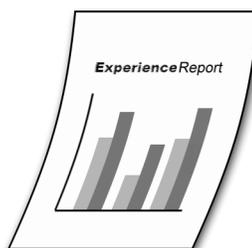


System i



IBM Migration Toolkit

Experience Report



System i



IBM Migration Toolkit

Contents

Chapter 1. IBM Migration Toolkit 1
Downloading and installing the MTK 1
Using the MTK 1
Finding more information 2
Starting the IBM Migration Toolkit 2
Managing migration projects 3
Specify the source 5

Convert source metadata 7
Refine the metadata conversion. 11
Generate data transfer scripts 13
Deploy to DB2 UDB for i5/OS 17

Chapter 2. Disclaimer 21

Chapter 1. IBM Migration Toolkit

The IBM Migration Toolkit (MTK) automates the process of migrating a database from competing database products to IBM DB2^(R) Universal Database^(TM) and Informix^(R). This document deals with migrating databases from the following sources to DB2^(R) Universal Database^(TM) for i5/OS.^(TM)

- Oracle, Versions 9i and 10g
- Sybase Adaptive Server Enterprise (ASE), Versions 11, 12 and 12.5. Note that Sybase SQL Anywhere will also appear as a selection for your project, but is not supported for System i^(TM) migrations.
- Microsoft^(R) SQL Server, versions 7, 2000 and 2005
- Informix^(R) Dynamic Server, versions 7.3 and 9. XPS versions 8.3, 8.4 and 8.5

This document describes MTK version 1.4.5.0. The MTK is only available in the English language. Please note that for the purposes of generating this report, we did not use the "migration wizard" portion of the tool. Also, we describe the use of the MTK on Microsoft^(R) Windows^(R), and not the Unix^(R) platforms that the MTK supports. We recommend that you run on Microsoft Windows. See the MTK readme file for the client operating systems that are supported.

Downloading and installing the MTK

The IBM Migration Toolkit is a free download. To download the MTK and see other migration resources, go to the DB2 Porting Information from IBM PartnerWorld (<http://www.ibm.com/servers/enable/site/db2/porting.html>).

A new requirement in MTK 1.4.5.0 is that you must have a Java^(TM) Run Time Environment (JRE) installed on your system. In previous releases, the JRE was shipped with the MTK which greatly increased the download size. The MTK readme file describes the required version of the JRE, the steps to check if you currently have a JRE installed and what version your JRE is. You can download the JRE from Sun Microsystems^(TM) at <http://java.sun.com>. If available on your PC, the IBM Runtime Environment is also acceptable. If you have an IBM PC and do not have the IBM 32-bit Runtime Environment for Windows, you may download it at <http://www.ibm.com/developerworks/java/jdk/index.html>.

Using the MTK

The first thing you will do in the MTK is create a project. Then you will use the MTK five-step process to migrate a database. Each is explained in detail below.

Starting the MTK and managing migration projects

The five step process:

Step 1: Specifying the source

Step 2: Converting the source metadata

Step 3: Refining the metadata conversion

Step 4: Generating data transfer scripts

Step 5: Deploying to DB2^(R) for i5/OS^(R)

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^(R) and Informix^(R) platforms, certain details in the generic help either don't apply to the System i5^(TM) 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 System i5 migration process.

You can ask questions or report problems about the MTK tool by sending email to mtk@us.ibm.com.

You can find comprehensive information about DB2^(R) Universal Database^(TM) for i5/OS^(R) in the iSeries Information Center (<http://www.ibm.com/eserver/series/infocenter>).

Starting the IBM Migration Toolkit

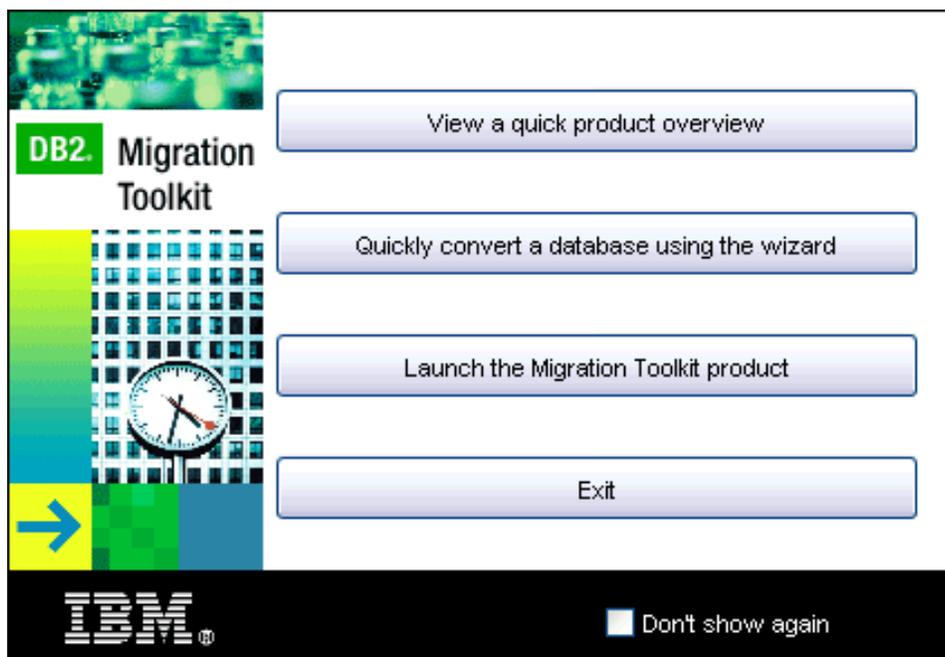
After installing the IBM Migration Toolkit (MTK) on Microsoft^(R) Windows^(R) you will have a program folder called 'IBM Migration toolkit 1.40'.



