



IBM DB2 for i5/OS and Development Center

*Using DB2 Development Center to create, test, and deploy your IBM
System i SQL and Java stored procedures*

*Kent Milligan and Gene Cobb
ISV Business Strategy and Enablement
June 2006*



Table of contents

Abstract	1
Introduction	1
Adding a database connection using Configuration Assistant Wizard	2
Creating a project	8
Creating a new procedure	10
Creating an SQL procedure	10
Creating a Java procedure	16
Importing procedures	21
Editing procedures	29
Exporting and deploying procedures	31
Exporting to a deployment project.....	31
Exporting to a deployment script.....	35
Deploying to target database	36
Summary	38
Resources	39
About the authors	39
Trademarks and special notices	40



Abstract

This paper helps you understand the IBM DB2 Development Center capabilities that can be used in the creation, testing, and deployment of SQL stored procedures and user-defined functions on DB2 for i5/OS.

Introduction

IBM® DB2® Development Center is a rapid iterative, client-based development environment that supports the entire DB2 family; therefore, it is an especially useful tool if you are developing and deploying procedures on multiple DB2 server platforms. DB2 Development Center is the replacement product for the IBM DB2 Stored Procedure Builder product that was included in IBM DB2 Application Development client Version 7.

DB2 Development Center Version 8.2 supports the development of SQL and Java™ stored procedures when used with IBM 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 Development Center does not support the creation of these objects on the IBM System i™ platform. Although a debugger is not provided with DB2 Development Center, a graphical debugger is available for IBM System i stored procedures (for more information about the IBM Toolbox for Java iSeries System Debugger feature, see ibm.com/servers/enable/site/education/ibo/record.html?2ae2).

To obtain the DB2 Development Center, load the DB2 Application Development client, which is part of IBM DB2 Personal Edition and can be downloaded from www14.software.ibm.com/webapp/download/search.jsp?go=y&rs=db2udbpde.

Note: DB2 Development Center is also included in the IBM DB2 Connect Unlimited Edition for iSeries product offering.

Adding a database connection using Configuration Assistant Wizard

Prior to launching DB2 Development Center, create a database connection to the System i model that contains your database objects. You can do this using DB2 Configuration Assistant, which is another utility in the DB2 Toolset family. Follow these steps to add the database connection:

1. Launch the DB2 Configuration Assistant from the Microsoft® Windows® **start** menu by clicking **start**→**All Programs**→**IBM DB2**→**Set-up Tools**→**Configuration Assistant**.

The Configuration Assistant window appears.

2. From the **Selected** pull-down menu, click **Add Database Using Wizard** as shown in Figure 1.

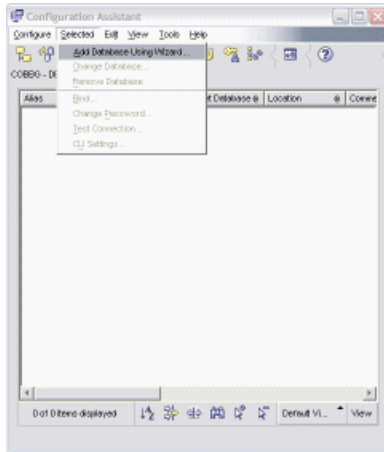


Figure 1. Configuration Assistant - Add Database Using Wizard... selection

3. From the **Select how you want to set up a connection** dialog window, click **Manually configure a connection to a database**.
4. The **Protocol** window is presented. As the communications protocol, select **TCP/IP**, and select **The database physically resides on a host or OS/400 system and Connect directly to the server**.

- The TCP/IP window is presented. In the **Host name** field, specify the name of the System i model as shown in Figure 2. Click **Next**.

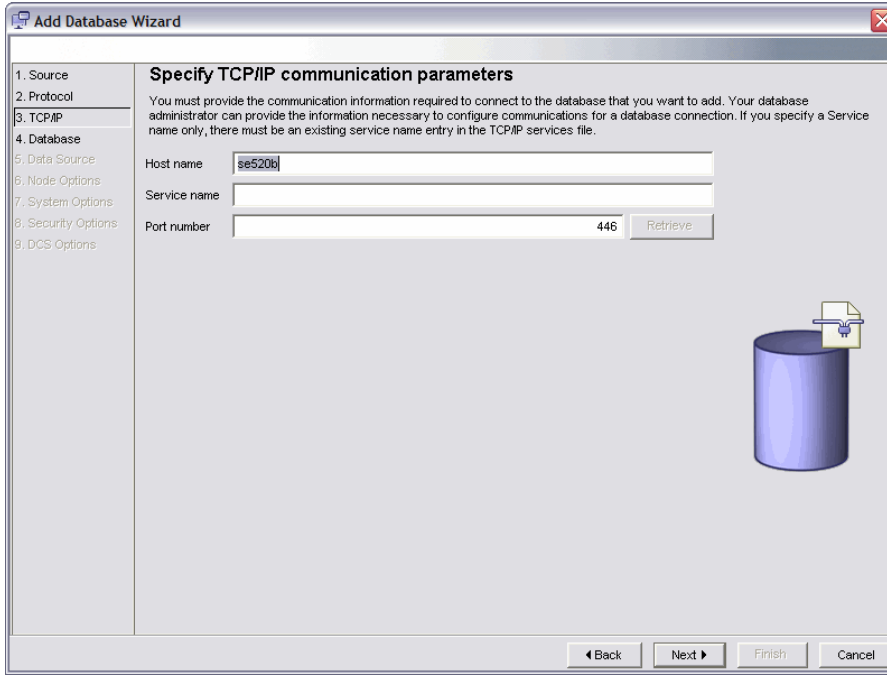


Figure 2. Specify TCP/IP communication parameters

- The Database window is presented. In the **Database name** field, specify the name of the database and the database alias (optional) as shown in Figure 3. Specify database name and alias. You can obtain the appropriate database name to specify by issuing the IBM i5/OS® WRKRDBDIRE command on the System i command line. Then, in the **Database alias** field, specify the database alias. Optionally, you can add a comment. Click **Next**.

