



IBM DB2 for i5/OS and DB2 Developer Workbench

*Using DB2 Developer Workbench to create, test and deploy your DB2 for
i5/OS SQL and Java stored procedures*

*Kent Milligan and Gene Cobb
ISV Business Strategy and Enablement
December 2007*



Table of contents

Abstract	1
Introduction	1
Creating a project	2
Creating a new procedure	7
Creating an SQL procedure	7
Creating a Java procedure	15
Importing procedures	21
Running a procedure	22
Repeating stored-procedure tests	22
Exporting and deploying procedures	23
Exporting to a deployment script.....	23
Deploying to target database	25
Summary	27
Resources	28
About the authors	29
Trademarks and special notices	30



Abstract

This white paper helps you understand the IBM DB2 Developer Workbench (formerly known as IBM DB2 Development Center) capabilities that help you in the creation, testing and deployment of SQL and Java stored procedures and user-defined functions on DB2 for i5/OS.

Introduction

IBM® DB2® Developer Workbench is a rapid iterative, client-based development environment that is based on Eclipse. It supports the entire DB2 family of database offerings; therefore, it is an especially useful tool if you are developing and deploying procedures on multiple DB2 server platforms. DB2 Developer Workbench is the replacement product for the IBM DB2 Development Center Version 8.2 and the IBM DB2 Stored Procedure Builder products that were first introduced in DB2 Version 7.

DB2 Developer Workbench Version 9 supports the development of SQL and Java™ stored procedures when used with DB2 for i5/OS servers or any other DB2 server. Even though DB2 for i5/OS supports user-defined functions and external stored procedures that are written in high-level languages such as RPG and COBOL, DB2 Developer Workbench does not support the creation of these objects on the IBM System i™ platform. You can also use the DB2 DWB debugger to debug SQL stored procedures that were created on a DB2 for i5/OS V5R4 server. For more information about using the DB2 Developer Workbench debugger, see ibm.com/servers/enable/site/education/ibo/record.html?2ae2.

The easiest way for developers on the on the IBM i5/OS® operating system to access the DB2 Developer Workbench tool is to download it from the following Web site: ibm.com/software/data/db2/ad/dwb.html. DB2 Developer Workbench is also included in DB2 Connect Unlimited Edition for iSeries.

Creating a project

DB2 Developer Workbench manages the development of stored procedures and function objects through the use of *projects*. Projects keep track of the database connections and associated procedure objects. For example, you can use a single project to group and manage all of the stored procedures that are associated with a single application (for example, payroll application). To launch DB2 Developer Workbench and create a new project, take the following steps:

1. From the Windows **Start** menu, click **All Programs** → **IBM DB2** → **DB2 Developer Workbench** → **Developer Workbench**.

You are given a choice to create a new project or to open an existing project. The Data Development Project window appears (see Figure 1) when creating a new project. This interface allows you to override the default schema value (the authorization ID) that is used during execution of the stored procedure for any SQL statements that contain unqualified table or view references. In this example, the default value is *Authorization ID*. The i5/OS equivalent of an authorization ID is a user profile.

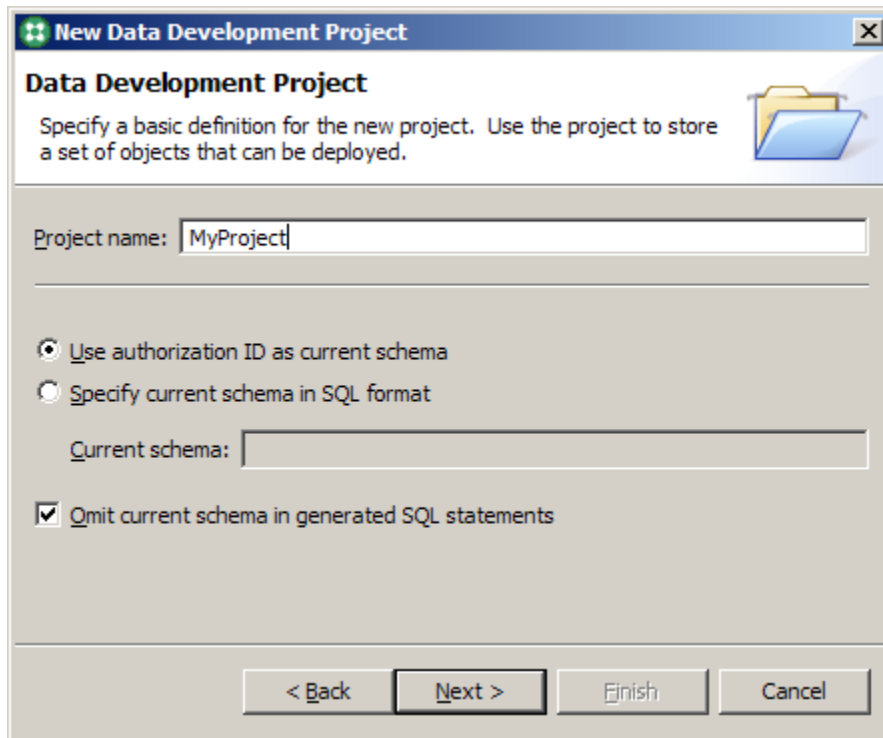


Figure 1. Developer Workbench Project Creation window

2. Click **Next** to continue the project creation process.

- The next step is identifying the connection to use for the new project from the Connection window (Figure 2). You can select a previously created database connection by left-clicking the connection name or by creating a new connection.

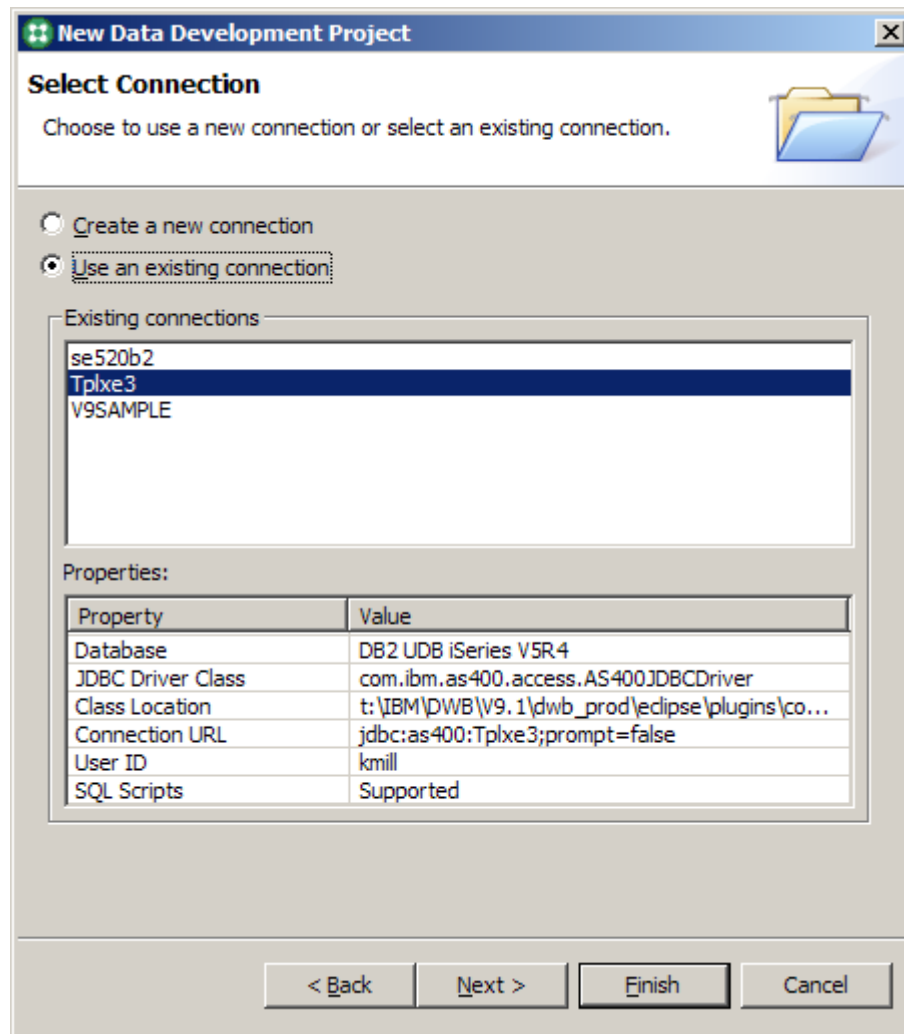


Figure 2. Select Connection window

4. If you need to define a new connection, the Connection Parameters window (shown in *Figure 3*) appears. The default naming convention results in the generation of a connection name from the database or host name for your i5/OS server.

On the **JDBC Driver** pull-down menu, you have three options:

- **Option 1:** The majority of i5/OS programmers select **AS/400 Toolbox for Java** because this JDBC driver is available at no charge. The IBM Toolbox for Java is automatically included with the i5/OS operating-system software. This JDBC driver is also available for download by visiting the JOpen Web site (<http://jt400.sourceforge.net>). JOpen is the open-source version of the IBM Toolbox for Java.

The **Host** name is the TCP/IP host name of your i5/OS server.