The 'Documentation' shortcut will take you to the MTK online help. You may also view the help in a PDF and print it.

'Wizard' will start the MTK wizard interface for a new migration. 'Toolkit' is the shortcut you will normally use to start the MTK.

When you select 'Toolkit', you will be presented with a startup dialog where you may choose to view a quick product overview in the online help, start the wizard or start the main toolkit. You must choose 'Launch the Migration Toolkit product' to work with existing projects.



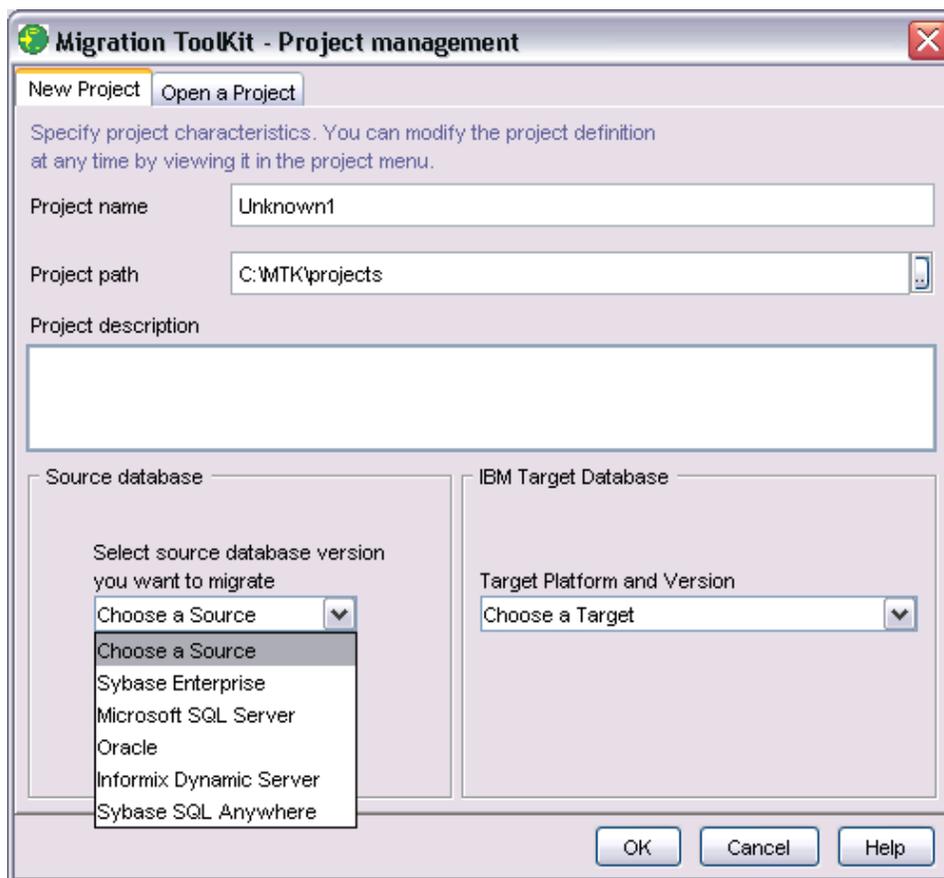
If you select the 'Don't show again' checkbox, this dialog will no longer appear and the MTK will automatically open the last project you worked with. This will take effect the next time you start the MTK.

You may find it difficult to find the start up dialog if you switch to another window. Use ALT+TAB to switch back to the start up dialog. Look for the Java^(TM) coffee cup icon.

Managing migration projects

The 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.

To create a new project in the MTK, enter the following information on the Project management dialog:



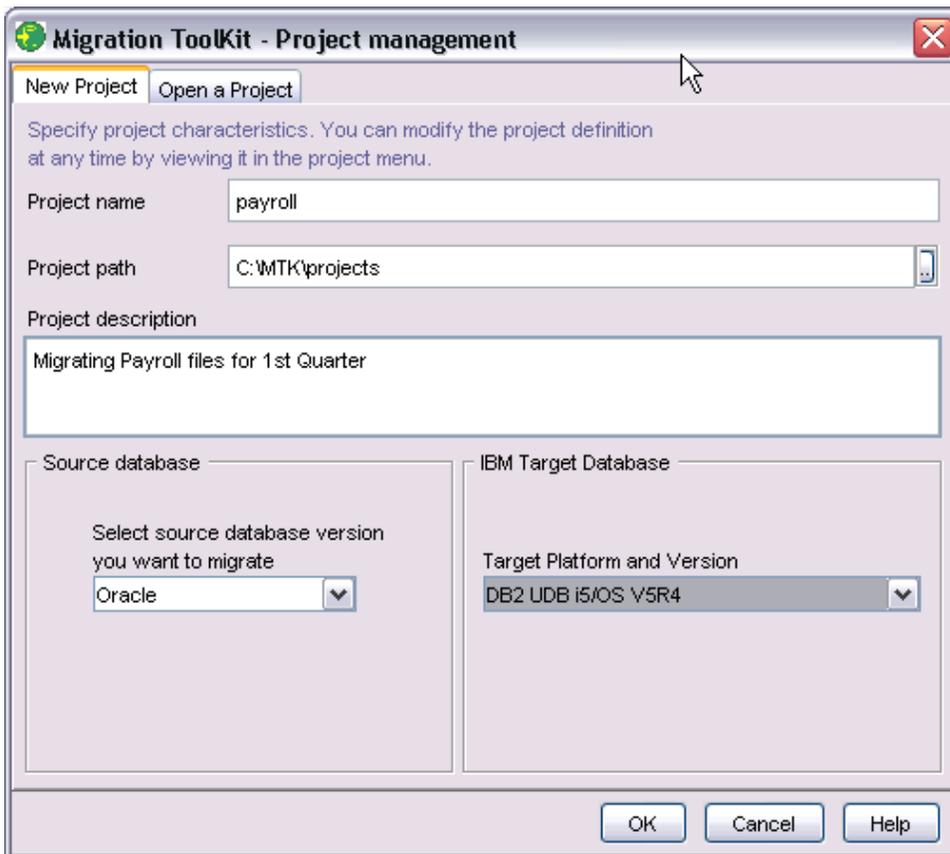
1. Specify a project name.
2. Specify a path for the project. This is the path under which all the files are stored. The MTK appends the project name as a subdirectory and includes that as part of the complete path. For this reason, we recommend that you use the same project path for all of your migration 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 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.

5. Specify your target platform. This is the system on which DB2 is installed, and where you will be targeting the migration of your source database. For System i5 migrations, you must choose the version and release of your system. This is very important, because different releases of DB2 UDB for i5/OS support different functions. At this time, the supported versions are V5R2M0, V5R3M0 and V5R4M0. If your system does not yet have either of these releases installed, choose 'DB2 UDB iSeries V5R2'. Certain functions might not run on your system, and will therefore fail to deploy properly; however, choosing V5R2 prevents the MTK from generating new features of DB2.
6. Click **OK**.

Example

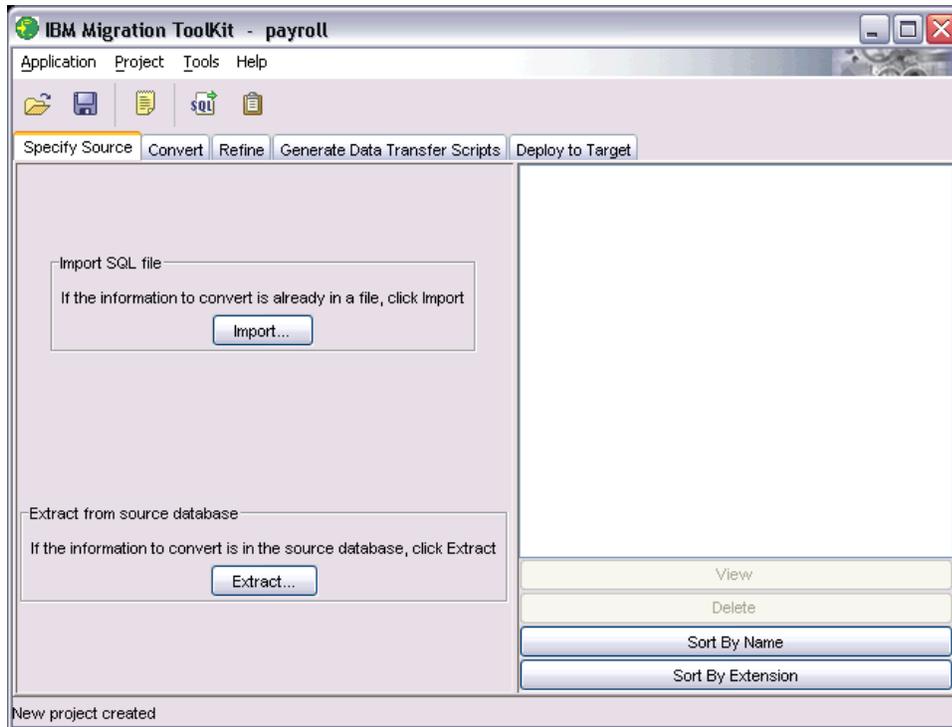
When the MTK starts for the first time, you are presented with a dialog box for Project Management. You type in "payroll" as the project name and leave the path name as the default value. You enter a meaningful description, to distinguish the project from any other migration projects that you might create,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

Next, choose "Oracle" from the 'Source database' drop-down list. Then select "DB2 UDB i5/OS V5R4" from the 'IBM Target database' Targets the target database using the drop-down list. At this point, you are done setting up the project, so you click **OK**.



Specify the source

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



It is important to keep in mind that the data itself is not being processed during this stage. You are simply identifying the source objects whose definitions will be translated into a form so that the same objects can be created into DB2 for i5/OS. The creating of the objects and the movement of the data that is contained in these objects is part of a later step.

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 source database. Using this method, the MTK takes the SQL syntax from all of the input files and translates them from source-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 source 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.
Note: By default, the 'Import file' dialog will only allow you to select one file at a time. You can change this behavior by going to the 'Application' menu and selecting 'User preferences'. Under 'File import', select the 'Java look-and-feel (allows selection of multiple files)' radio button. Now you will be able to select multiple files in the 'Import file' dialog.
3. Click the **Convert** tab to continue the migration process.