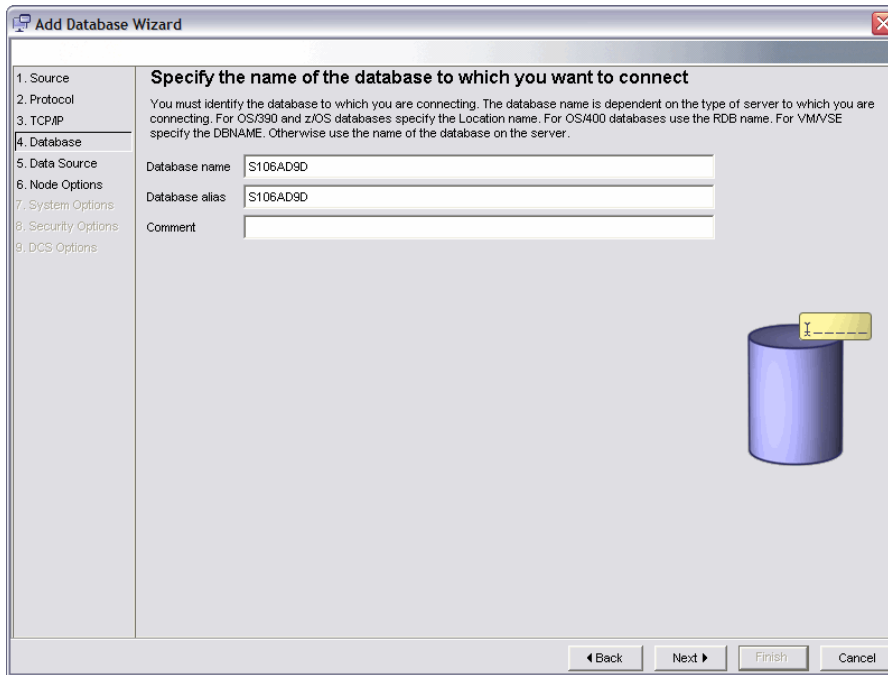


Figure 3. Specify database name and alias

- The Data Source window is presented. The database name is automatically populated in the **Data source name** field. Click **Next** to continue.

8. From the Specify the node options window, type OS/400 for the operating system. For the **instance name** parameter, specify the name of the System i model. Optionally, you can add a comment. An example is shown in Figure 4. Specify the node options. Click **Next**.

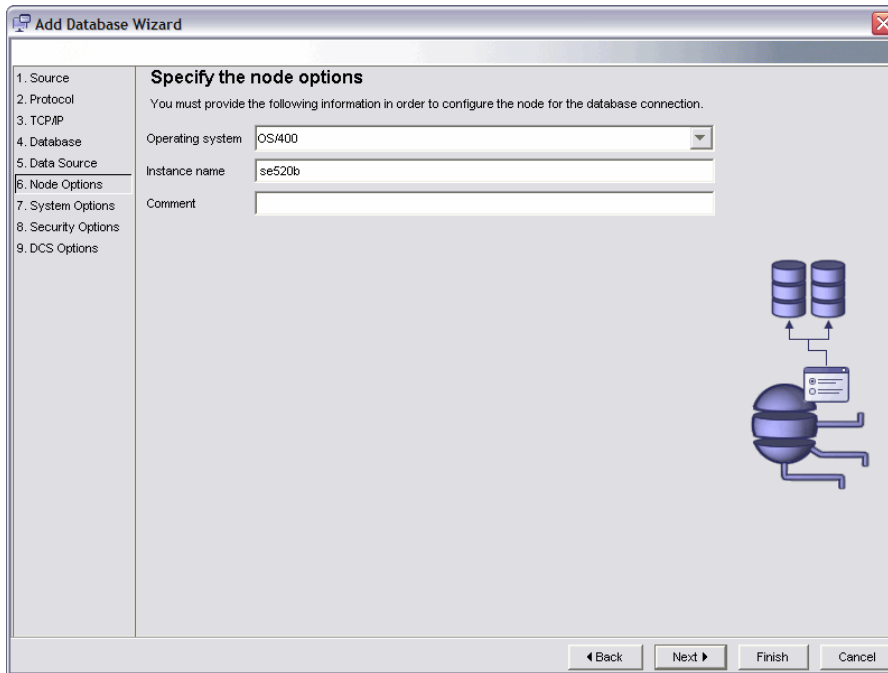


Figure 4. Specify the node options window

- The Specify the system options window is presented as shown in Figure 5. Specify system options. For both the **System name** and **Host name** fields, specify the name of the System i model. From the **Operating system** drop-down list, select **OS/400**. Optionally, you can add a comment. Click **Finish**.

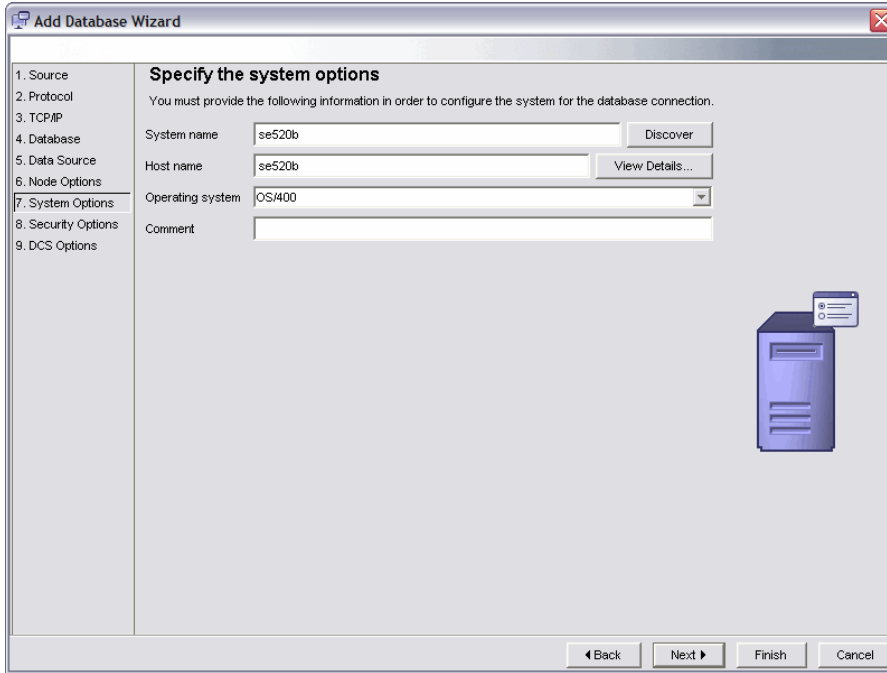


Figure 5. Specify system options window

- A window appears that confirms the successful addition of the configuration (see Figure 6. Database confirmed). Do **not** test the connection at this time. Exit the Add Database wizard by clicking **Close**.

