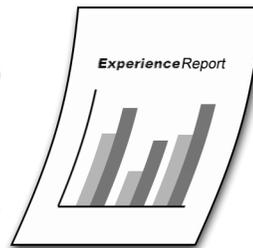


iSeries



# Oracle Migration Toolkit

# Experience Report





iSeries



# Oracle Migration Toolkit



---

## Contents

<b>Chapter 1. Oracle Migration . . . . .</b>	<b>1</b>	Generate data transfer scripts . . . . .	8
Manage migration projects . . . . .	1	Deploy to DB2 UDB for iSeries . . . . .	9
Specify the source . . . . .	2		
Convert source metadata . . . . .	3		
Refine the metadata conversion . . . . .	7	<b>Chapter 2. Disclaimer . . . . .</b>	<b>11</b>



---

## Chapter 1. Oracle Migration

This document describes the process of migrating an Oracle<sup>(TM)</sup> database to DB2<sup>(R)</sup> Universal Database<sup>(TM)</sup> for iSeries<sup>(TM)</sup> using the Oracle Migration Toolkit (MTK). The process has been broken down into 6 subtasks, and each is explained in detail. Please note that for the purposes of generating this report, we did not use the "migration wizard" portion of the tool. We went through the process of doing all of the subtasks that the tool provides, so that each could be described in this report.

*Manage migration projects*

*Specify the source*

*Convert source metadata*

*Refine the metadata conversion*

*Generate data transfer scripts*

*Deploy to DB2 for iSeries*

### **Finding more information**

The MTK tool provides extensive online help that covers each topic in more detail than this experience report. However, since the tool is built to run against different DB2 platforms, certain details in the generic help either don't apply to the iSeries deployment or are missing from the help. That is why this experience report was created. It should be considered a complement to the help text, since it is specifically aimed at the iSeries migration process.

You can find comprehensive information about DB2 UDB for iSeries in the iSeries Information Center (<http://www.ibm.com/eserver/iseres/infocenter>).

---

## **Manage migration projects**

The Oracle Migration Toolkit (MTK) lets you organize your work into projects. Each migration project is kept separate, so you can easily manage all the files that are generated during each migration. Once you have created a project, you can come back at any time and open a previously created project. When you start the MTK, you can choose to create a new project or open a project that you have created previously.

To create a new project in the MTK, perform the following steps:

1. Specify a project name.
2. Specify a path for the project. This is the path under which all the files are stored. The MTK appends the project name as a subdirectory and includes that as part of the complete path. For this reason, we recommend that you use the same project path for all of your migration projects. They are stored in separate directories, since the project names are different, but you have the advantage of having them all grouped together.
3. Provide a text description for your project. Even if you are going to be migrating the same source Oracle database, you might want to create multiple projects to organize your work (for example, if you are doing the work in multiple steps). A good description helps when you are trying to distinguish the work you do for each project.
4. Specify the source database and version. (At this time, however, the only supported source database is Oracle version 8.)

5. Specify your target platform. This is the system on which DB2 is installed, and where you will be targeting the migration of your Oracle database. For iSeries migrations, you must choose the version and release of your iSeries server. This is very important, because different releases of DB2<sup>(R)</sup> Universal Database<sup>(TM)</sup> for iSeries support different functions. At this time, the supported version and release combinations are Version 5 Release 2 and Version 5 Release 3. If your iSeries server does not yet have either of these releases installed, choose V5R2. Certain functions might not run on your system, and will therefore fail to deploy properly; however, choosing V5R2 prevents the MTK from generating V5R3-only features of DB2.
6. Click **OK**.

### Example

When the MTK starts, you are immediately presented with a dialog box for Project Management. You type in payroll as the project name and leave the path name as the default value. You enter a meaningful description, to distinguish the project from any other migration projects that you might create, such as Migrating Payroll files for 1st Quarter.

Next, you specify DB2 UDB for iSeries as the target. Using the drop-down list, you change the platform to DB2 UDB iSeries V5R2 to indicate that the data will be migrated to an iSeries system running Version 5 Release 2. At this point, you are done setting up the project, so you click **OK**.