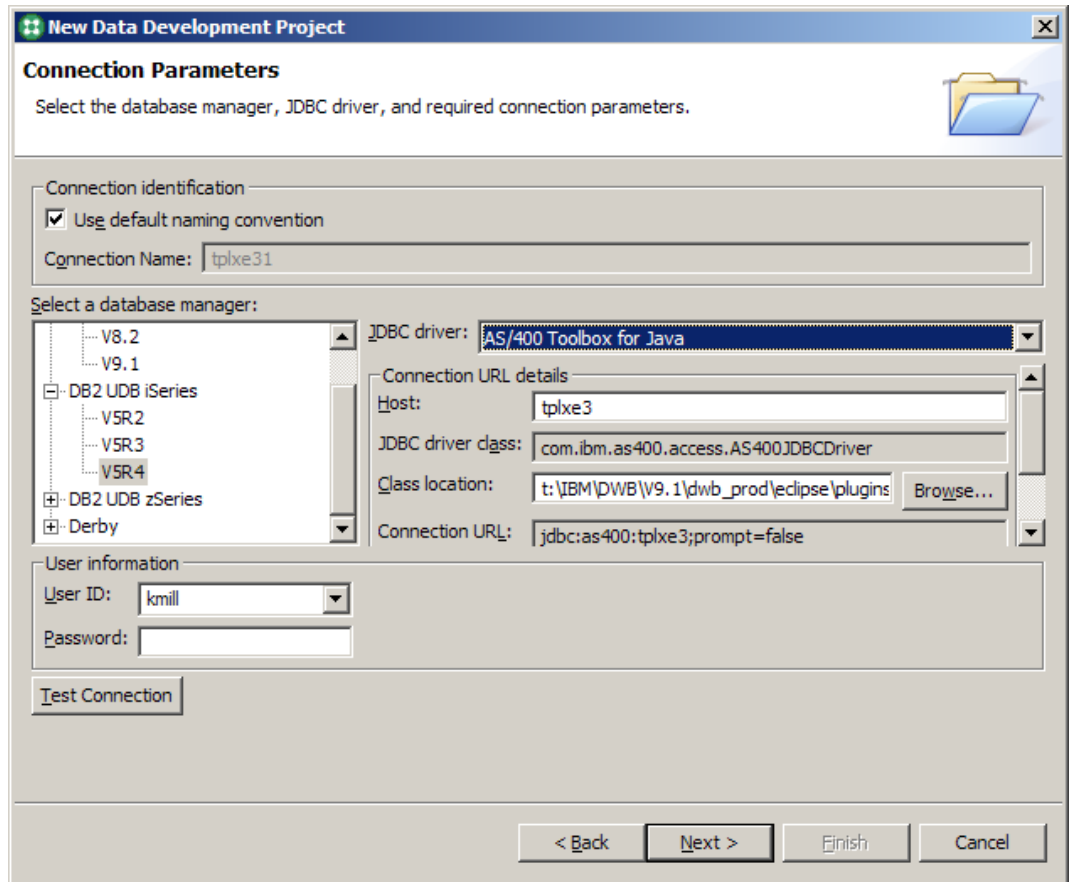


Figure 3. Connection Parameters window



- **Option 2:** It is also possible to use the IBM DB2 Universal JDBC driver. You must select this driver if you want to use the DB2 Developer Workbench debugger.

The IBM DB2 Universal JDBC driver is also a Type-4 JDBC driver (as is the Toolbox JDBC driver). In contrast, the DB2 Universal JDBC driver is not included within i5/OS and requires the purchase of an additional license to access DB2 for i5/OS servers. You can obtain the software and license for evaluation and development purposes by downloading the DB2 Connect™ Developer's Edition from one of the following Web sites:

- *Microsoft® Windows® client:*
www14.software.ibm.com/webapp/iwm/web/preLogin.do?lang=en_us&source=swg-db2cdew
- *Linux® client:*
www14.software.ibm.com/webapp/iwm/web/preLogin.do?lang=en_us&source=swg-db2cdel

For production usage, the DB2 Connect Unlimited Edition for iSeries packaging usually offers the best terms for System i users.

For the DB2 Universal JDBC driver, the **Database** name is obtained by issuing the IBM i5/OS® WRKRDBDIRE command on the command line. The **Host** name is again the TCP/IP host name of the i5/OS server.

- **Option 3:** Select **Other** to use a driver from a third-party software provider.
5. You can either verify that you have set up the connection correctly by clicking **Test Connection** or simply move on by clicking **Next**.

- The Specify Schema Filter window (see Figure 4) allows you to include particular schemas that are accessible, or inaccessible, to a database connection.

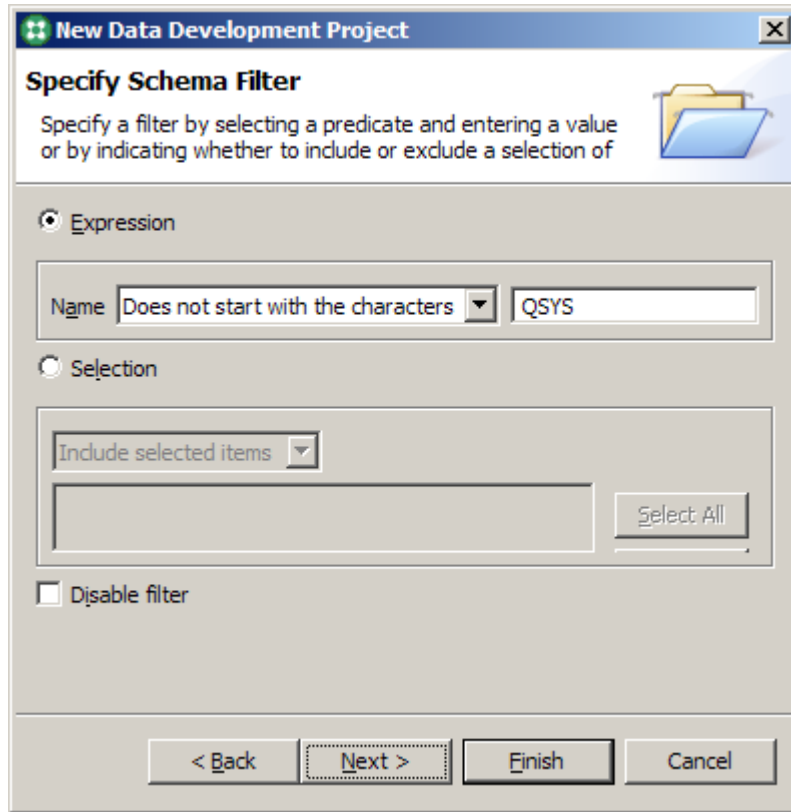


Figure 4. Specify Schema Filter window

- The final Connection configuration option, **JDK home**, does not apply to DB2 for i5/OS servers, so just select **Finish** to complete the project creation.

Creating a new procedure

At this point, you have completed all of the required configuration steps. You are now ready to use DB2 Developer Workbench to code your stored procedures.

Creating an SQL procedure

To create an SQL procedure, follow these steps:

1. To begin coding a new procedure, right-click on the project you just created in the Data Project Explorer view and select the **New** → **Stored Procedure** task (see Figure 5).

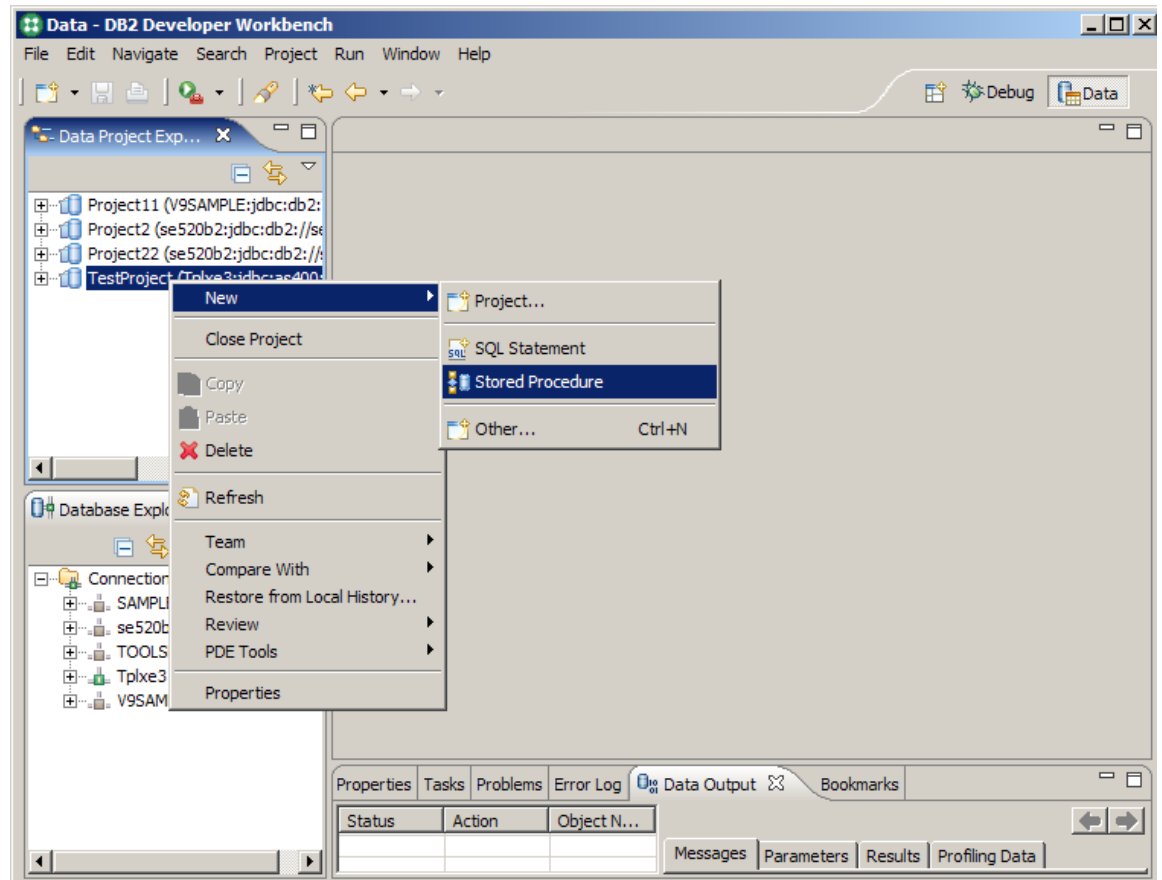


Figure 5. DB2 Developer Workbench window

- The New Stored Procedure window appears. To create either an SQL or Java stored procedure, select the **Language** pull-down menu (as shown in Figure 6), select the required stored procedure and click **Next**.

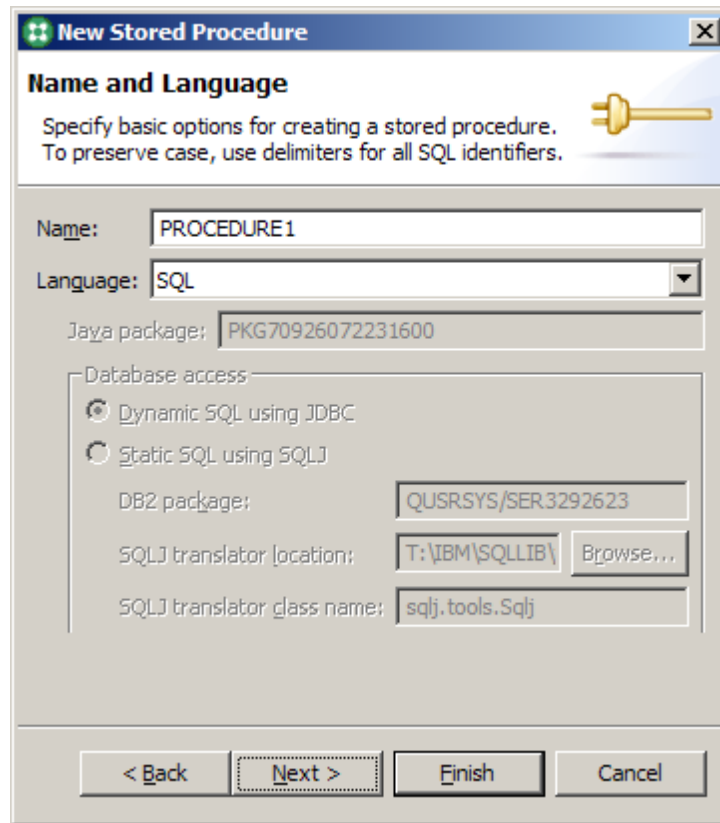


Figure 6. Selecting SQL as the stored procedure language type

- The SQL Statements window then appears (see Figure 7). To start the coding process, you can manually type the SQL statement (as Figure 7 also shows).