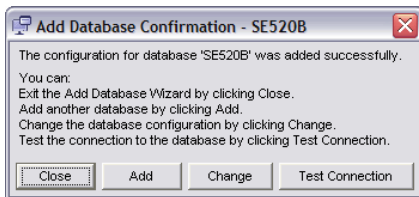


Figure 6. Database confirmed

The new database connection now appears in the **Configuration Assistants** list of connections (see Figure 7. Database connection added).

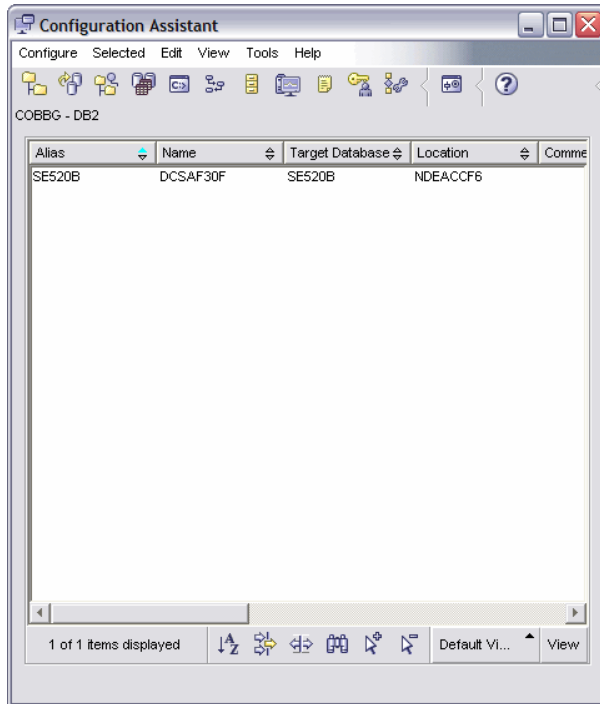


Figure 7. Database connection added

11. Close the **Configuration Assistant** window by clicking **Configure > Exit** from the pull-down menu.

Creating a project

DB2 Development Center 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 associated with a single application (for example, payroll application). To launch the DB2 Development Center and create a new project, take the following steps:

1. From the Windows **start** menu, click **All Programs**→**IBM DB2**→**Development Tools**→**Development Center**.

The Development Center Launchpad window appears (see Figure 8. Development Center Launchpad). From here, you can either create a new project or open an existing project.

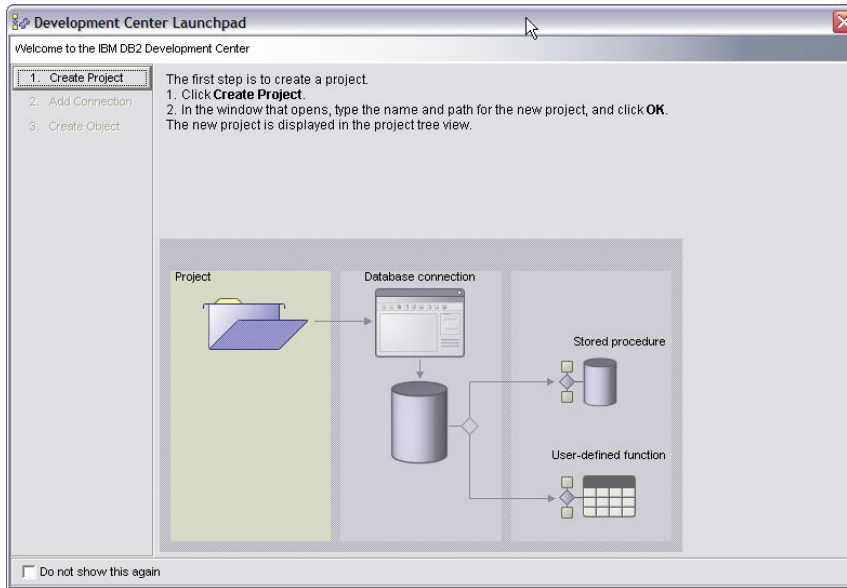


Figure 8. Development Center Launchpad window

2. Click **1. Create Project**.
3. From the Open Project window, specify the name of the new project and click **OK**.
4. After identifying the project to use, establish a database connection by clicking **Add Connection**.

Note: DB2 Development Center requires the use of an IBM DB2 Connect™ connection that uses Distributed Relational Database Architecture (DRDA) and a Java™ Database Connectivity (JDBC) connection that uses the IBM Toolbox for Java JDBC driver, which you can download from ibm.com/series/toolbox.

5. From the Connection Type window, select **Online** as the connection type. An online connection type means that your workstation has network connectivity to the System i model where your stored procedure is to be deployed. Click **Next**.

The **Connection** dialog window appears (see Figure 9. Connection).