Specify the source directly from the source database

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

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

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

1. Click **Extract...** to open the interface that connects to your source database. Specify the name of the source database that you wish to migrate, along with a user ID and password. You may use an ODBC data source, or you can use a JDBC^(TM) connection if the appropriate driver files are referenced in your CLASSPATH environment variable.
2. Once connected, all of the available objects within your source 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:

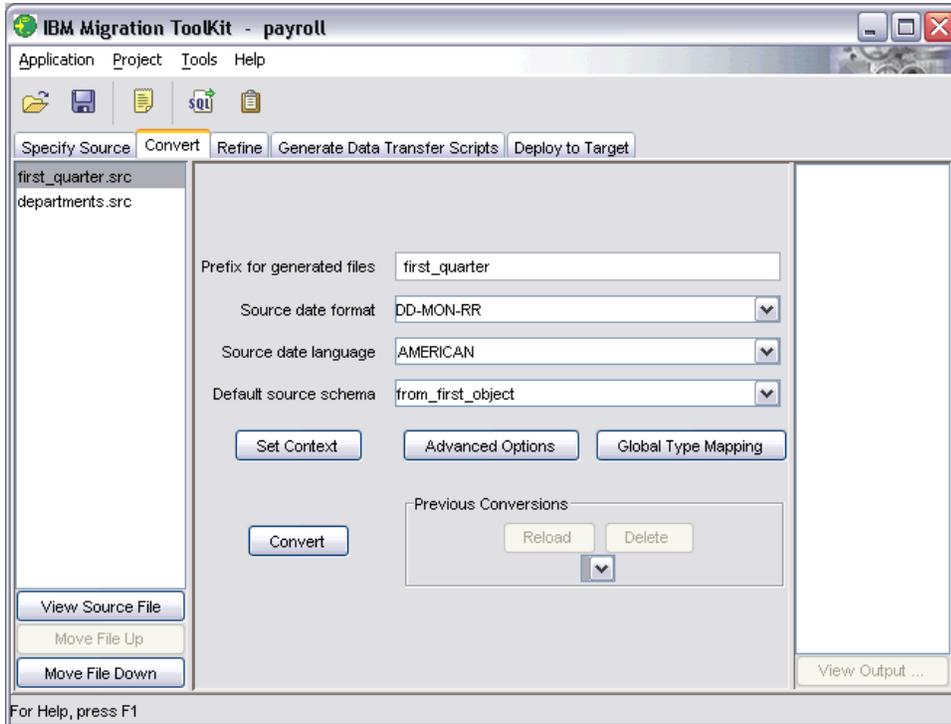
```
-- The following statement creates a table

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 in this example. For Oracle SQL procedures and triggers, a forward-slash (/) will appear. For Sybase and SQL Server, the statements are delimited by the word "go".

Convert source metadata

In the previous steps, you created the migration project and defined the set of objects in your source database that will be migrated to DB2^(R) Universal Database^(TM) for i5/OS^(R). 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. Choose the default source schema. When encountered in the source script, the MTK will omit this schema. As such, when the converted DB2 script is executed, the object will be created in the default schema. You can choose from the following options:
 - a. "from_first_object" - This is the default. The first schema referenced in the source script will become the default source schema.
 - b. "specify_schema_for_all_objects" - No schema becomes the default source schema. All source statements with a schema reference will reference the schema in the converted DB2 script. Note that if you use this option, the 'Target schema' field on the generate scripts and deploy steps will not be used if every object is qualified with a schema.
 - c. Enter a schema name that will be used as the default source schema.
4. 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.

5. 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. Use the **Move File Up** or **Move File Down** buttons below the list of files.
6. 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 source data types and the corresponding DB2 data types that are used during the translation of the SQL statements. In You can also use this dialog to map VARCHAR data to CHAR which can offer a performance advantage on DB2 UDB for i5/OS.
7. 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.
8. 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 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. These error messages are also viewable in the **Refine** stage, and in the HTML report files generated by the tool.

You can access the reports at any time by going to the 'Tools' menu and selecting 'Migration Reports', or by pressing the yellow notebook icon on the toolbar.

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 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 System i5, 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 i5/OS does not support certain set operations, like INTERSECT and EXCEPT, in V5R2. When the 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 i5/OS 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 i5/OS 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 i5/OS.

CREATE INDEX

There are index attributes used within the source database that do not exist on DB2 UDB for i5/OS. Attributes such as CLUSTER and ALLOW REVERSE SCANS are supported by Oracle, but not by DB2 UDB for i5/OS. Those clauses, if encountered in the input scripts, are omitted from the DB2 output scripts.

NOT NULL clause

Other DB2^(R) platforms require that columns included in UNIQUE and PRIMARY KEY constraints must be defined as NOT NULL. DB2 UDB for i5/OS has no such requirement. Therefore, the clause is not be added when the target of the migration is DB2 UDB for i5/OS.

Mismatch of constraint columns

DB2 UDB for i5/OS 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 for i5/OS. 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 i5/OS 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 System i5. 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 V5R2 of DB2 for i5/OS. 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 System i5, 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 PREVVAL 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
```

With V5R3 of DB2 UDB for i5/OS, the MTK will allow the proper conversion of the Oracle scripts to the DB2 equivalent.

