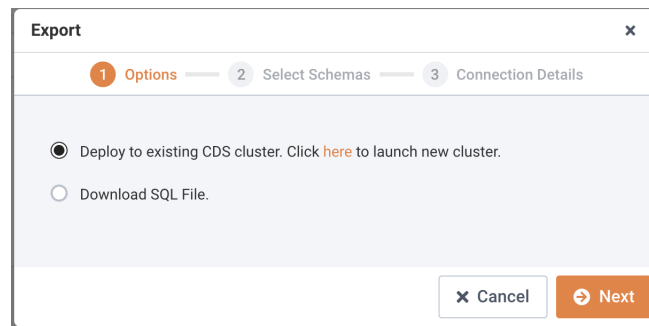


Fig. 4.16: *Migrating database to CDS cluster.*

3. Click `Next`.
4. Select the schemas you wish to migrate.

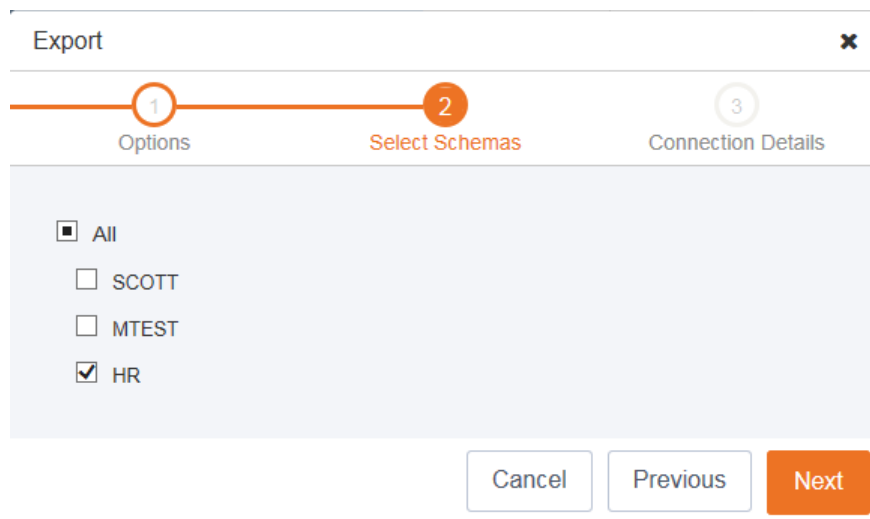


Fig. 4.17: *Select schemas for migration.*

5. Click `Next` to continue.

The screenshot shows a web-based 'Export' dialog box. At the top, there's a title bar 'Export' with a close button 'x'. Below it are three tabs: '1 Options', '2 Select Schemas', and '3 Connection Details' (which is highlighted in orange). The 'Connection Details' tab contains several input fields: 'Target database' with the value 'RHS', 'Host name/address' which is empty, 'Port' with the value '9999', 'Maintenance database' with the value 'edb', 'Username' which is empty, and 'Password' which is masked with dots. At the bottom of the dialog are three buttons: 'Cancel' (with a close icon), 'Previous' (with a left arrow icon), and 'Test Connection' (with a test icon and highlighted in orange).

Fig. 4.18: *Provide connection details.*

6. Enter the following details in the `Connection Details` dialog:
- Enter the host name or IP address in the `Host name/address` field.
 - Enter the port number in the `Port` field.
 - Enter the database name in the `Maintenance database` field.
 - Enter the user name in the `Username` field.
 - Enter the password associated with the user in the `Password` field.

Export

1 Options 2 Select Schemas 3 **Connection Details**

Target database RHS Edit

Host name/address 52.5.148.59

Port 5444

Maintenance database

Username

Password

Connection successful.

Note:

- Deployment will create a new database RHS on target server, if the database does not exist.
- Ensure that all schemas have a 100% success ratio before deployment.
- Existing schemas with similar names will be dropped during deployment.

Cancel Previous Deploy

Fig. 4.19: Testing a successful connection.

- 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.

- Once the connection is successful, click **Deploy**.
- You can view the deployment details on the **Deploy** dialog; click **Download Summary** to download the deployment log.

Deploy

Successfully deployed.

HOST: 123.45.7
PORT: 5444
DATABASE: Test2_EPAS11

Download Summary Done

Fig. 4.20: A successful deployment.

- 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`.

Please 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:

- 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](#).

CHAPTER 6

Conclusion

EDB Postgres Migration Portal User's Guide

Copyright © 2007 - 2019 EnterpriseDB Corporation.

All rights reserved.

EnterpriseDB® Corporation

34 Crosby Drive, Suite 201, Bedford, MA 01730, USA

T +1 781 357 3390 F +1 978 467 1307 E

info@enterprisedb.com

www.enterprisedb.com

- EDB designs, establishes coding best practices, reviews, and verifies input validation for the logon UI for EDB Postgres products where present. EDB follows the same approach for additional input components, however the nature of the product may require that it accepts freeform SQL, WMI or other strings to be entered and submitted by trusted users for which limited validation is possible. In such cases it is not possible to prevent users from entering incorrect or otherwise dangerous inputs.
- EDB reserves the right to add features to products that accept freeform SQL, WMI or other potentially dangerous inputs from authenticated, trusted users in the future, but will ensure all such features are designed and tested to ensure they provide the minimum possible risk, and where possible, require superuser or equivalent privileges.
- EDB does not warrant that we can or will anticipate all potential threats and therefore our process cannot fully guarantee that all potential vulnerabilities have been addressed or considered.

A

Advanced Data Migration, [28](#)

C

Conclusion, [29](#)

D

Data Migration, [27](#)

DDL extractor, [16](#)

K

knowledge base, [11](#)

M

Migrating a Database, [12](#)

Migration Toolkit, [27](#)

O

Overview of the Migration Portal Home Page, [8](#)

Overview of the Migration Portal Projects Page, [9](#)

Overview of the Migration Portal Wiki Page, [11](#)

P

Performing a Schema Assessment, [16](#)

Performing a Schema Extraction, [13](#)

pgAdmin, [23](#)

psql, [23](#)

R

repair handlers, [11](#)

resources, [11](#)

S

schema assessment report, [20](#)

schema connection details, [25](#)

Schema Migration, [23](#)

supported browsers, [5](#)

supported databases, [5](#)

supported object types, [14](#)

supported operating systems, [5](#)

Supported Versions, [5](#)

T

test schema connection, [26](#)

Toad Edge, [23](#)

U

Using the EDB Migration Portal, [6](#)

W

What's New, [2](#)