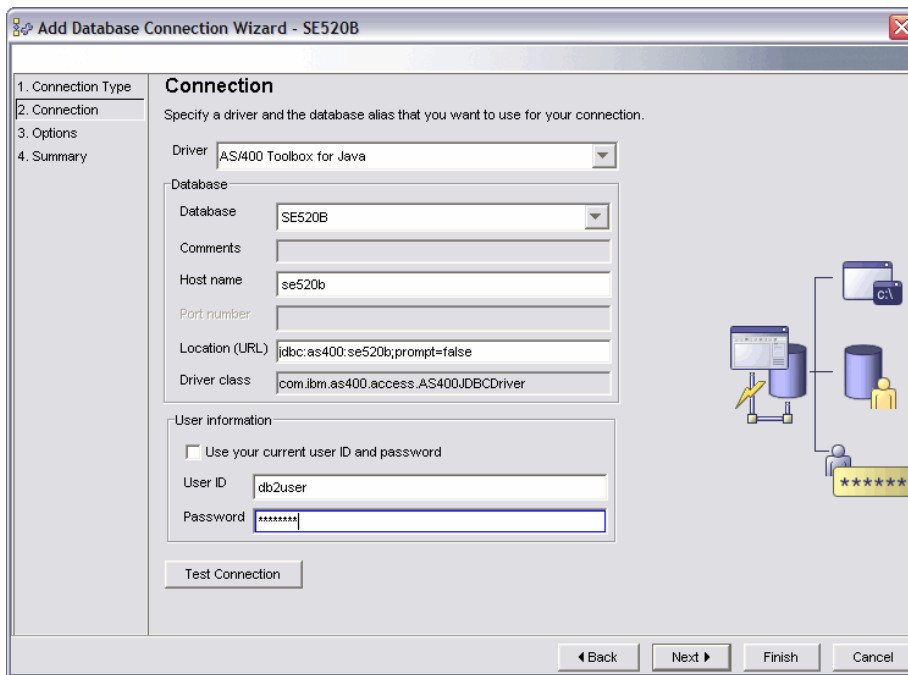


Figure 9. Connection window

6. Specify the following input:
 - On the **Driver** pull-down menu, you have two options:
 - If you need to use the DB2 Connect JDBC driver, click **IBM DB2 Alias**.
 - If you do not need to use the DB2 Connect JDBC driver, click **AS/400 Toolbox for Java**.
 - On the **Database** pull-down menu, choose the name of the DRDA connection for your System i model. Typically, this is the same as the TCP/IP host name and relational database name. (To see the local database name, use the Work with Relational Database Directory Entries [WRKRDBDIRE] CL command.)
 - Specify your System i user ID and password.
7. You can either verify that you have set up the DRDA connection correctly by clicking **Test Connection** or simply move on by clicking **Next**.

- From the Options window, specify the name of the schema (that is, the library name) in which you want the stored procedures created and click **Finish** to complete the connection definition step. Make sure that you leave the **Development Center Launchpad** open.

Creating a new procedure

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

Creating an SQL procedure

To create an SQL procedure, follow these steps:

- To begin coding a new procedure, click **Create object** on the **Development Center Launchpad**.

The New Object dialog window appears and gives you the option of creating either an SQL or Java stored procedure, as shown in Figure 10. Selecting SQL as the procedure.

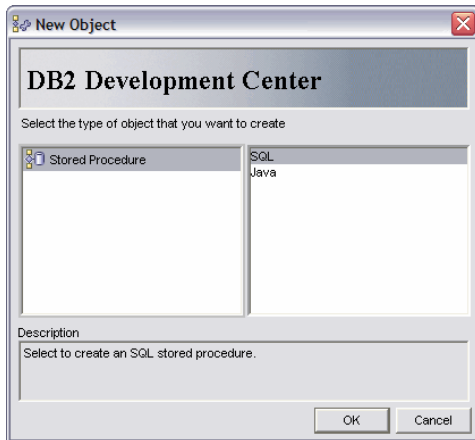


Figure 10. Selecting SQL as the procedure type

- To create a new SQL procedure, from the New Object dialog window, select the SQL procedure object type. The Create SQL Stored Procedure Wizard window appears. Specify the fully qualified procedure name and click **Next**. The Definition window appears (see Figure 11. Definition).

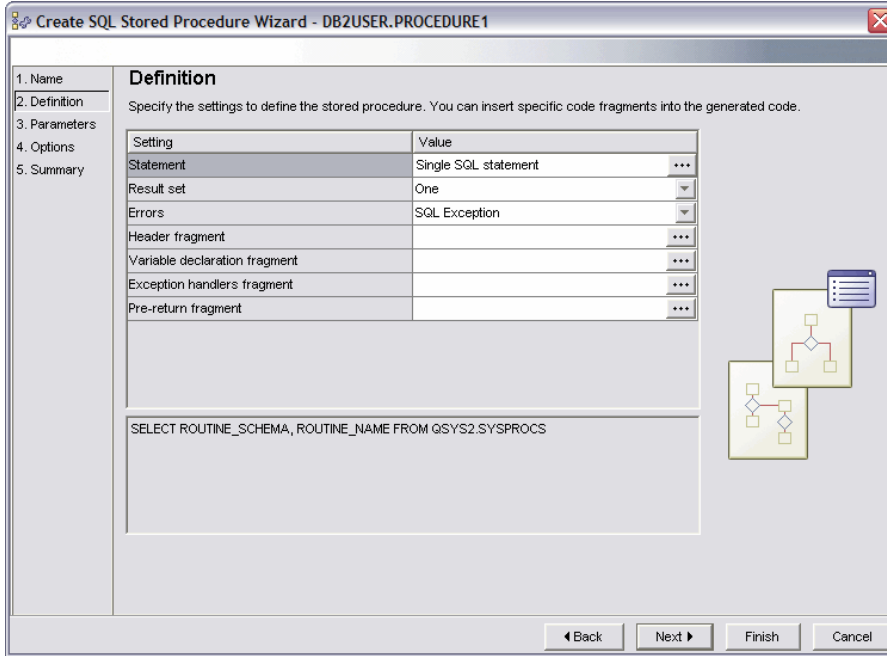


Figure 11. Definition window for stored procedures

- To start the coding process, click the button at the end of the statement value input field. The SQL Statement window is displayed (see Figure 12. SQL Statement window).