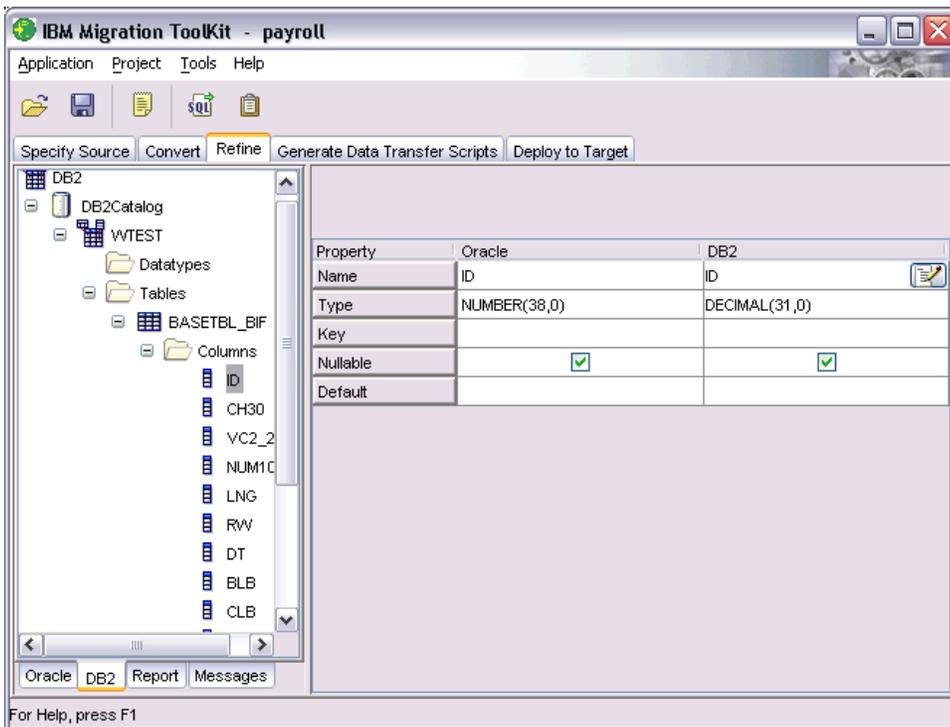
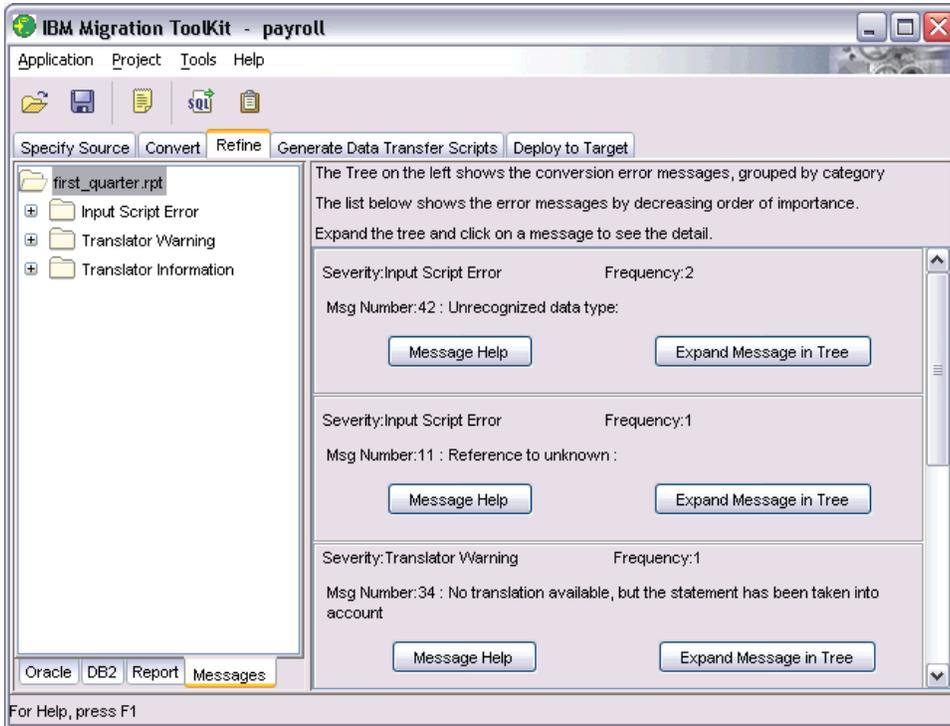
Examples

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 WRKPFCS command directly on the System i5, since the ALTER TABLE statement cannot be used to disable a constraint. The WRKPFCS command gives you the option of enabling and disabling constraints. You can also enable and disable constraints in iSeries Navigator by right-clicking the constraint and selecting 'Enable' or 'Disable'.

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 **Source** view displays the source database. If Oracle is the source, this view is named **Oracle**
- The **Target** view displays the target database. If Oracle is the source, this view is named **DB2**.
- 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.