---

## Specify the source

The Oracle Migration Toolkit (MTK) lets you specify the source of the migration based on a set of script files, or directly from the Oracle database.

It is important to keep in mind that the data itself is not being processed during this stage. You are simply identifying the Oracle objects whose definitions will be translated into a form so that the same objects can be created into DB2 on your iSeries system. The movement of the data that is contained in these objects is part of a later step. This step creates an *empty* database on DB2 that closely matches your Oracle database.

### Specify the source using script files

You can identify a file (or set of files) that contains the SQL statements that you want to translate. Use this method when you already have scripts that contain all of the statements that are needed to create the Oracle database. Using this method, the MTK takes the SQL syntax from all of the input files and translates them from Oracle-compatible SQL to DB2-compatible SQL.

Make sure that the set of files includes all of the files that have interdependencies. This is because the translation process might refer to related objects when generating the DB2-compatible SQL. For example, if a script contains a CREATE INDEX statement, it would be necessary for the underlying table's CREATE TABLE statement to be included in the file or set of files being specified as the source. Even if you have already migrated the underlying table, you must include the CREATE TABLE statement.

The next phase of the migration process, **Convert**, lets you specify whether certain source files are intended only for context. This is useful in the case where you do not want to migrate the table again, but you need the CREATE TABLE statement in order for a dependent object, such as an index or view, to be properly translated.

To specify a set of script files for the migration, perform the following steps:

1. Determine if you have scripts that contain all of the SQL statements that are needed to create the Oracle database objects. These might include tables, views, indexes, procedures, triggers, functions, and so on.
2. Click **Import** to select all of the files that contain the SQL statements you want to translate. You can then view, sort, or remove the selected files as necessary.

3. You might want to sort the script files at this point to ensure that there is a logical flow of objects and their dependencies. Tables should be created first, followed by views, indexes, and triggers over those tables, followed possibly by any SELECT statements or CREATE PROCEDURE statements that rely on any or all of the objects. You can also sort the files during the **Convert** step.
4. Click the **Convert** tab to continue the migration process.

### Specify the source directly from the Oracle database

The second way to specify the source of the migration is to extract the descriptions directly from the Oracle database. Use this method when you do not have script files describing the Oracle database, or when you want to subset the objects to be migrated using the MTK itself.

To use this method, the Oracle database must be accessible on the machine you are using to run the MTK. This technique uses the Oracle catalogs and metadata to generate the lists of all the objects available for migration to DB2.

To specify the Oracle database directly, perform the following steps:

1. Click **Extract** to open the interface that connects to your Oracle database. Specify the name of the Oracle database that you wish to migrate, along with a user ID and password.
2. Once connected, all of the available objects within your Oracle database are listed. Select the objects that you want to migrate.
3. When you have made your selections, click **Extract** to produce a file containing the SQL statements. We recommend that you check the **Create one file / stored procedure** and **Include other needed objects** options. By creating one file, you can use the file for future migrations, since you've already completed the migration, and it can be a time-consuming process. By choosing to include other needed objects, you can let the tool identify any dependent objects that you might have forgotten.
4. Click the **Convert** tab to continue the migration process.

### Details

An SQL script file usually contains comments interspersed among the SQL statements. These appear as follows:

```
-- The following statement creates a table

create table oracle_worker(
    Name varchar2(25),
    Age number check(Age between 18 AND 65),
    Lodging varchar2(15));
```

Also notice that semicolons are used as ending delimiters for the SQL statements.

---

## Convert source metadata

In the previous steps, you created the migration project and defined the set of objects in your Oracle database that will be migrated to DB2<sup>(R)</sup> Universal Database<sup>(TM)</sup> for iSeries. This step uses these definitions to convert the SQL into DB2-compatible SQL statements.

To convert the source metadata to DB2-compatible scripts, perform the following steps:

1. Specify a prefix for the generated files, if you want to override the default value. By default, the tool uses the name of the highlighted source file, but you may want to change it to something more descriptive. This name is the prefix for the .db2 and .rpt files that are produced by the Convert step, and affects only the names of files generated by the MTK. It has no impact on the names of the database objects themselves, and does not affect the contents of the output scripts.