If you need help with the syntax of constructing a SELECT, INSERT, UPDATE or DELETE SQL statement, click **Create SQL**. This support walks you through the steps of constructing or changing an SQL statement to help ensure valid syntax. You also have the option of using DWB to create a procedure shell and then using the DWB editor to add your business logic to the SQL procedure.

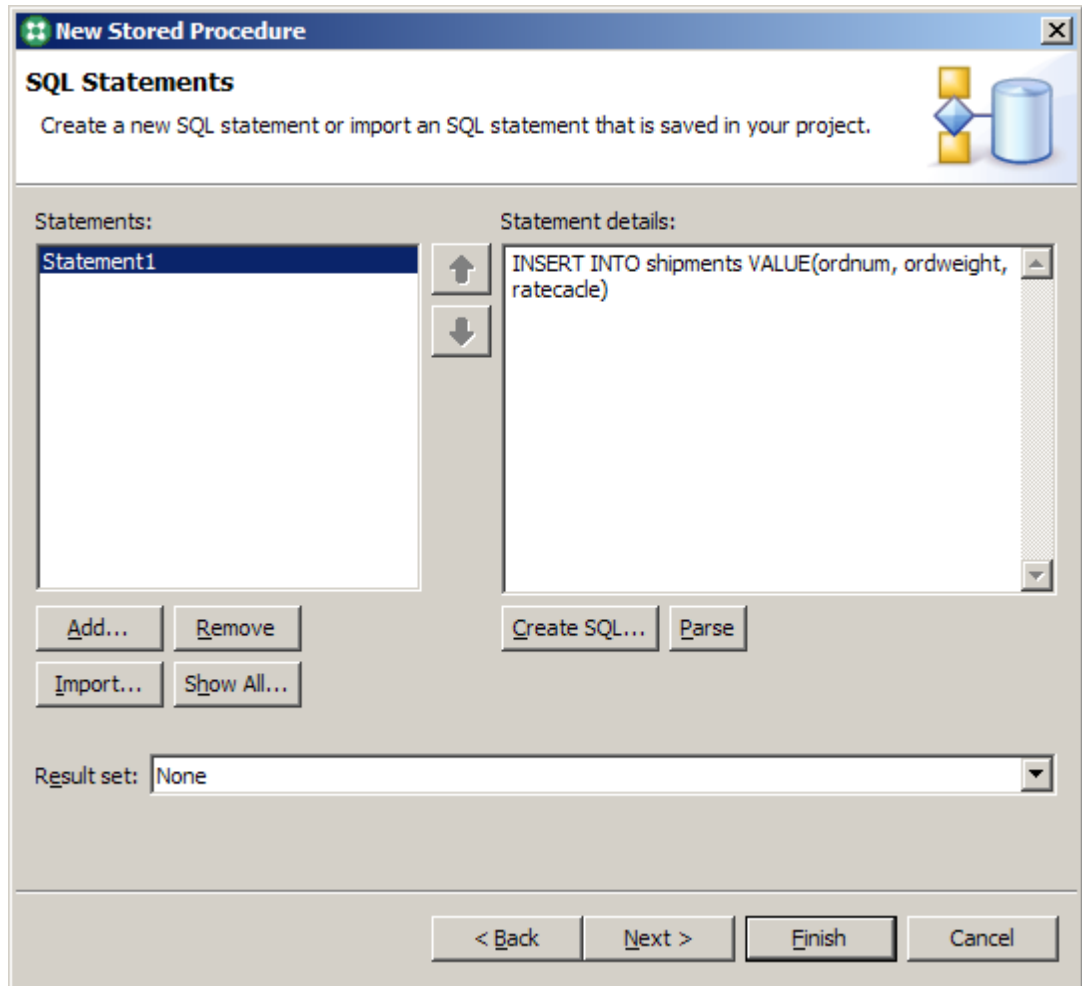


Figure 7. SQL Statements window for stored procedures

If your SQL stored procedure is going to return result sets, then select the proper value on the Result Set pull-down menu (not shown here). This example returns no result sets.

- When you have finished, click **Next** to advance to the Parameters window (see Figure 8).

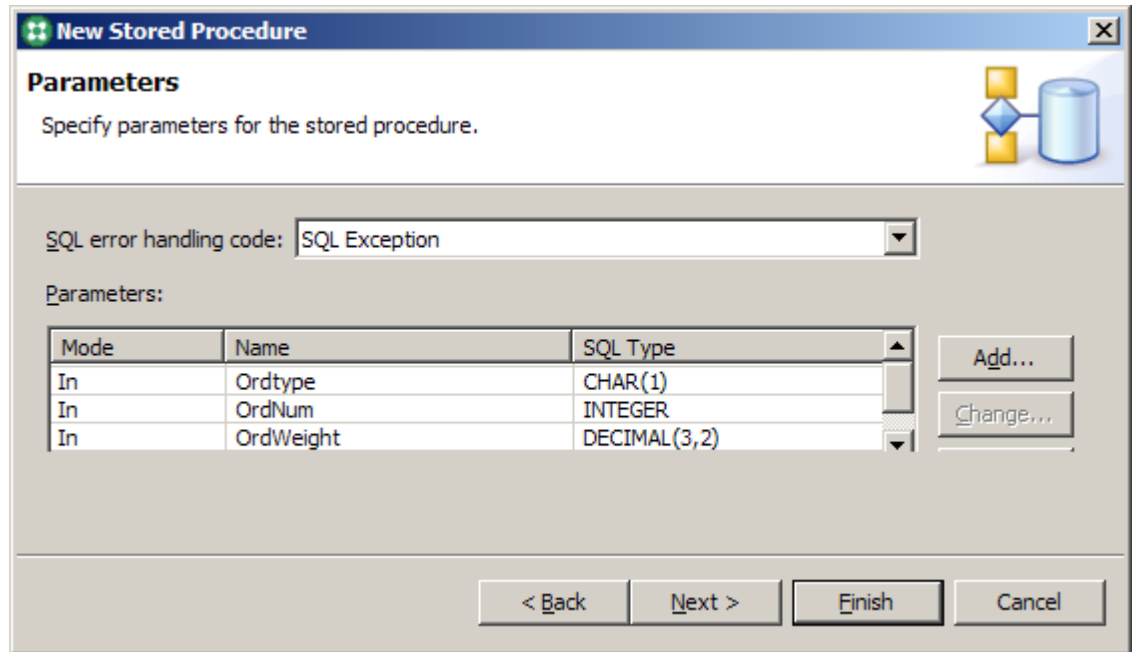


Figure 8. Parameters window for stored procedures

- Under the **Errors** setting in Figure 8, define the error-handling behavior. SQL Exception is the default value. This default value causes no error-handling code to be generated, and any SQL exception that occurs during execution is returned to the invoker. If you select the **SQLSTATE** or **SQLCODE** values, the wizard generates an exit handler to trap all SQL exception conditions and to copy the associated SQLSTATE or SQLCODE values into an output variable that the invoker can interrogate.
- DB2 Developer Workbench tries to generate some parameters automatically. However, if the parameters are not automatically generated, click **Add** to define explicitly the input and output parameters that your stored procedure requires. The example in Figure 8 contains the three input parameters that are necessary for this stored procedure example.
- After defining the parameters, click **Next**.

8. The Deploy Options window then appears (Figure 9). From this window, you can specify that the stored procedure be built (object created) on the system when the wizard has finished. You can also specify the debug mode. From the Deploy Options window, you can specify an optional, specific clause for the procedure. Because the SQL standards allow stored procedures to be overloaded (meaning you can have multiple procedures with the same name, provided that each version accepts a unique number of input parameters), you can use the specific clause to give that procedure an alternative name for unique identification.

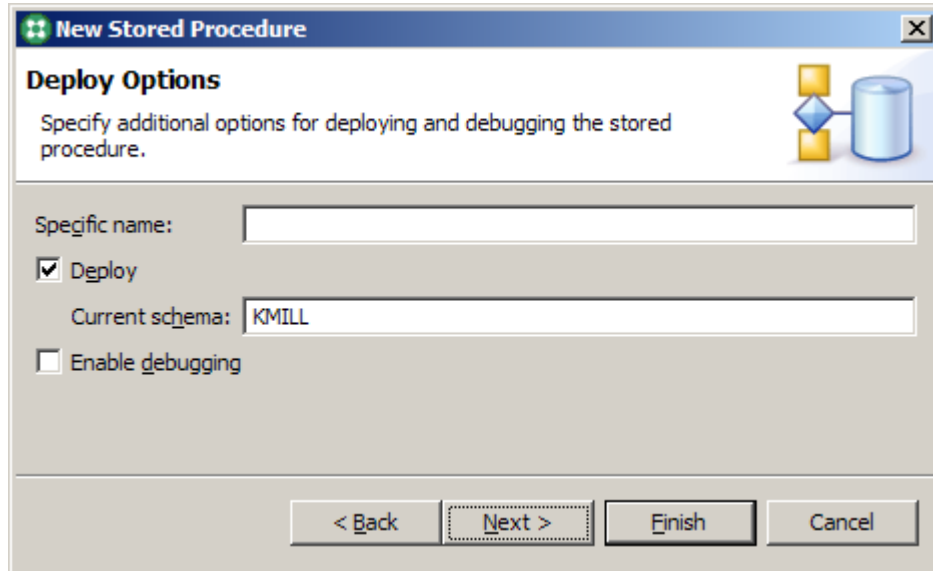


Figure 9. Deploy Options window for stored procedures

Furthermore, the specific clause has a unique value with DB2 for i5/OS. When an SQL procedure is created, DB2 generates a C program object that implements the business logic that is defined in your SQL procedure. DB2 generates a name for the C program object as part of the program-creation process. On DB2 for i5/OS, if the specific name is 10 characters or fewer, the specific name is used as the name of the i5/OS C program object.