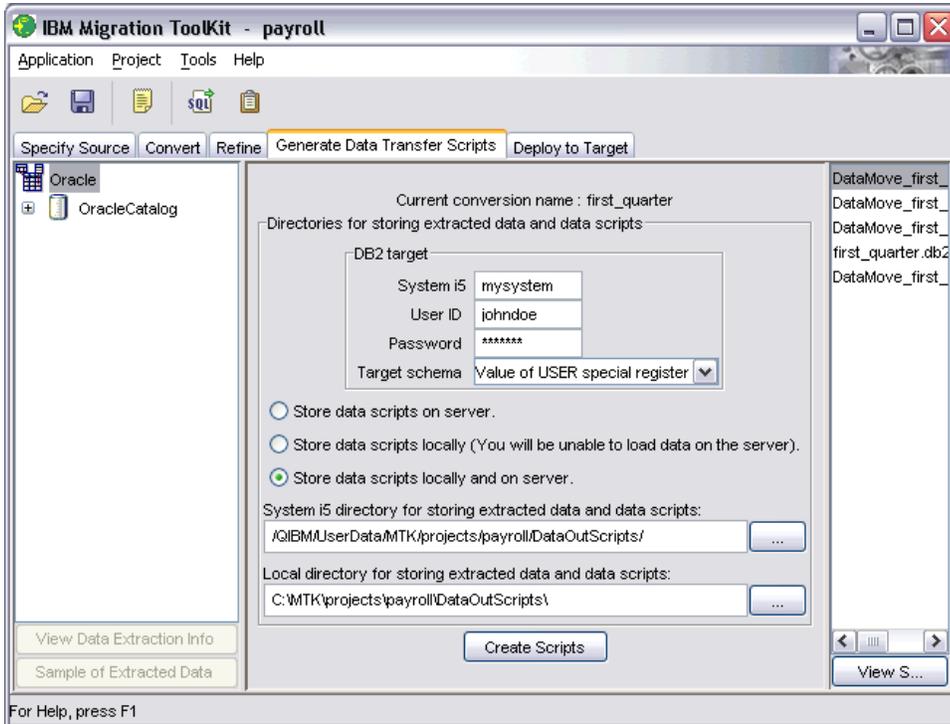
Details

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, go to the 'Tools' menu and 'SQL Translator'. 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 source tables into your DB2 tables on i5/OS. 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 the source database 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 source database into DB2, you can skip this step.



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

1. Decide where you want to store the scripts. Script files can be generated locally on the machine running the MTK, remotely on the System i5, or on both at the same time. The location that you choose on this panel is also used as the destination for the data files that are created during the data extraction step on the **Deploy to Target** 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 system, you might want to generate the scripts only locally. This option runs much faster, and you can then manually send the files to the target server 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 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 System i5 destination directory shown on this window.

The only reason that you would choose to store the scripts only on the system 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 System i5, and everything is okay.

2. If you selected the 'Store data scripts on server.' or 'Store data scripts locally and on server' option, enter the name of the System i5 that you want to deploy the objects and data. Also enter your user profile and password.
3. Starting in version 1.4.5.0, you can specify a target schema where the MTK will create your objects. In the past, the MTK would deploy to a schema with the same name as the user profile you specify. You can use the default "Value of USER special register" if you want that behavior, or you can enter your own schema name, such as "PAYROLL".

The image shows a dialog box titled "DB2 target" with the following fields:

System i5	mysystem
User ID	johndoe
Password	*****
Target schema	PAYROLL

The MTK will remember all of the target schemas you used to create scripts. You can choose a different schema for the next time you generate scripts if you wish.

The target schema field is not available in the MTK Wizard interface. See the the deploy step for more information on the default target schema.

4. 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 or create any database objects yet. The actual data movement occurs on the final **Deploy to Target** step of the migration process.

Details

To replicate the data on multiple systems, the easiest method is to simply run the **Generate data transfer scripts** and the complete **Deploy to Target** process for each system. If for some reason the source 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.