2. If there are any SQL statements that have date literals, you should make sure that you choose the correct **Source date format** from the available choices. If you do not have any date literals, or you simply are not sure, leave the default format.
3. Determine if any of the source files are only in the list to provide context for other objects. If so, click **Set Context** and select all the files that are used only to provide the definitions for objects that are referred to by other objects in the scripts.

The SQL statements in the files you identify as context files are not translated into the output scripts. Therefore, not all of your files can be context files. At least one file, and possibly several files, must remain as source files.

4. If you have not already set the file order as discussed in the **Specify Source** step, modify the order of the source files using the **Move File Up** or **Move File Down** buttons below the list of files.
5. If you desire, you can view the **Advanced Options** and **Global Type Mapping**, but we do not recommend that you change anything. The **Global Type Mapping** might be of interest to you if you want to see the Oracle types and the corresponding DB2 types that are used during the translation of the SQL statements.
6. Click **Convert**. This starts the conversion process, which can take several minutes, depending on the number and complexity of the statements that are being processed.

7. After the conversion process completes, the **Refine** tab opens, which displays the Translator information. However, at this point we recommend that you return to the **Convert** tab, where you can see the output files that the conversion process generated. Two files are particularly interesting:

The .db2 file is your actual output script. This contains all of the SQL statements from the input script, converted into DB2-compatible SQL. In some cases, statements are modified or even omitted. The .db2 file contains embedded comments that indicate where errors were encountered, where statements were purposely omitted, and where statements were modified. A nice feature of the .db2 file is that it retains each original SQL statement from the input script, and makes it into a comment. Each one is then followed by the equivalent SQL statement for DB2. This allows you to see how the conversion process modified each statement.

The .rpt file is a straight list of all the messages. This file does not contain any of the SQL statements. It is simply a report of all the messages that the conversion process generated. You can use this file to get an idea of how smoothly the conversion process went.

## Details

### Name qualification

Depending on how the input scripts have their objects qualified, the resulting scripts may or may not have qualified names for tables, views, indexes, and other SQL objects. For instance, if all of the objects in the input Oracle script are unqualified (they have no table space specified) then all of the objects in the output script will also be unqualified. If all objects in the input script are qualified using the same table space, the output script is still unqualified. For these cases, when you deploy to the iSeries server, the objects are implicitly qualified by the profile you use when you connect to DB2. If there are different qualifiers used in the input scripts, the first one encountered is the default, and any objects using this default table space appear unqualified in the output script. Any objects using a qualifier other than the default are qualified by that same schema name in the output script. A unique case occurs when the first reference to an object is qualified, but then later in the input script there are unqualified references. The MTK handles this case by making the first schema the default (and therefore leaving these unqualified in the output script) and qualifying any subsequent unqualified references using the schema name DBA.

Some simple examples are:

1st reference:	tab1	output:	tab1
2nd reference:	tab2	output:	tab2

1st reference:	tab1 table space x	output:	tab1
2nd reference:	tab2 table space x	output:	tab2

1st reference:	tab1 table space x	output:	tab1
----------------	--------------------	---------	------

```

2nd reference:  tab2 table space y  output:  y.tab2
1st reference:  tab1 table space x  output:  tab1
2nd reference:  tab2                output:  dba.tab2

```

### Set operations

DB2 UDB for iSeries does not support certain set operations, like INTERSECT and EXCEPT, in V5R2. When the Oracle scripts contain SQL statements that use these operations, they are replaced by UNION. This is done to insure that the output SQL is valid, but is likely to produce a different result than what you desire. We recommend that you review the places where these warnings occur, and decide if they should be rewritten or if the UNION is an acceptable replacement.

### SQL triggers

Other DB2 platforms allow trigger programs to implicitly refer to the old row image and the new row image with the identifiers OLD and NEW, respectively. DB2 UDB for iSeries requires that the SQL statement CREATE TRIGGER have an explicit line REFERENCING OLD AS OLD NEW AS NEW. This clause is added during the translation process for CREATE TRIGGER statements.

### Large Object (LOB) fields

The DB2 UDB for iSeries implementation of LOB fields does not support the NOT LOGGED attribute. Any of the Large Object types (CLOB, BLOB, and DBCLOB) have the NOT LOGGED clause removed when being translated to DB2 UDB for iSeries.