Note: When using DB2 Development Center V8.1 to create SQL procedures on DB2 for Linux®, UNIX® and Windows operating-system-based servers, you must ensure that a C compiler is installed on the client workstation. However, starting with DB2 Development Center V8.2, this is no longer a requirement because the stored procedure is not converted to C code. This is also not necessary for DB2 for i5/OS, but for a different reason: the i5/OS operating system provides a C compiler. If deploying SQL procedures to IBM OS/400® V5R2, no additional products are necessary.

9. Click **Next**.

- As an option, you can add code fragments as demonstrated in Figure 10. These fragments are user-defined sections of source code or comments that you can insert at predefined locations in the generated source code. They are text files that you can create to facilitate code reuse. This is useful when you want to include a standard set of error-handling logic, variable declarations and so on, in multiple stored procedures.

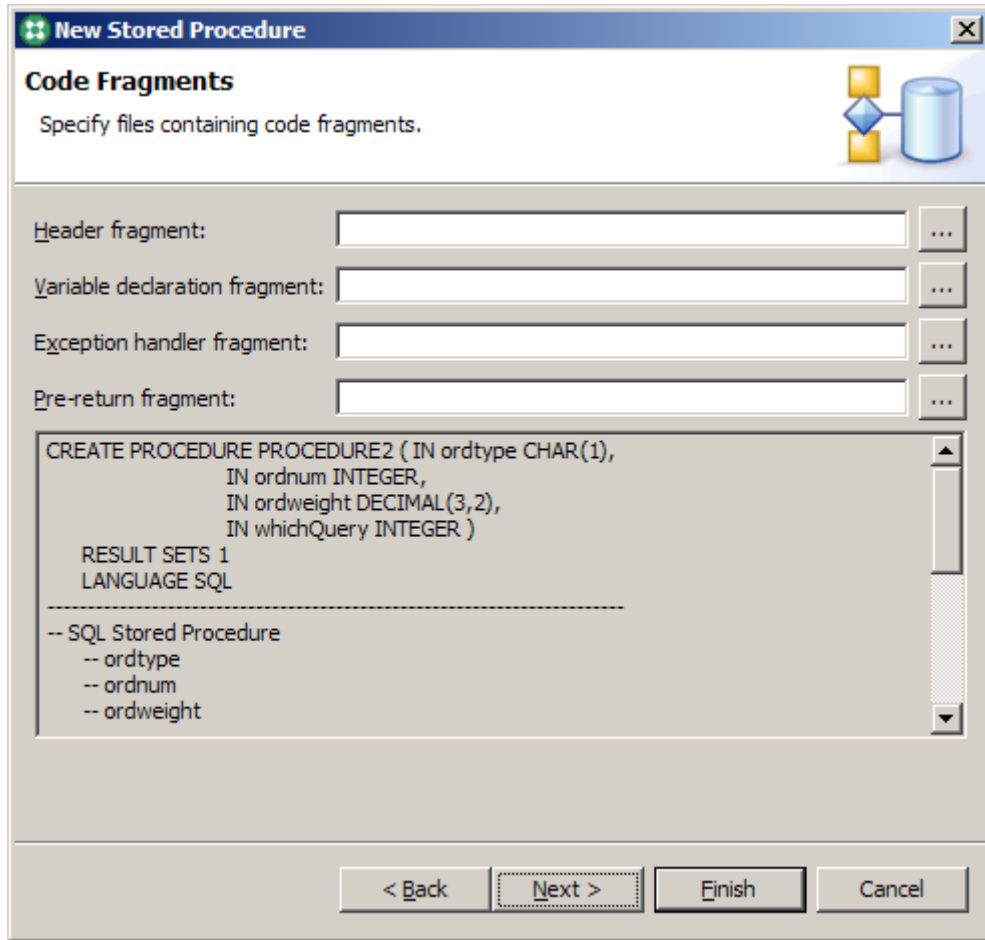


Figure 10. Code Fragments window for stored procedures

- After specifying code fragments (if any), click **Next**.

12. The **Summary** window (Figure 11) opens as the final step. The Summary window shows all the configuration options that you have specified to this point for the SQL procedure. Click **Show SQL** to display the body of the SQL stored procedure.

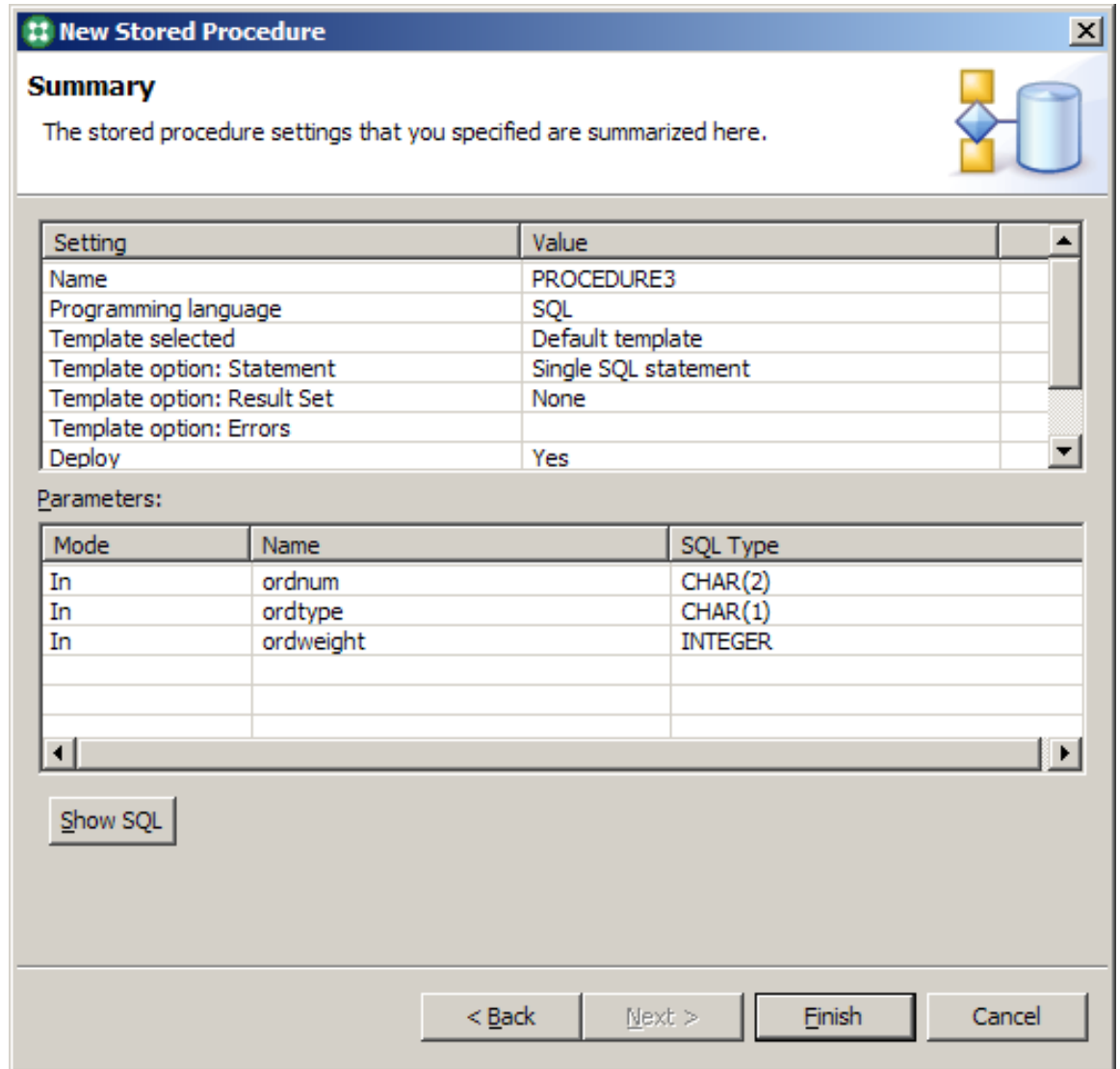


Figure 11. Summary window for stored procedures

13. Click **Finish** to complete the creation of the stored procedure. There are two possible outcomes:
 - If you selected the Deploy check box earlier on the Deploy Options window (Figure 9), then DB2 Developer Workbench attempts to create this stored procedure on the specified DB2 for i5/OS server. In the bottom-right corner of the DB2 Developer Workbench window (Figure 12), a message area shows the status of the stored procedure creation process.

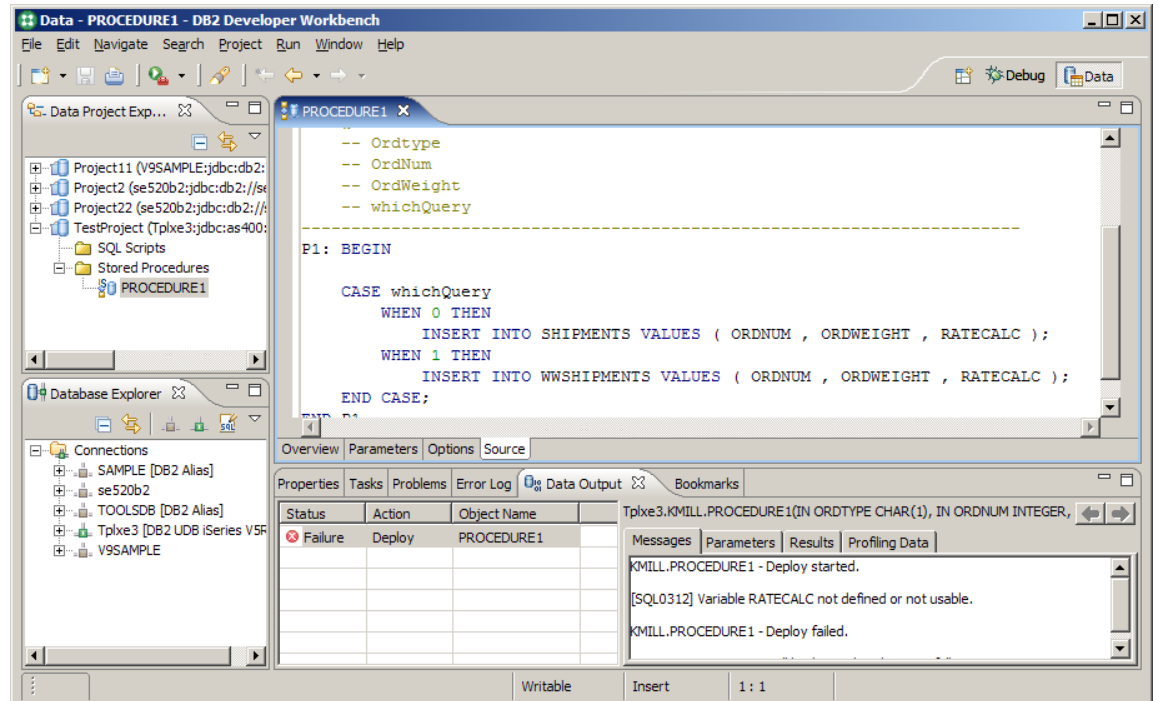


Figure 12. Developer Workbench window

- As you can see in the Messages area, the stored procedure creation failed in this case because the procedure shell generated by DB2 Developer Workbench does not contain any code that declares and assigns a value to RATECALC. Now that the error is identified, you can go into the Source Edit view (in the top-right corner of the DB2 Developer Workbench window) to make changes to the code.
- If you did not select the Deploy option, the stored procedure is presented in the Source Edit view without any interaction with the server.