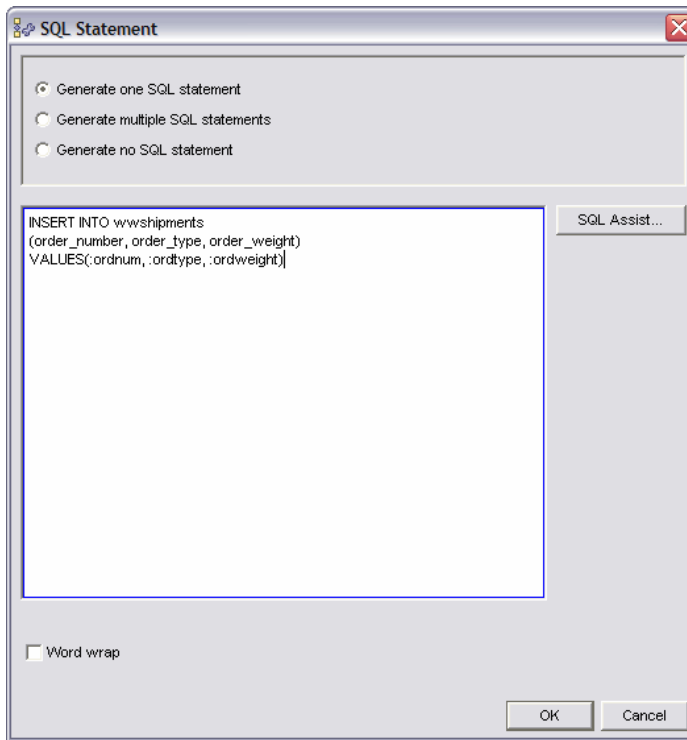


Figure 12. SQL Statement window

- If you need help with the syntax of constructing a SELECT, INSERT, UPDATE or DELETE SQL statement, click **SQL Assist** This support walks you through the steps of constructing or changing an SQL statement to help ensure valid syntax.

Note: The SQL statement wizard is also available in IBM iSeries™ Navigator starting in IBM OS/400® Version 5 Release 2.)

If you do not need assistance, you can type the SQL statements for your business logic directly into the edit window. Even though your code logic contains more than a single SQL statement, leave the **Generate one SQL Statement** option selected and code your business logic in SQL.

A BEGIN and END statement pair (that is, a compound statement) is not required because the procedure creation wizard generates the BEGIN and END statement as part of the code generation. Use the **Generate multiple SQL statements** option only when you need different SQL statements to run as part of a generated CASE statement.

5. When you have finished, click **OK**. The Definition window appears.
6. On the Definition window, indicate whether the stored procedure you are developing will return any result set by selecting **Yes** or **No** under the **Result set** setting.
7. Under the **Errors** setting, define the error-handling behavior. **SQL Exception** is the default value. This default value results in no error handling code being 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.
8. As an option, you can add code fragments. 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 all stored procedures. When you have finished, click **Next**.
9. The Parameters dialog window appears. SQL Assistant attempts to parse any variables that are prefixed with colons and automatically generates the parameters. In some cases, such as when complex statements are specified in the stored procedure, the SQL Assistant might not do this successfully.

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 13. Parameters window contains the three input parameters that are necessary for this stored procedure example.

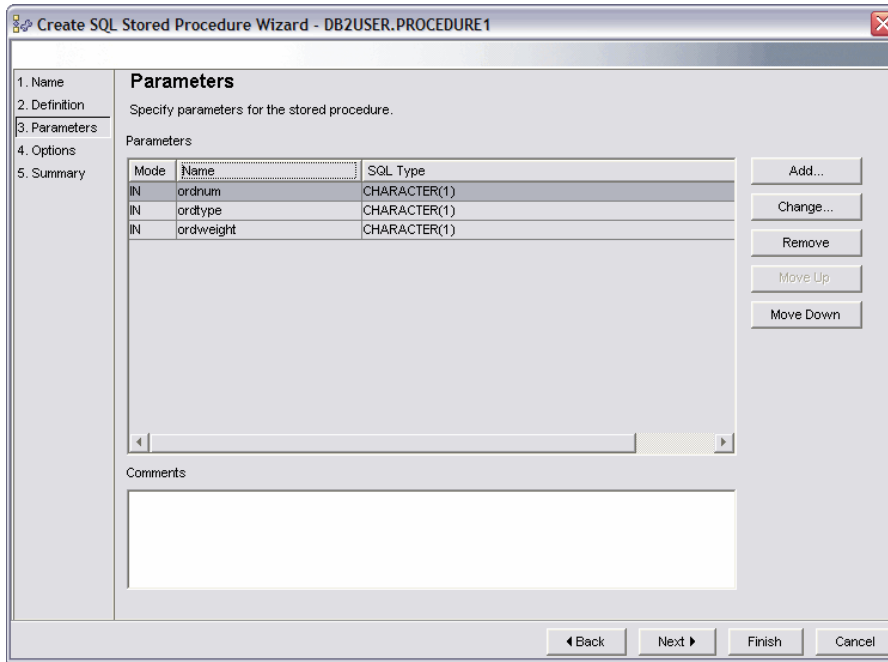


Figure 13. Parameters window

10. After you have defined the parameters, click **Next**.
11. The Options window is displayed. From this window, you can specify that the stored procedure be built (object created) on the system when the wizard has finished. From the **Options** window, you can also 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.

Furthermore, the specific clause has unique value with DB2 for i5/OS. When an SQL procedure is created, DB2 generates a C program object that implements the business logic 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 OS/400 C program object.

Note: When using DB2 Development Center V8.1 to create SQL procedures on DB2 for Linux®, UNIX®, and Windows servers, a C compiler must be 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: a C compiler is provided with the i5/OS operating system. If deploying SQL procedures to OS/400 V5R2, no additional products are needed. With OS/400 V5R1, the DB2 SQL Development and Query Manager product must be installed.

- Click **Finish** to initiate the stored-procedure build process and to display the DB2 Development Center project view (see Figure 14: Successful stored procedure build). In the bottom-right corner of this window, a message area shows the status of the stored-procedure creation process.