### CREATE INDEX

There are index attributes used within Oracle that do not exist on DB2 UDB for iSeries. Attributes such as CLUSTER and ALLOW REVERSE SCANS are supported by Oracle, but not by DB2 UDB for iSeries. Those clauses, if encountered in the input scripts, are omitted from the DB2 output scripts.

### NOT NULL clause

Other DB2 platforms require that columns included in UNIQUE and PRIMARY KEY constraints must be defined as NOT NULL. DB2 UDB for iSeries has no such requirement. Therefore, the clause is not be added when the target of the migration is DB2 UDB for iSeries.

### Mismatch of constraint columns

DB2 UDB for iSeries does not allow any differences between the data types of columns in a referential integrity constraint. The primary key column must have the exact same data type as the foreign key data type. One of the more common differences is INTEGER and SMALLINT. This relationship is allowed on other DB2 platforms, and on Oracle, but not DB2 UDB for iSeries. To lessen the chances of encountering this error, the translation changes the NUMBER data type in Oracle to INTEGER, and does not try to distinguish between INTEGER and SMALLINT. There still might be cases in which constraint columns are not identical, such as CHARACTER and VARCHAR, so you might still have to modify column definitions in some places.

### NULL; statements

The Oracle procedural language, PL-SQL, lets you use the NULL; statement to indicate an empty logic branch, as in the following trivial example:

```

CREATE TRIGGER TRIG1 AFTER INSERT ON T
FOR EACH ROW
BEGIN
    NULL;
END;

```

DB2 UDB for iSeries does not allow this use of the NULL; statement. To preserve the behavior of the PL-SQL logic, the NULL; statement is replaced by a valid statement that sets a dummy variable to 0, as in the following example:

```

CREATE TRIGGER TRIG1 AFTER INSERT ON T
REFERENCING NEW AS NEW FOR EACH ROW MODE DB2SQL
BEGIN ATOMIC

```

```

DECLARE NULLSTMT INTEGER;

--| NULL;

SET NULLSTMT = 0;

END;

```

### Statements that do not translate

Certain statements, when encountered in the input scripts, are not included in the output scripts. For some statements, such as DROP TABLE, it does not make sense to include them: since the migration is starting from scratch, dropping tables is not needed. Other statements have certain forms that are not valid within DB2. An example of this would be an ALTER TABLE that is enabling or disabling a constraint. Even though the ALTER TABLE statement is supported on DB2, it cannot be used to enable or disable a constraint. Finally, there are cases in which the statement itself cannot run on the iSeries server. An example of this is the CREATE SEQUENCE statement. This feature is not yet available in V5R2, so the entire statement is omitted. In all of these cases, the statement text is commented out, but you still see the text in the output script. Also, a message appears in the comments indicating that the statement was not translated, and the reason why it was not translated. These comments appear in both the .rpt and .db2 files, so you can easily check all the statements that were excluded.

### Oracle types

There is no DB2 support for types containing multiple elements, as in the following example:

```

create type name_ty as object
  (firstName varchar2(20),
   lastName varchar2(20),
   title varchar2(10));

```

This is another example of a statement that will not translate. If you encounter this, you should find any tables that use this type, and replace the reference to the type with the explicit column definitions. The tool does not handle these modifications to your CREATE TABLE statements, since it is not always clear what the user intends. But in the simple case, like this one:

```

create table persons
  (deptNo int primary key,
   deptName char(20) not null,
   managerID number(5),
   managerName name_ty not null);

```

You would include the columns from the type object, and rename them to something appropriate for the table, as follows:

```

create table persons
  (deptNo int primary key,
   deptName char(20) not null,
   managerID int,
   mgrfirstName varchar(20),
   mgrlastName varchar(20),
   mgrtitle varchar(10));

```

### SEQUENCE objects

Oracle SEQUENCE objects are not supported on DB2 UDB for iSeries. We were able to get the same behavior using IDENTITY columns, using these steps. If, for example, the sequence in Oracle was created with this statement:

```

CREATE SEQUENCE IDENTITY_C2
  INCREMENT BY 1
  START WITH 1000;