Creating a Java procedure

In Version 8.2, DB2 Developer Workbench (through its predecessor DB2 Development Center) introduced the ability to create Java procedures. This feature generates stub Java code with the appropriate SQL statements; even if your Java skills are limited, you might find it quite useful. As is the case with SQL procedures, a wizard interface lets you enter the SQL statements, and the SQL Assistant attempts to define the parameters and generate the stub code. In addition, the wizard allows you to add multiple SQL statements and it also adds an argument to select one of them.

To create a new Java procedure, take the following steps:

1. From the Project View pane, right-click **Stored Procedures**. Then, select **New** → **Stored Procedure** (see Figure 13).

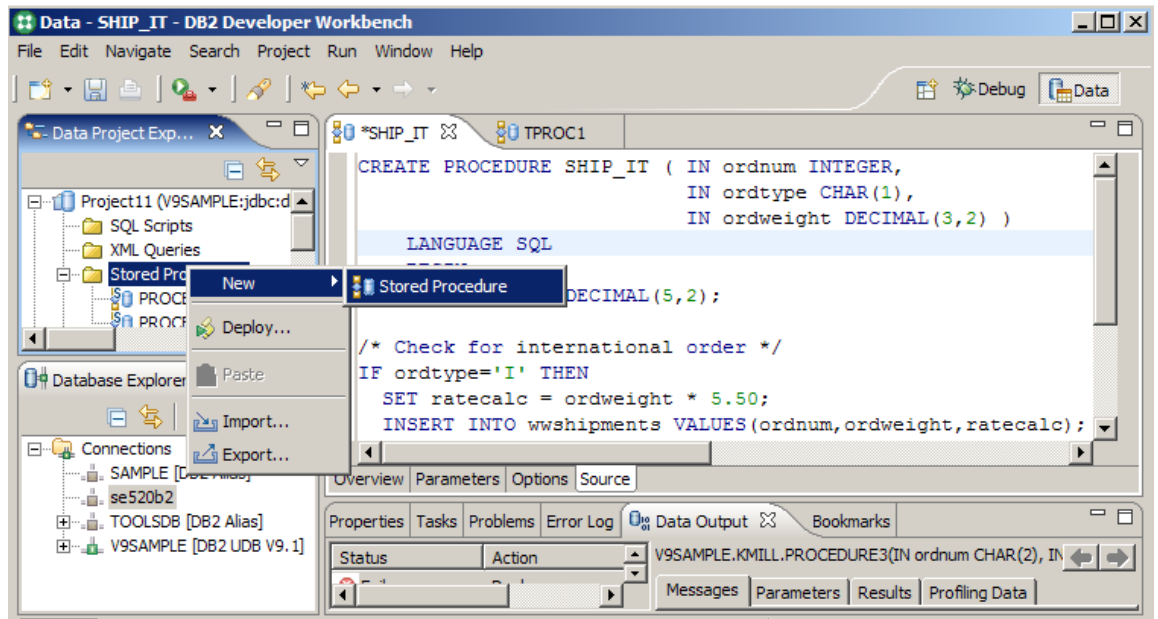


Figure 13. Creating a new Java procedure

- From the Name and Language dialog window, select **Java** as the procedure type (see Figure 14) and click **OK**. You probably want to specify your own Java package name, instead of using the cryptic package name that DB2 Developer Workbench generates by default. The Java package name is the basic grouping unit within a Java stored-procedures environment; it allows you to associate related stored procedures.

Identifying the Java database-access method is the final configuration option on this dialog. The majority of Java programmers who develop applications for the i5/OS operating system, use the JDBC access method. SQLJ has the same performance signature as JDBC — because the internal SQL plan caches are automatically employed by DB2 for i5/OS.

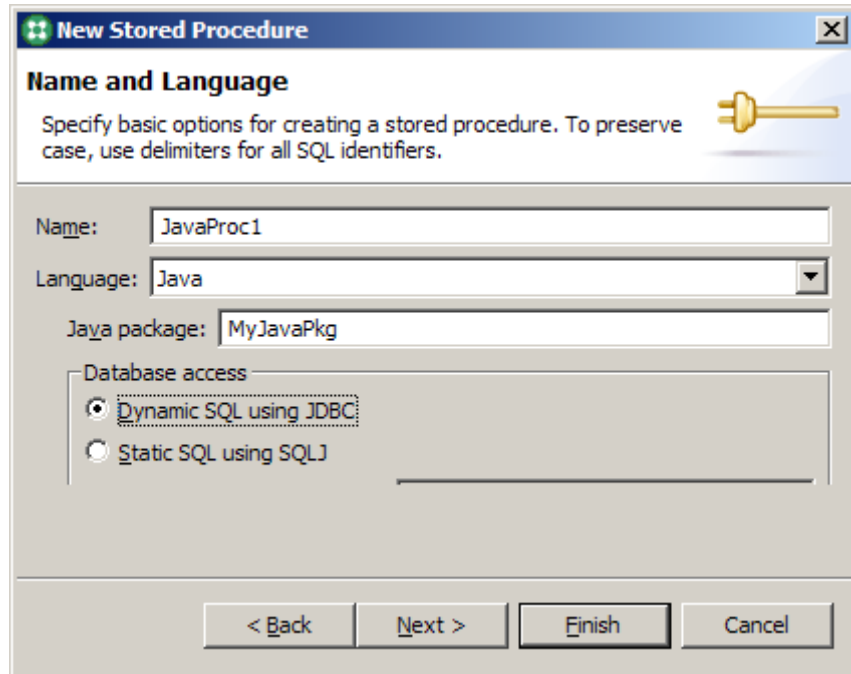


Figure 14. Selecting Java as the procedure type

- Click **Next** to advance to the SQL Statements dialog window. This dialog has the exact same options and behavior as they do for SQL procedures (see Figure 7).
- Click **Next** to enter the Parameters for the Java procedure.

The Parameters window is displayed, as shown in Figure 15. This window is different from the window used for SQL procedures because it contains a column for the Java type. As parameters are added, DB2 Developer Workbench automatically matches the SQL data type to the equivalent Java type.

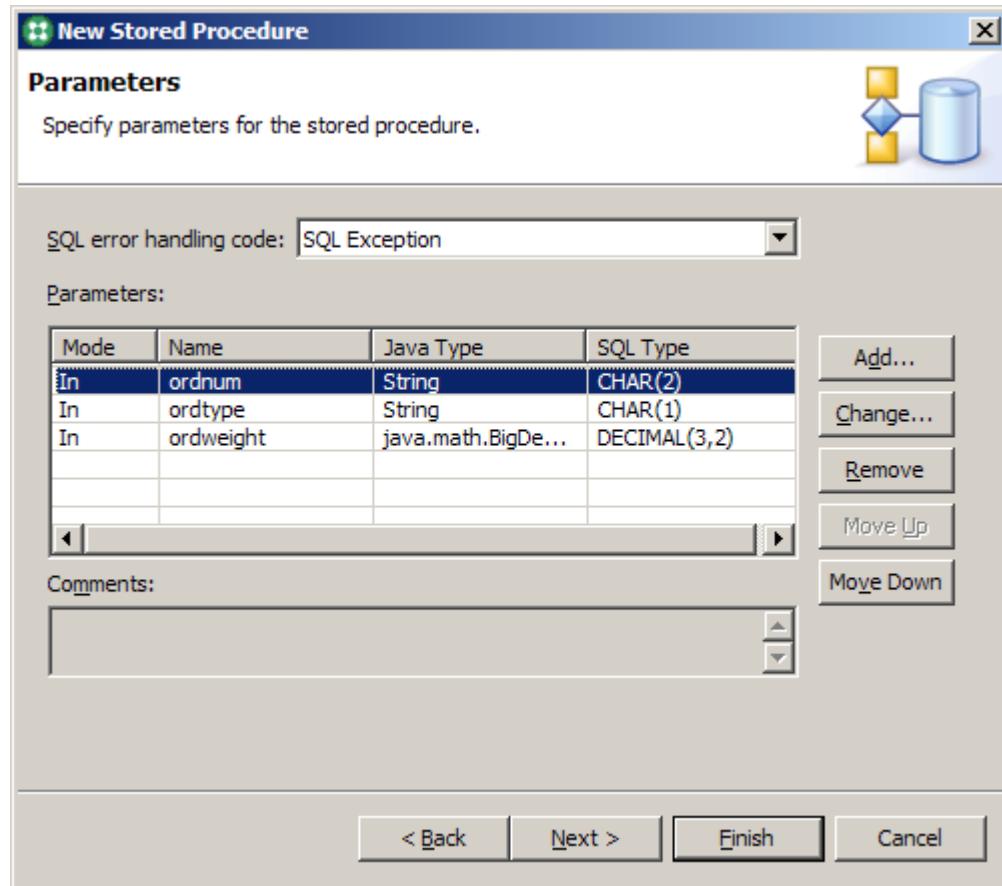


Figure 15. Parameters window for stored procedures

5. After specifying the parameters, click **Next**.

- The Deploy Options window is displayed (see Figure 16). From here, enter the **JAR ID**. One possible naming convention is to make the JAR identifier the same as the Java package name. However, there is no limitation that requires the JAR identifier to be the same as the package name.

You only need to provide the **Specific name** if overloaded Java stored procedures are being created. The Debugging and Deploy settings work the same as they do for SQL procedures.