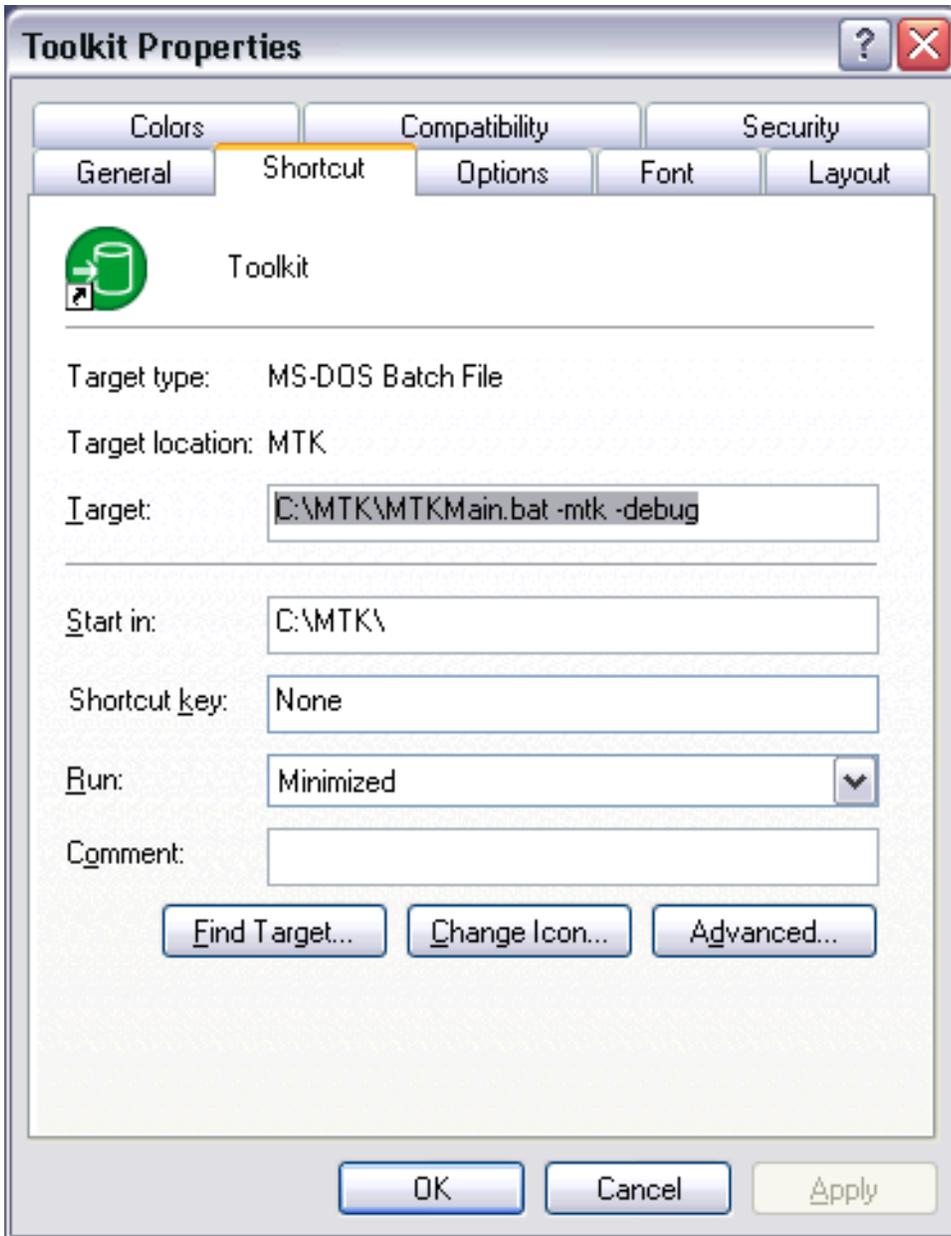
- Binary copy the entire /QIBM/UserData/MTK/projects/*projname*/DataOutScripts data directory to the correct directory on the other System i5. The comments in the *.qsh script file indicate exactly where they should be stored.
- Run the data deployment step from the MTK (skipping the extract step) against 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 server and/or C:\MTK\projects*projname*\DataOutScripts\ directory:

- The *.qsh script that contains the steps required to import of the data using the CPYFRMIMPF CL command.
- The *.fdf Field Definition Files define the format of the data file for each extracted source table. CPYFRMIMPF requires this information so that it can extract data from the data files that the **Deploy to Target** tab will produce, and import the data into the DB2 tables.

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.

You can set the debug option by going to 'Start/All Programs/IBM Migration Toolkit 1.4' and then right click on the 'Toolkit' icon and select 'Properties. Append " -debug" (no quotes) to the contents of the 'Target' field on the 'Shortcut' tab and press the **OK** button.



As an alternative, you can add " -debug" to the end of the 'java' command in C:\MTK\MTKMain.bat. You must turn off the read only attribute on the file to do this.

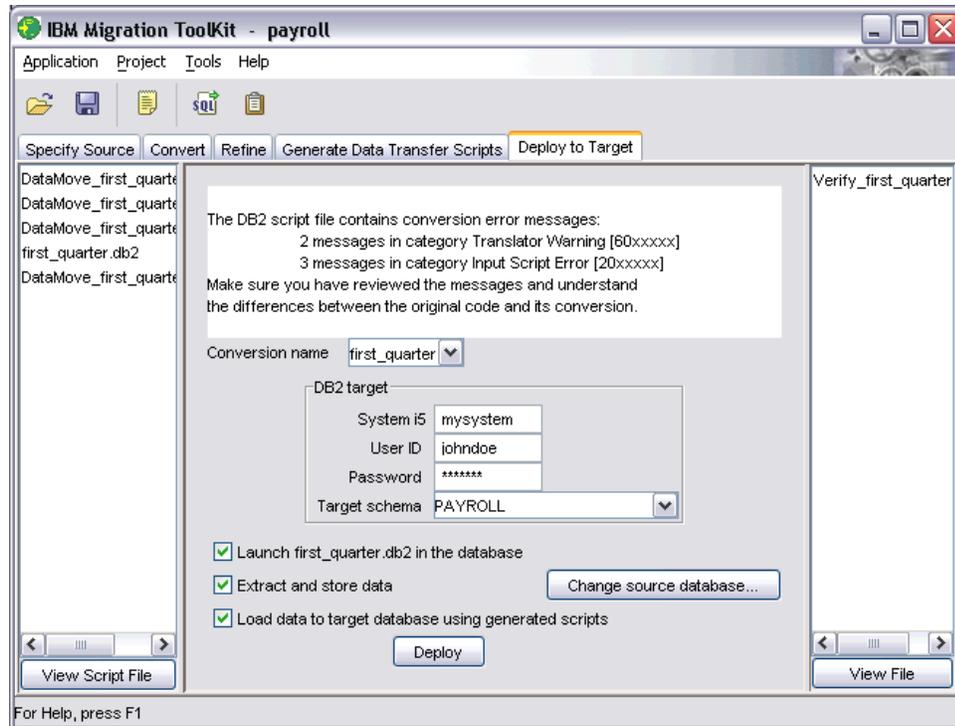
The debug messages will appear in C:\MTK\mtk.log and in the 'Toolkit' command window. In debug mode, the System i5 job logs will be saved to a spooled file when the job ends. Use WRKSPLF <my-user> to see the spooled file. You can use the 'Basic Operations/Printer Output' view in iSeries Navigator to view the spooled files and copy them to your PC via drag and drop.