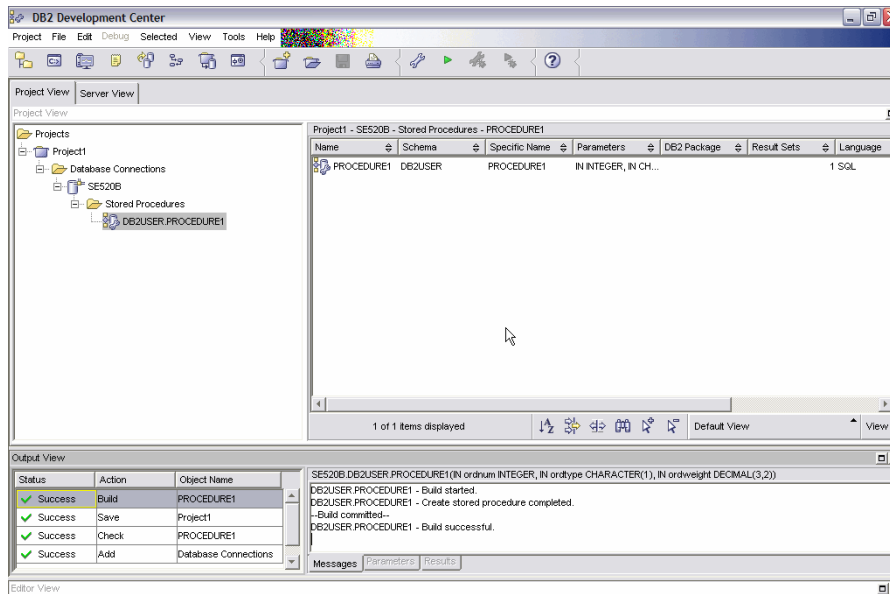


Figure 14: Successful stored procedure build

Creating a Java procedure

DB2 Development Center V8.2 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 is provided to 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 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 Using Wizard** (see Figure 15. *Creating a new Java procedure*).

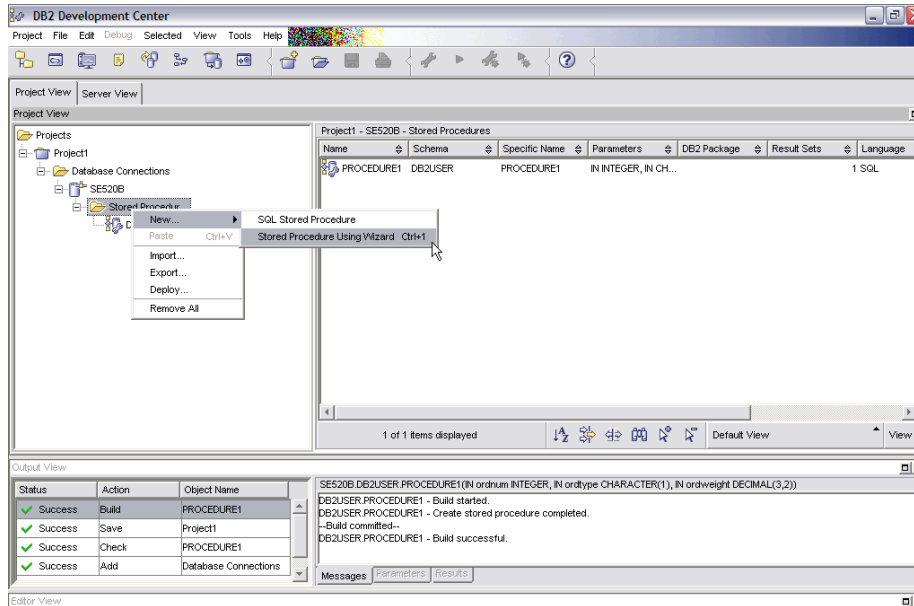


Figure 15. Creating a new Java procedure

- From the New Object dialog window, select **Java** as the procedure type (see Figure 16: *Selecting Java as the procedure type*) and click **OK**.



Figure 16: *Selecting Java as the procedure type*

The Create Java Stored Procedure wizard appears as shown in Figure 17: *Stored Procedure Name* window.

- Specify the fully qualified procedure name and click **Next**.

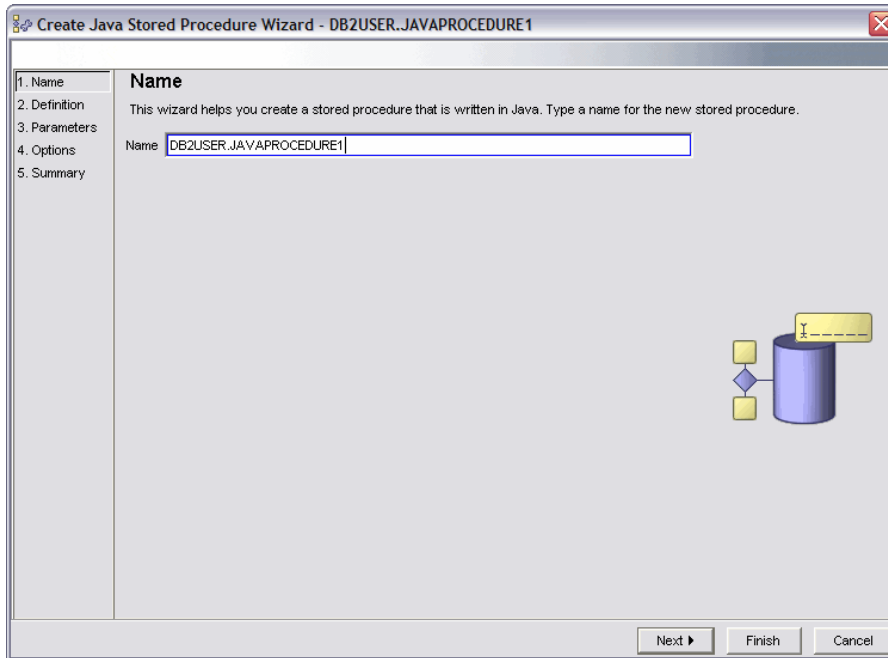



Figure 17: *Stored Procedure Name* window

The Definition window appears.

4. To start the coding process, click the  button at the end of the Statement value input field.

The SQL Statement window is displayed. Just as was the case with creating SQL procedures, you can enter SQL statements manually or use the SQL Assist wizard interface to help you construct the SQL statement.

5. If you do not want to enter the SQL statements manually, click **SQL Assist** To open the SQL Assist window (see Figure 18).