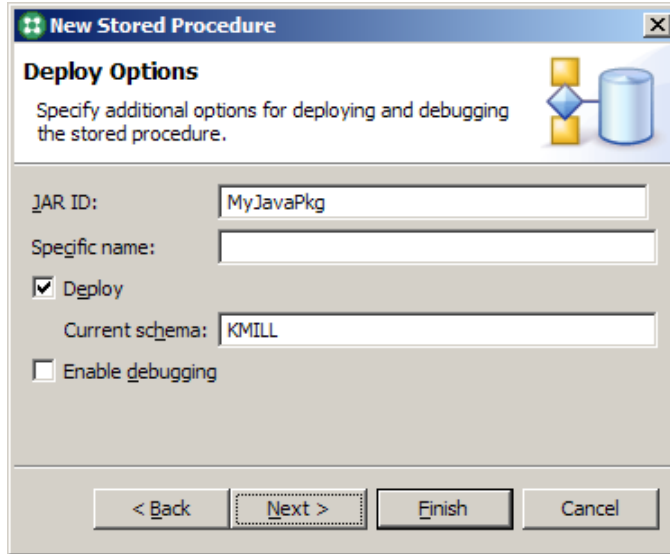


Figure 16. Deploy Options window for Java stored procedures

- Click **Next** to display the Code Fragments window. This window is not shown here; however, the Code Fragments dialog provides the same function as it did for SQL procedures (see Figure 10).

- After specifying the code fragments, click **Next** to display the Summary window (see Figure 17).

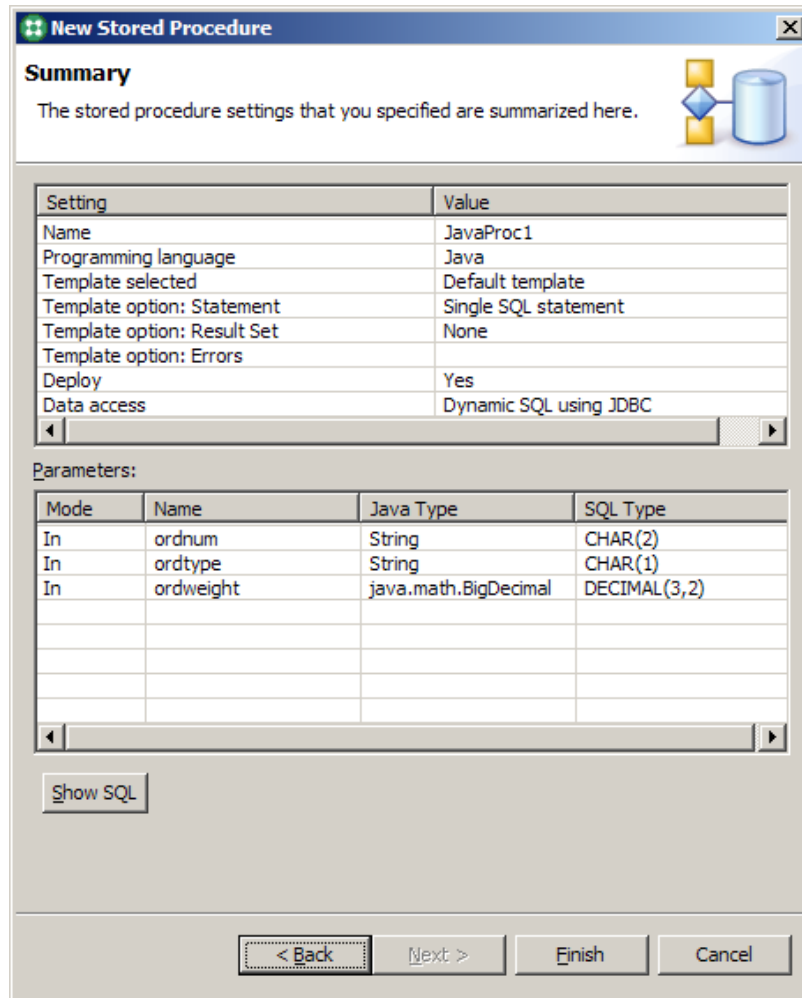


Figure 17. Summary window for Java stored procedures

- Verify the settings and parameters. If you are satisfied, click **Finish**. DB2 Developer Workbench generates a stub Java program with any specified SQL statements and also to display the generated Java code in the Source Edit view (as shown in the example in Figure 18).

If you specified the Deployment option, DB2 Developer Workbench attempts to create this Java procedure on the specified DB2 for i5/OS server. In the bottom-right corner of the Developer Workbench window (Figure 18), a message area shows the status of the creation process for the stored procedure.

If the Java stored procedure creation was successful, the JAR file for the Java stored procedure is placed in the following directory on the server:

/QIBM/UserData/OS400/SQLLib/Function

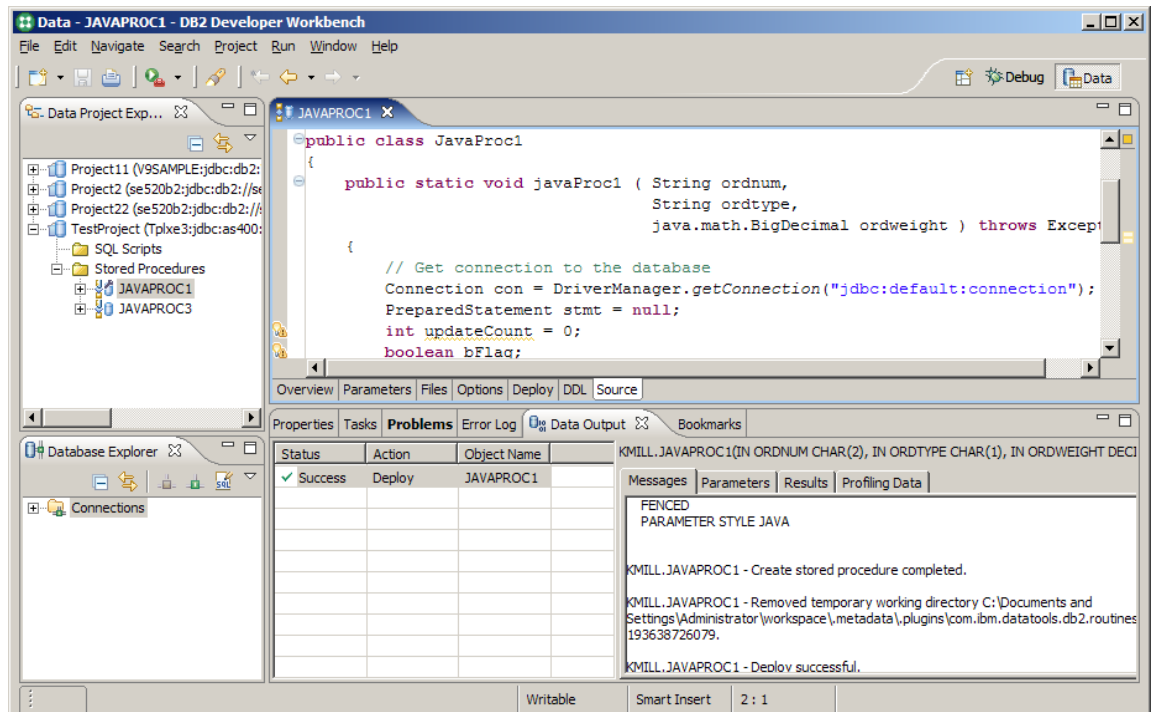


Figure 18. Stored procedure successfully built

At this point, you can use the visual editor to further modify the Java procedure code.

Importing procedures

In addition to creating new procedures, you can import existing procedures into your project. This allows you to organize all of your SQL and Java stored procedures into projects, which simplifies maintenance and deployment. You can import these objects from another database connection, an existing project or a file that contains the SQL procedure source code.

From the Data Project Explorer view, you can access an Import function — by right-clicking the **Stored Procedures** folder in the **Project** view. However, this Import function only supports procedure imports from an existing project or a file system object. Thus, you must use the Database Explorer for importing stored procedures from a DB2 for i5/OS server. You initiate the import process from the Database Explorer view in the bottom-left corner of the Data Perspective (see Figure 19).

1. Select the database connection for the server (that is, SE520B2) where the stored procedure resides.
2. Select the schema (or library) where the stored procedure is created (in this example, KKTEST).
3. With the schema now identified, you can select the procedure from the Stored Procedures folder view under the schema. (**Note:** The easiest way to import the selected stored procedure into DB2 Developer Workbench is to left-click and drag the procedure [follow the arrow] into the Stored Procedures folder that resides in the Data Project Explorer view.)

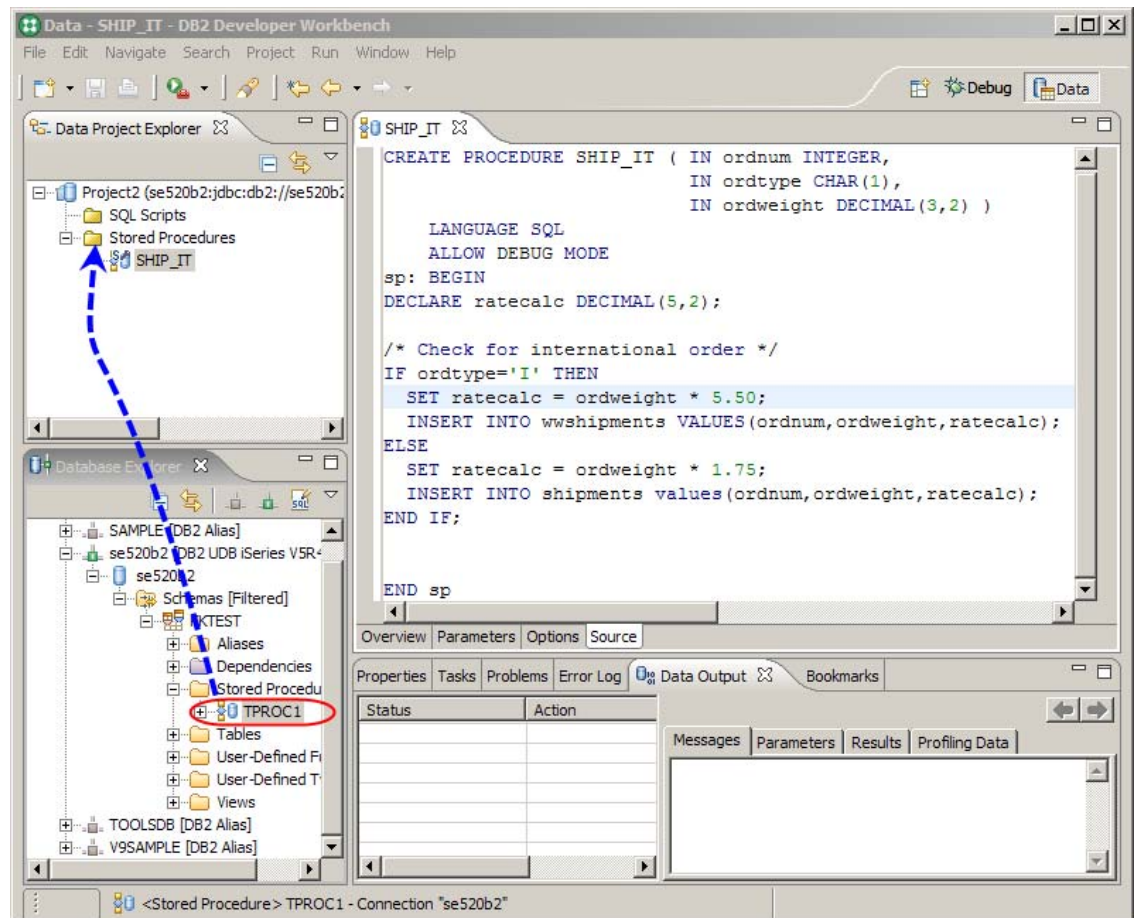


Figure 19. The import process

Running a procedure

After you successfully create or import the procedure, you can use DB2 Developer Workbench to run and test it. DB2 Developer Workbench also allows you to run any existing stored procedure on your System i model.