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, CRTSRC PF, and CPYFRMIMP F 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 System i5 instead of executing them from the MTK tool. You can also change the default target schema by changing the 'MTKTARGET' environment variable near the top of the script.

Deploy to DB2 UDB for i5/OS

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 tables in the source database, and import the data into the DB2 tables.



To deploy to DB2^(R) Universal Database^(TM) for i5/OS^(R), perform the following steps:

1. If you did not generate data scripts in the previous step, enter the name of the server that is the target of the deployment. The name you enter must be a valid system name that the JDBC^(TM) driver uses to access the remote system.
2. Enter the user ID and password for connecting to the system. 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 System i5, 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.
3. As with the generate data scripts step, starting in version 1.4.5.0, you can specify a target schema where the MTK will create your objects. The target schema you specified when you created the data scripts will be shown here. The schema you specify here should be the same as the schema you used to create the data scripts with, or the data may fail to import, or possibly be imported into the wrong tables. The MTK will warn you if the schemas are not the same.

In the past, the MTK would deploy to a schema with the same name as the user profile you specify. You can use the default "Value of USER special register" if you want that behavior, or you can enter your own schema name, such as "PAYROLL".

DB2 target	
System i5	mysystem
User ID	johndoe
Password	*****
Target schema	PAYROLL <input type="button" value="v"/>

If you enter a name for a schema, the MTK will create the schema, if it does not already exist, when you press the **Deploy** button. If you selected the "Value of USER special register" option, you must create the schema manually. Most people already have a schema with this name. You can create this schema using iSeries Navigator; you can also sign on directly to the System i5 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.

The behavior of the target schema field also depends on the choice for the default source schema you selected on the Convert step. If you selected "specify_schema_for_all_objects", the target schema may not be used, as every object may already be qualified with a schema.

The MTK also does not create other schemas that may be referenced in the script. You must create those schemas manually following the steps above.

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 the source database and store it on the system 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 System i5 to another, then check only the third option. If you need to migrate across sites and do not have access to the source and target databases, see "Disconnected locations" below.

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 source tables. Once the data is extracted and stored on the System i5, 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 System i5 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 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 System i5, 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 deployment to a system, the MTK tool generates user-defined functions (UDFs) for many of the source database built-in functions. These UDFs are built into a schema specific to the source database, which is called the 'Compatibility Library' by the MTK online help:

- ORA for Oracle
- MSSQL for SQL Server
- SYB for Sybase
- INFX for Informix

Note that the Oracle and SQL Server names have changed in this version of the MTK. Previous versions used ORA8 and MS7, respectively.

This process is completely automated and is performed only once, unless someone deletes the schema or deletes one or more of the UDFs contained in the schema.

IMPORTANT: Please ensure you have a recent Database group PTF loaded on the System i5, and a recent PTF for the CPYFRMIMPF CL command. At a minimum, we recommend the following PTFs for your target system:

V5R2 SI06748
SI08237

V5R3 SI13843
SI13050
SI18900

V5R4 SI21359

Disconnected locations

You can also migrate a database from a remote, disconnected location. The site that hosts the source database and the site that hosts the System i5 may be separate and the MTK cannot communicate with the source and target at the same time. At the site that hosts the source database, perform the following steps:

1. At the first site, complete the first 3 steps of the MTK, **Extract** through **Refine**.
2. **Generate data transfer scripts** using the 'Store data scripts locally' option.
3. On the **Deploy to Target** step, choose only the second option to 'Extract and store data'.
4. Send the contents of the C:\MTK\Projects\proj directory to the site that hosts the System i5.

At the site that hosts the System i5, perform the following steps:

1. Restore the project with one of the following methods
 - a. Copy the project first and specify the project file (.mtk file) in the Project Management dialog's 'Existing file' field.
 - b. Create a project with the same name in the MTK. Close the MTK. Copy the project directory from the source site to the C:\MTK\Projects directory.

2. Copy the .qsh, .out and .fdf files in the 'DataOutScripts' directory to a 'DataOutScripts' subdirectory of the System i5 directory specified on the **Generate Data Transfer Scripts** step. For example, /QIBM/UserData/MTK/projects/*projname*/DataOutScripts/.
3. Go to the **Deploy to Target** step in the MTK
4. Update the system, user, password and target schema fields as needed.
5. Select the 'Launch *conversion-name.db2*' and 'Load data to target database using generated scripts' check boxes.
6. Press the **Deploy** button.

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.

DB2, DB2 Universal Database, i5/OS, IBM, Informix, iSeries, System i and System i5 are trademarks of International Business Machines Corporation in the United States, other countries, or both.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft and Windows are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the US and other countries.

Other company, product or service names may be trademarks or service marks of others.



Printed in USA