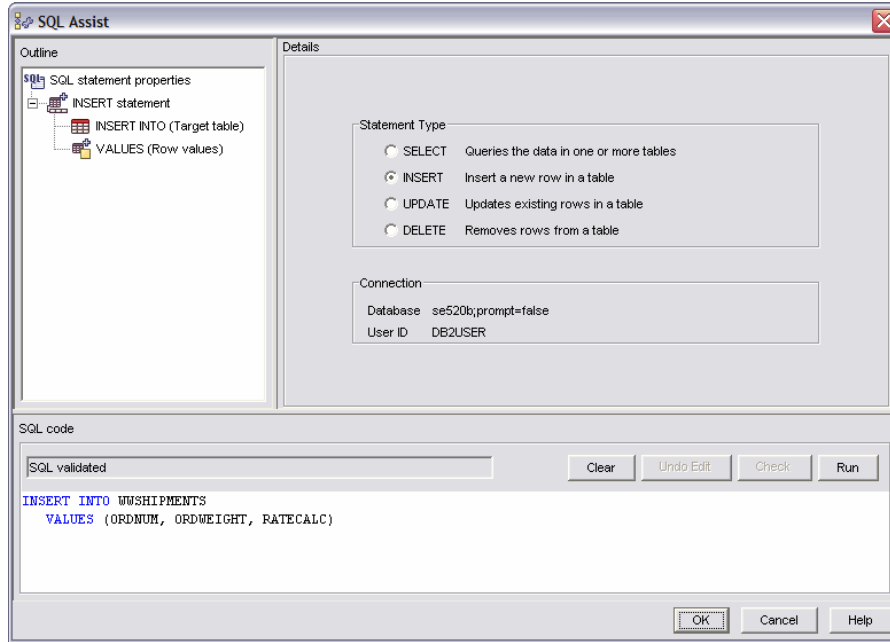


Figure 18: SQL Assist window

6. Use the wizard interface to help you build the SQL statement. When you have finished using the SQL Assist window, click **OK**.

The Definition window is displayed again.

- Click **Next** to enter the parameters.

The Parameters window is displayed as shown in Figure 19. Parameters window.

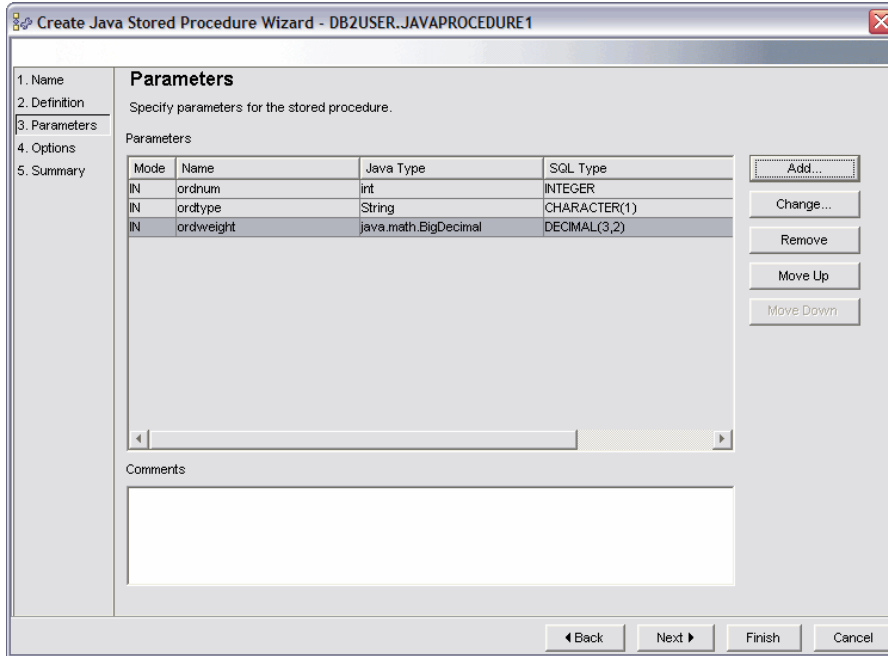


Figure 19. Parameters window for stored procedures

- This is the same interface that was used to define the SQL procedure parameters. When you have finished specifying the parameters, click **Next**.
- The Options window is displayed. From here, enter the specific Java program name. This is the 10-character short name given to the System i program object that will be created when the procedure build has been successfully completed.
- Click **Next** to display the Summary window.

11. Verify the settings and parameters. If you are satisfied, click **Finish**.

A stub Java program with the specified SQL statement is generated and the Java procedure is built, as shown in the example in Figure 20. Stored procedure successfully built.

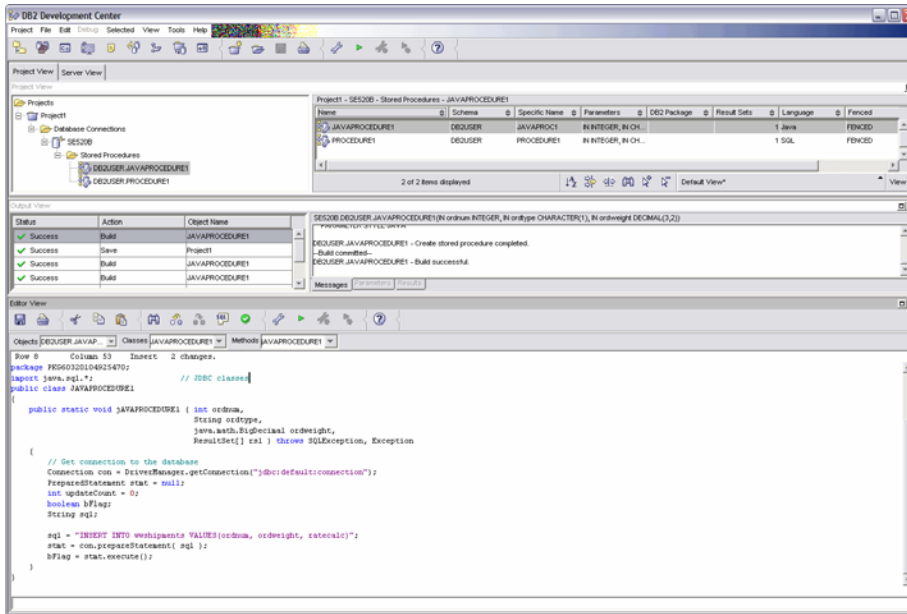


Figure 20. Stored procedure successfully built

At this point, you can use the visual editor to make further modifications to 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, simplifying 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. Launch the wizard by right-clicking the **Stored Procedures** folder in the **Project** view.

1. To import an existing procedure using the supplied wizard, from the Projects View pane, right-click **Stored Procedures**→**Import**, as shown in Figure 21. Import stored procedure.