To run a stored procedure, follow these steps:

1. Right-click the procedure object and click **Run**.

If the stored procedure has input parameters, a window similar to the input-parameter window in Figure 20 is displayed to prompt you for the needed input values.

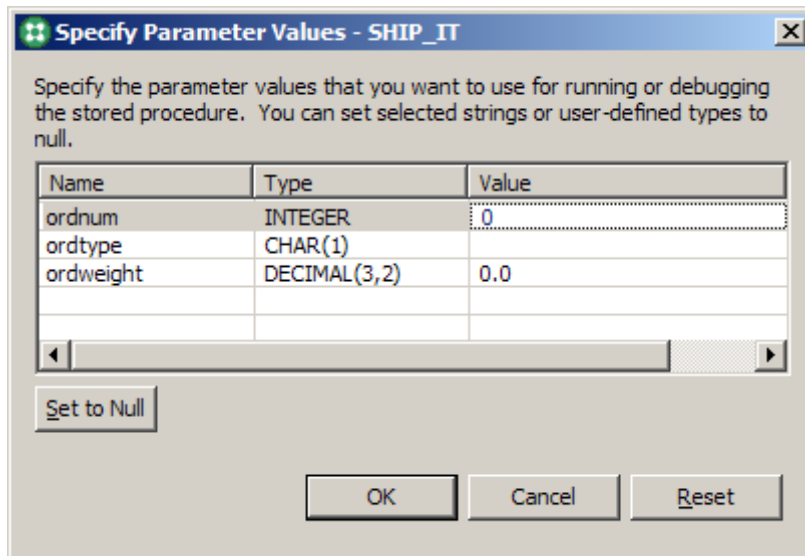


Figure 20. Specify Parameter Values window

2. Specify the input values and click **OK** to run the procedure.

Again, the status and output from running the stored procedure is shown in the bottom-right corner of the DB2 DWB window. On the **Parameters** and **Results** tabs, you can view any output parameters or result sets that are returned.

Repeating stored-procedure tests

DB2 Developer Workbench also allows you to specify run settings to simplify repeated testing of stored procedures. You can specify pre- and post-execution scripts, parameter values and other test options, such as rollback or commit after test execution. To input these settings, right-click the procedure object and click **Run Settings**. The specified settings persist as part of your project settings for later reuse; therefore, it is not necessary to retype test parameters and options every time you test the procedures.

Exporting and deploying procedures

DB2 Developer Workbench provides two methods for deploying stored procedures: *exporting to a deployment script* or *deploying directly to a target database*.

Exporting to a deployment script

The deployment option that involves exporting to a script file essentially puts your stored-procedure source code into a text file. This method is appropriate when your intent is to generate a script that you can later load into a change-management tool or iSeries Navigator. You can deploy stored procedures that are stored in the script file by running the script again. Depending on the target database, you might have to modify the exported script files.

1. Right-click a project's **Stored Procedures** folder (see Figure 21) and click **Export**.

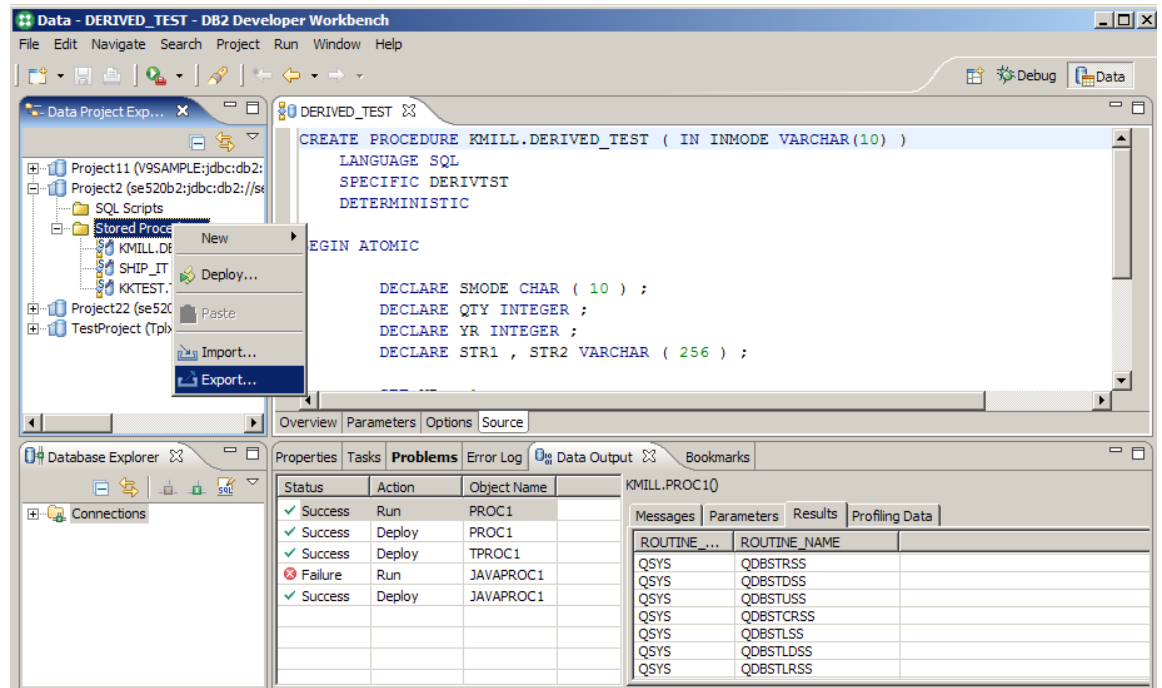


Figure 21. Exporting stored procedures

The **Export Selection** window is displayed (see Figure 22).

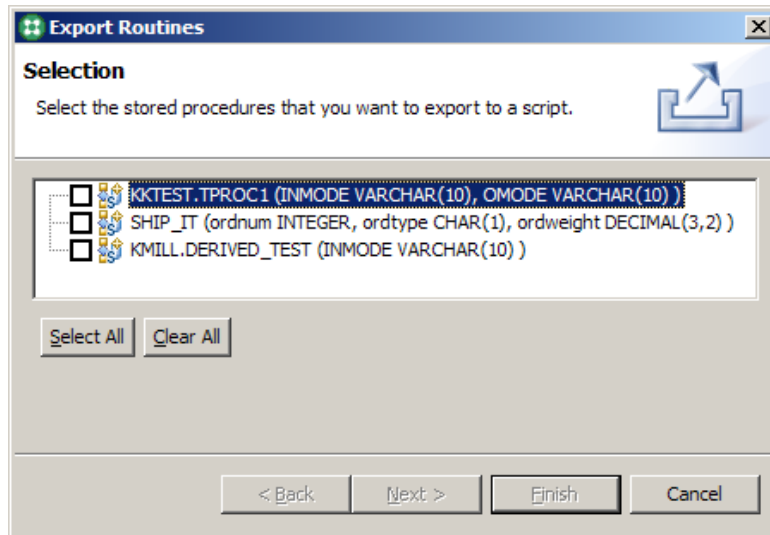


Figure 22. Export –Selection window

2. Select the procedures to export. When you have made all of the selections, click **Next**.

The **Export Target and Options** window appears (see Figure 23).

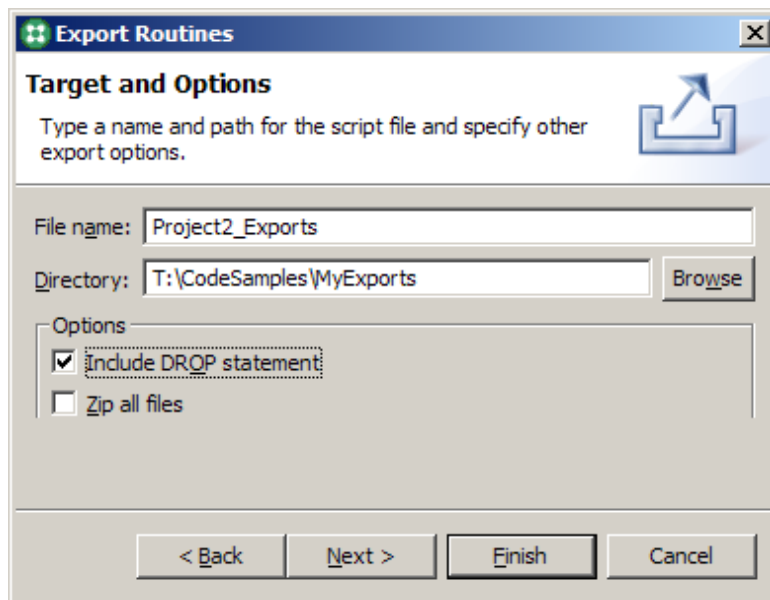


Figure 23: Export Target and Options window

3. Specify the path and file name of the export file. The **Include DROP statements** option lets DB2 Developer Workbench include DROP statements in the export file for each procedure. This provides users of the export file with a script that cleans up after itself by deleting old versions of the stored procedure before creating a stored procedure.
4. When you have all the appropriate options selected, click **Next**.
5. A Summary window lets you review your settings. Click **Finish** to complete the export request.

Deploying to target database

This option lets you deploy objects immediately to a target production database. This method is appropriate if you determine that an intermediate export step is unneeded and that the objects and production environment are ready for immediate deployment. This requires a connection to the server.