```

On the iSeries server, you would create a table with a single IDENTITY column:

```

CREATE TABLE IDENTITY_C2 (IDENTITY_C2 DECIMAL(31,0) GENERATED ALWAYS AS IDENTITY
  (START WITH 1000, INCREMENT BY 1));

```

Then, you create functions that provide the previous value and the next value of the sequence. You would go through your Oracle script and replace PREVIOUS FOR IDENTITY\_C2 with PREV\_IDENTITY\_C2( ) and replace NEXTVAL FOR IDENTITY\_C2 with NEXT\_IDENTITY\_C2( ). Following the same example, the statements to create the functions are:

```
CREATE FUNCTION PREV_IDENTITY_C2 ( ) RETURNS DECIMAL(31,0) LANGUAGE SQL
MODIFIES SQL DATA
BEGIN
    RETURN (SELECT identity_val_local() FROM QSYS2.QSQPTABL);
END;

CREATE FUNCTION NEXT_IDENTITY_C2 ( ) RETURNS DECIMAL(31,0) LANGUAGE SQL
MODIFIES SQL DATA
BEGIN
    INSERT INTO IDENTITY_C2 VALUES(DEFAULT) ;
    DELETE FROM IDENTITY_C2;
    RETURN (SELECT identity_val_local() FROM QSYS2.QSQPTABL);
END
```

When full support of SEQUENCE objects is added to DB2 UDB for iSeries, the MTK will allow the proper conversion of the Oracle scripts to the DB2 equivalent.

## Examples

You see a '25.12.2002' date literal in the input script, so you choose DD.MM.YYYY as your Source date format.

You have an ALTER TABLE statement that disables a constraint. To accomplish the same thing, you use the WRKPCFCST command directly on the iSeries system, since the ALTER TABLE statement cannot be used to disable a constraint. The WRKPCFCST command gives you the option of enabling and disabling constraints.

---

## Refine the metadata conversion

The purpose of the **Convert** step was to convert source SQL statements to DB2-compatible SQL statements. The **Refine** step gives you the opportunity to view the results of the conversion and to make changes. For example, you can change source stored procedures and some DB2 object names. However, you must return to the **Convert** step to apply these changes.

You can view the results of the conversion using any of the following tabs:

- The **Oracle** view displays the source database.
- The **DB2** view displays the target database.
- The **Report** view displays the error messages report, sorted by database object.
- The **Messages** view displays the messages, sorted by message number.

If an edit icon (a pencil) exists in the DB2 column, you can change the name of the object. For example, you can change the names of columns, tables, views, indexes, and procedures. You cannot change trigger and foreign key names. Changes made in the table are global and take effect only when you reconvert the source metadata. Perform the following steps to change an object name:

1. Select an object in the source or DB2 view of the Refine page.
2. Click on the edit icon (a pencil) in the DB2 column. The Edit Object Name window opens.
3. In the Name field, type the new name.
4. Click **Apply**. The new name appears in the table, but will not take effect until the next time you convert.

You can also change the logic of an SQL procedure or trigger:

1. Select a procedure or trigger from the source or target view of the Refine page.
2. Click **Edit Source**. The procedure (trigger) body opens in the default editor.
3. Enter any changes, save, and close the editor.
4. Click **Refresh Changes** to view the changes you made to the procedure (trigger) body. You can view the new changes here, but they do not take effect until the next time you convert.

## Details

The thing to remember about the **Refine** step is that, unlike the other steps, it is optional. In many cases, you can skip right over the refinement process when doing a migration. If you viewed the messages that came from the conversion process, and you do not have a need to modify any object names, you can skip this step entirely.

While you are refining a conversion, you might find it helpful to test the translation of individual statements. You can use the SQL Translator to accomplish this. First you must convert any objects that are referred to by the statement that you want to test. Once you have done the conversions of any related objects, or if the statement you want to test is completely self-contained, select **Tools > SQL Translator** from the MTK menu. The SQL Translator window opens; you can either type in the SQL statement or paste it from the clipboard. Then click the **Convert** button.