1. To deploy the project's stored procedures, right-click the project's **Stored Procedures** folder or right-click on an individual stored procedure.

The Deployment Options window is displayed (see Figure 24).

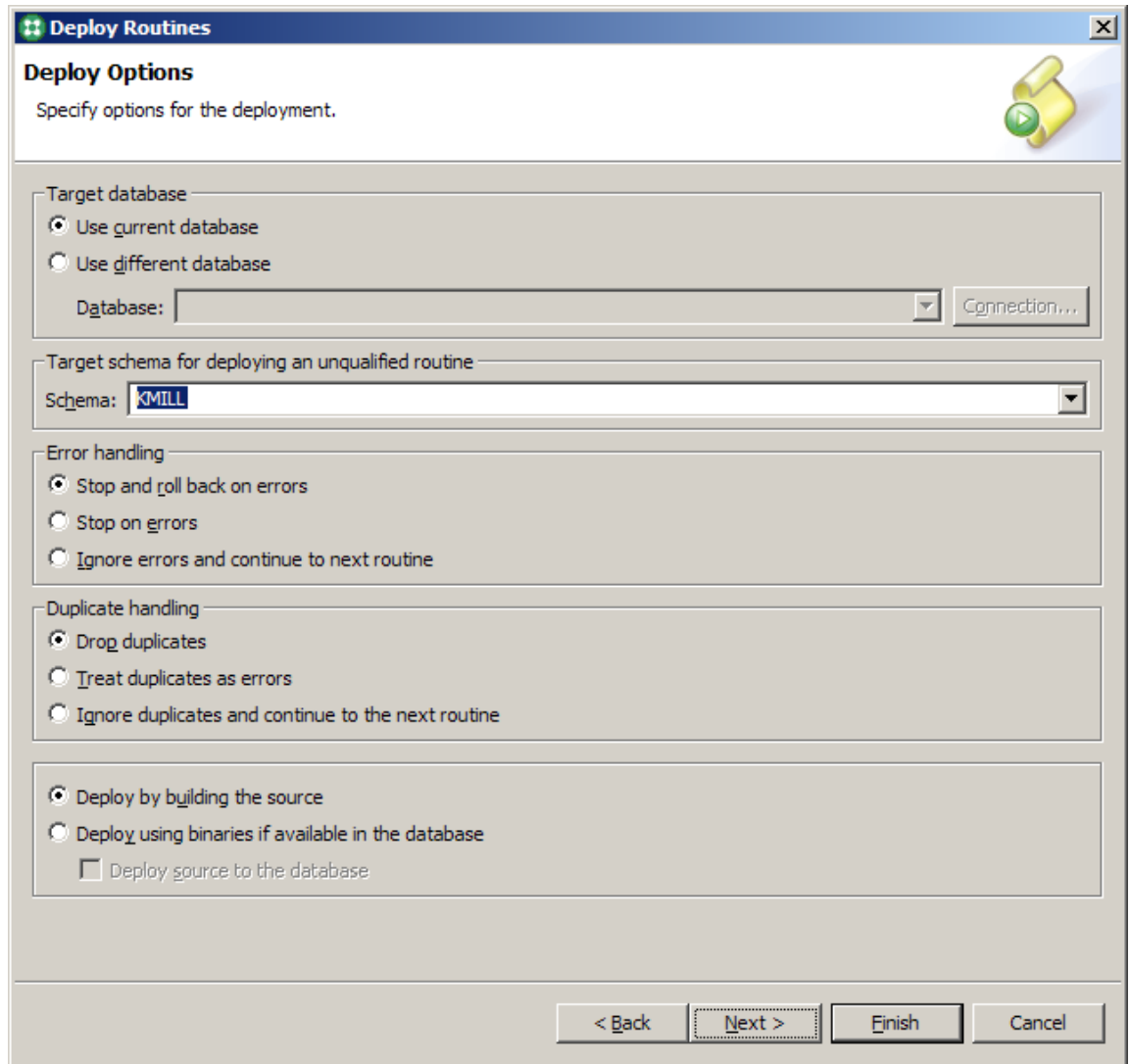


Figure 24. Deployment Options window

Here is a description of the deployment options that are available:

- The **Target database** input field enables you to deploy the selected procedures to a different DB2 for i5/OS server. Similarly, the Target Schema input fields allow you to control the schema where unqualified procedures will be created.
 - The **Error handling** inputs are only available when you have selected multiple stored procedures to deploy. If one of the stored procedure deployments fail, these options allow you to control what actions DB2 Developer Workbench takes with the other stored procedures.
 - The **Duplicate handling** options specify the actions that DB2 Developer Workbench should take when the stored procedure already exists on the specified server.
 - “Deploy by building the source option” is the only option that applies to DB2 for i5/OS stored procedures. There is no support for the binary deployment option.
2. After you have selected all of the appropriate deployment options, click **Next** to advance to the Routine Options window (Figure 25). The precompile and compile options are not available for DB2 for i5/OS, so the **Enable debugging** setting is the only option applicable to i5/OS stored procedures.

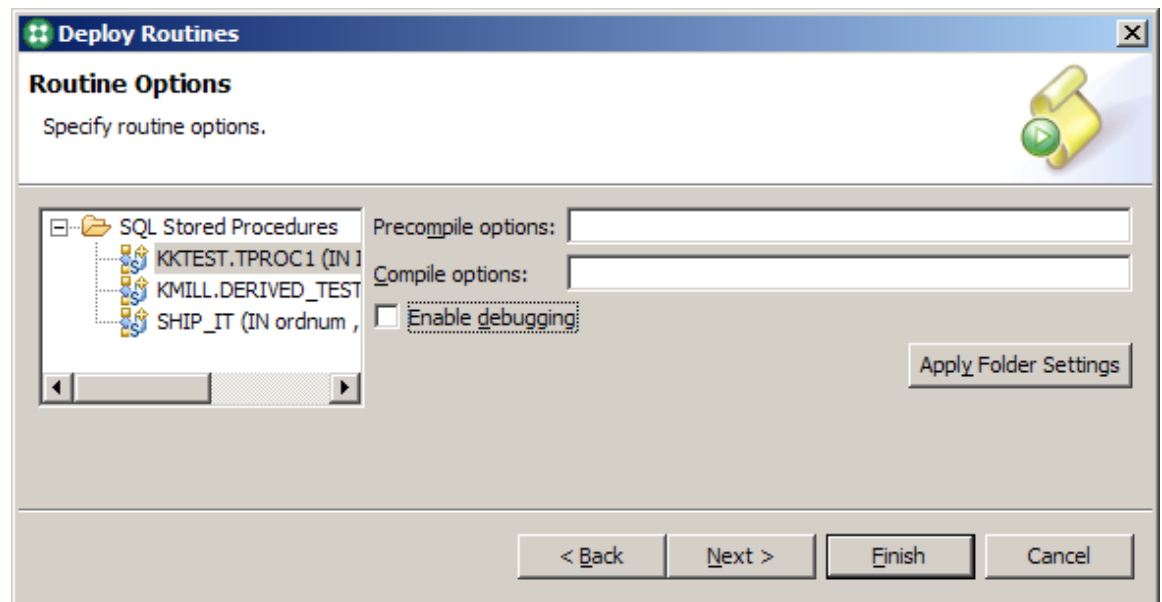


Figure 25. Routine Options window

3. At the Options window, click **Finish** to submit the deployment request.
4. Back at the DB2 Developer Workbench panel, check the status in the Data Output panel in the bottom right-hand corner to monitor the progress of the deployment request.



Summary

The intent of this white paper is to help you understand the DB2 Developer Workbench capabilities and how you can use them to create, test and deploy SQL and Java stored procedures on DB2 for i5/OS. This iterative development toolset provides an environment that simplifies your efforts to create and encapsulate your business logic inside SQL stored procedures.

Resources

These Web sites provide useful references to supplement the information contained in this document:

- IBM System i Information Center
<http://publib.boulder.ibm.com/series>
- IBM Publications Center
www.elink.ibm.com/public/applications/publications/cgibin/pbi.cgi?CTY=US
- IBM Redbooks™
ibm.com/redbooks
- i5/OS on IBM PartnerWorld®
ibm.com/partnerworld/i5os
- DB2 for i5/OS online manuals
ibm.com/series/db2/books.html
- DB2 Information Center
<http://publib.boulder.ibm.com/infocenter/db2help>
- IBM Toolbox for Java driver download site
www.iseries.ibm.com/toolbox
- IBM white paper: *Graphical debugging makes procedural SQL debugging on i5/OS even easier*
ibm.com/servers/enable/site/education/ibo/record.html?2ae2
- DB2 Developer Workbench download
ibm.com/software/data/db2/ad/dwb.html
- DB2 Connect Developer's Edition downloads:
 - Microsoft Windows client:
www14.software.ibm.com/webapp/iwm/web/preLogin.do?lang=en_us&source=swg-db2cdew
 - Linux client:
www14.software.ibm.com/webapp/iwm/web/preLogin.do?lang=en_us&source=swg-db2cdel
- JTOpen Web site
<http://jt400.sourceforge.net>

About the authors



Kent Milligan is a DB2 technology specialist in IBM ISV Business Strategy and Enablement. Kent spent many years of his IBM career as a member of the DB2 development group in Rochester, Minnesota. He speaks and writes regularly on various System i relational database topics.



Gene Cobb is a DB2 technology specialist in IBM ISV Business Strategy and Enablement. He has worked on IBM midrange systems since 1988, with 10 years in the IBM Client Technology Center (CTC), IBM Rochester. During his tenure at the CTC, he assisted IBM clients with application design and development using RPG, DB2 for i5/OS, IBM CallPath/400 and IBM Lotus® Domino® applications. His current responsibilities include providing consulting services to System i developers, with special emphasis in application and database modernization.



Trademarks and special notices

© Copyright IBM Corporation 2007. All Rights Reserved.

References in this document to IBM products or services do not imply that IBM intends to make them available in every country.

DB2, DB2 Connect, Domino, i5/OS, iSeries, IBM, the IBM logo, Lotus, OS/400, PartnerWorld, Redbooks and System i 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, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

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

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

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

Information is provided "AS IS" without warranty of any kind.

All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics may vary by customer.

Information concerning non-IBM products was obtained from a supplier of these products, published announcement material, or other publicly available sources and does not constitute an endorsement of such products by IBM. Sources for non-IBM list prices and performance numbers are taken from publicly available information, including vendor announcements and vendor worldwide homepages. IBM has not tested these products and cannot confirm the accuracy of performance, capability, or any other claims related to non-IBM products. Questions on the capability of non-IBM products should be addressed to the supplier of those products.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.