---

## Generate data transfer scripts

Now you are ready to generate the scripts that move the data from your Oracle tables into your DB2 tables on the iSeries. The scripts that we generated previously on the **Convert** step were only concerned with the creation of the database objects. This step produces the scripts that import the data from Oracle and populate your DB2 files. If you are interested only in deploying the script to create the objects, and do not want to move any data from the Oracle database into DB2, you can skip this step.

To generate the data transfer scripts, perform the following steps:

1. In the panel on the left side of the screen, highlight the script you want to use to transfer data. The script that you select is used to determine the Oracle tables whose data will be transferred. You do not need to go back and select the Oracle objects again, as you did on the **Specify Source** screen.
2. Decide where you want to store the scripts. Script files can be generated locally on the machine running the MTK, remotely on the iSeries server, or on both at the same time. The location that you choose on this panel is also used as the destination for the binary data files that are created during the data extraction step on the **Deploy to DB2** tab.

We recommend that you store the scripts in both places at the same time, and that you take the default locations for both sets of scripts. However, if you have a lot of data and plan to deploy to more than one iSeries server, you might want to generate the scripts only locally. This option runs much faster, and you can then manually send the files to the iSeries servers and store them in the directory shown. Another case in which you might choose to generate the scripts locally is if you are running disconnected from the iSeries server and you want to do only the work of generating the scripts now, to manually replicate the scripts later. In this scenario, you could also move the files later, into the iSeries destination directory shown on this window.

The only reason that you would choose to store the scripts only on the iSeries server would be to save storage on your local machine. In this case, you won't be able to view the \*.qsh file after it is generated, since it is not on your local machine. However, if no errors occurred during script generation, you can be sure that it exists on the iSeries, and everything is okay.

3. Click **Create Scripts**. This creates the scripts needed for data transfer, and stores them on the machines that you specified. This does not move any data yet. The actual data movement occurs on the final **Deploy to DB2** step of the migration process.

## Details

To replicate the data on multiple iSeries servers, the easiest method is to simply run the **Generate data transfer scripts** and the complete **Deploy to DB2** process for each iSeries. If for some reason the Oracle database is not available, or if it is simply too large to extract again, you can manually send the entire project directory to the target system.

- Binary copy the entire /QIBM/UserData/MTK/projects/*projname*/DataOutScripts data directory to the correct directory on the other iSeries server. The comments in the \*.qsh script file indicate exactly where they should be stored.
- Run the data deployment step from the MTK (skipping the extract step) against the new system. Alternatively, you could use STRQSH to manually run the qsh script on the new system, by executing /QIBM/UserData/MTK/projects/*projname*/DataOutScripts/*name*.qsh

The MTK tool generates several files in the /QIBM/UserData/MTK/projects/*projname*/DataOutScripts directory on the iSeries server:

- The \*.qsh script actually does the import of the data using CPYFRMIMPF
- The \*.fdf files define the fields for CPYFRMIMPF

If problems occur that cannot be diagnosed from the MTK logs or from the output messages, use the -debug option when you run the MTK. This option produces verbose output from the MTK tool and from the generation and execution of the data transfer scripts. This output might be useful for further problem determination, or possibly to send to IBM for analysis. In addition, you would also be able to manually issue the CL commands shown in the messages from the execution of the \*.qsh script to further diagnose problems. For instance, the CHGPF CST, DLTF, CRTSRCPF, and CPYFRMIMPF commands could be run one by one to try to pinpoint the cause of the problem.

The scripts themselves contain comments that show how to execute them, if you want to do it manually on the iSeries server instead of executing them from the MTK tool.

---

## Deploy to DB2 UDB for iSeries

You have reached the final step in the migration process. By this time, you have created all the scripts necessary to create the DB2 objects, extract the data from the Oracle tables, and import the data into the DB2 tables.

To deploy to DB2 UDB on iSeries, perform the following steps:

1. Enter the name of the iSeries server that is the target of the deployment. The name you enter must be a valid system name that the JDBC driver uses to access the remote system.
2. Make sure that the proper schema exists on the iSeries server. In the most common case, if you've extracted Oracle objects all from one Oracle schema, the generated DB2 scripts will not contain a schema name. This means that the default schema name is used, which is the user profile you use to connect. Most people already have a schema by this name, but if not, you should create one manually on the iSeries server. This is where the objects are created. You can create this schema using iSeries Navigator; you can also sign on directly to the iSeries server and use STRSQL to enter the interactive SQL environment and issue CREATE SCHEMA *myprofile*, where *myprofile* is the user profile you are using to connect.

If you do not want to use your user profile, you can modify the DB2 script to contain a SET SCHEMA statement at the beginning to set the default schema to something other than your user profile. However, we recommend using the default schema for the entire process. The objects can be copied or moved after the migration, if desired.

Another alternative, if you already have a schema and you absolutely must have your objects created into it, would be to do a CRTUSRPRF with that name, and use that profile to connect.

3. Enter the user ID and password for connecting to the iSeries. If you do not enter them on the initial screen, a window displays, which prompts you for the profile and password. The values you enter

must indicate a valid user profile on the iSeries server, along with the current password. If the profile is not defined, if it is not active, or if the password is either expired or incorrect, the connect fails and the deployment process stops.

4. Indicate whether you want to deploy the .db2 script only or extract and load the data also. For the complete migration, check all three boxes. These three options correspond to the three steps of a complete deployment:
  - a. Create the database objects in DB2
  - b. Extract the data from Oracle and store it on the iSeries server in intermediate form
  - c. Import the extracted data into the DB2 tables

We recommend that you perform all three steps together, but there are exceptions to that rule. If you only want to migrate the object definitions, but not the data, then check only the first option. If you have already created the database objects and now you just want to move the data, then check the second and third options. If you have copied all of the script files and data files from a one iSeries server to another iSeries server, then check only the third option.

5. Click **Deploy**. This step can be a very long-running process. The MTK tool performs several steps that can add to the processing time. All of the data is extracted and stored remotely, and therefore the performance is directly proportional to the amount of data in the Oracle tables. Once the data is extracted and stored on the iSeries server, the CPYFRMIMPF command runs for each table to import the data into DB2. If there are Large Object (LOB) columns in the tables, then each individual LOB value has its own file stored in the iSeries integrated file system that must be imported. Even if you choose to not extract the data, there can be many objects that need to be created in DB2, and this can also take several minutes.
6. Review the results of the deployment on the resulting output display. The MTK tool automatically generates very complete HTML files that contain useful information about the migration. When the deployment completes, a browser window displays; the window shows all of the objects that were migrated and, if you chose to do data movement, the complete number of rows from both the Oracle source database and the DB2 target database. The tool generates complete text for any errors or warnings that were encountered during the deployment process. All of these logs and reports are stored in the projects directory, along with all of the scripts and files, so your work is saved, and can be reused and reviewed even after your migration is complete.

## Details

In the event of a failure, the MTK tool generates a report that shows the objects that failed to deploy. By clicking on the object name, you can see the error message associated with the attempted creation of that object. There is also a deployment log that shows all of the messages that occurred during the entire deployment process. This file is called *filename\_deploy.log* and resides in the projects directory with all of the scripts. There is also a direct link to this log from the HTML deployment report. This log shows both the messages from the deployment of the metadata as well as the deployment of the data itself. If there are problems that can be fixed by manually updating the script or by creating or deleting something on the iSeries, you can rerun the entire script. You might want to DROP and CREATE the schema before you do this, to avoid getting *already exists* errors on every object. Another option, if you do not want to rerun the entire script, is to either create a new script with just the statement that failed, or manually execute that SQL statement in interactive SQL or with iSeries Navigator.

During the first deploy to an iSeries server, the MTK tool generates user-defined functions (UDFs) for many of the Oracle built-in functions. These UDFs are built into the ORA8 schema. This process is completely automated and is performed only once, unless someone deletes the ORA8 schema or deletes one or more of the UDFs contained in the ORA8 schema.

**IMPORTANT:** To ensure that all of the necessary code is on your iSeries server, check that the PTF SI06748 and SI06675 are applied to the system before doing the deployment. Without these PTFs, the deploy process will not complete properly.

---

## Chapter 2. Disclaimer

Information is provided "AS IS" without warranty of any kind. Mention or reference to non-IBM products is for informational purposes only and does not constitute an endorsement of such products by IBM.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.







Printed in USA