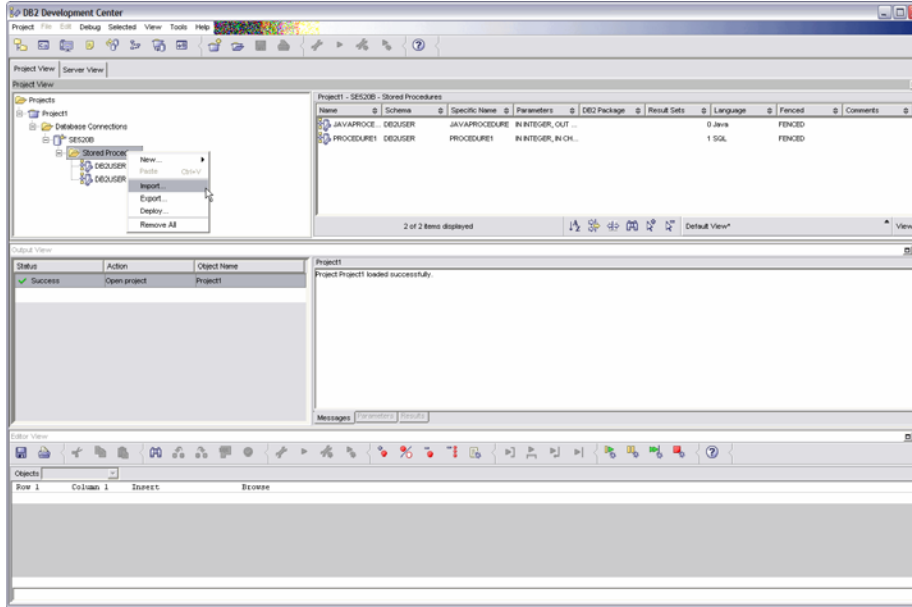


Figure 21. Import stored procedure

- At the Import dialog window (see Figure 22. Import), select **Database** as the type of source that you need to import and click **OK**.



Figure 22. Import window

- At the Import Wizard dialog window (see Figure 23), select **AS/400 Toolbox for Java** as the Driver, then type the database name, and specify the System i user ID and password. Click **Next**.

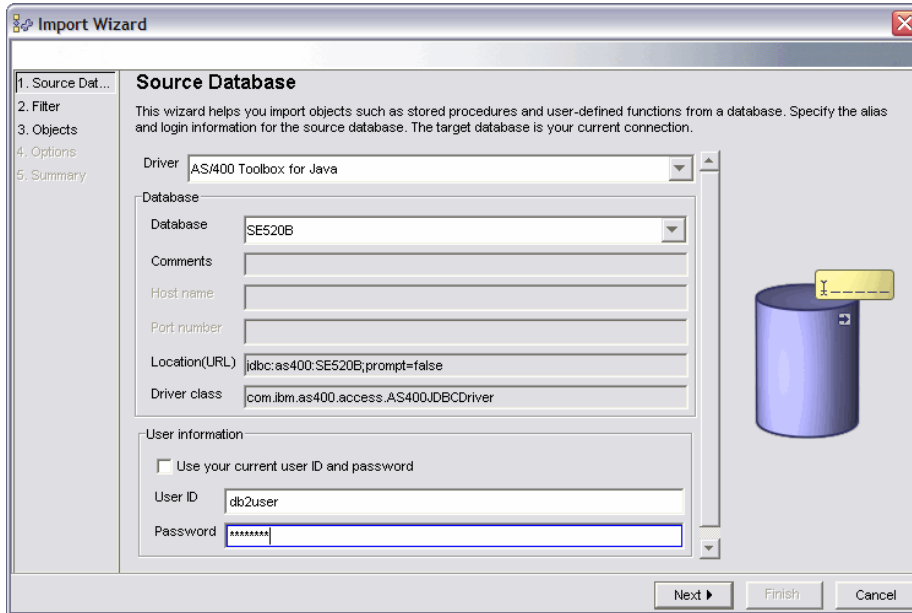


Figure 23: Import Source Database window

- If there are many stored procedures in the database, the **Filter** dialog window (see Figure 24) provides a means of filtering to reduce the number of procedures that are available for selection.

If you wish to use this feature, select **Filter using specified criteria** and specify the criteria. Click **Next**.

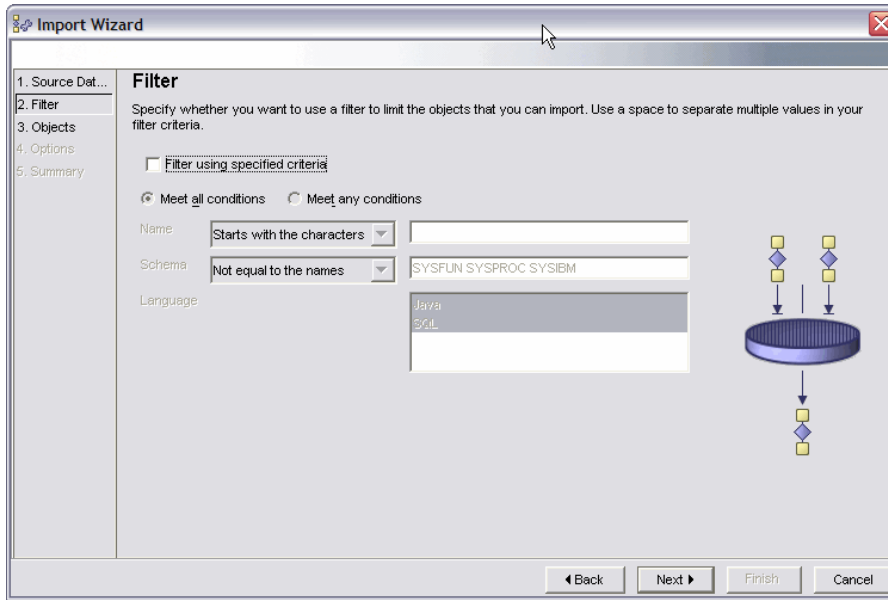


Figure 24. Import Filter window

The Objects window is presented and provides a list of procedures that match the filter criteria (see Figure 25. Import Objects window).

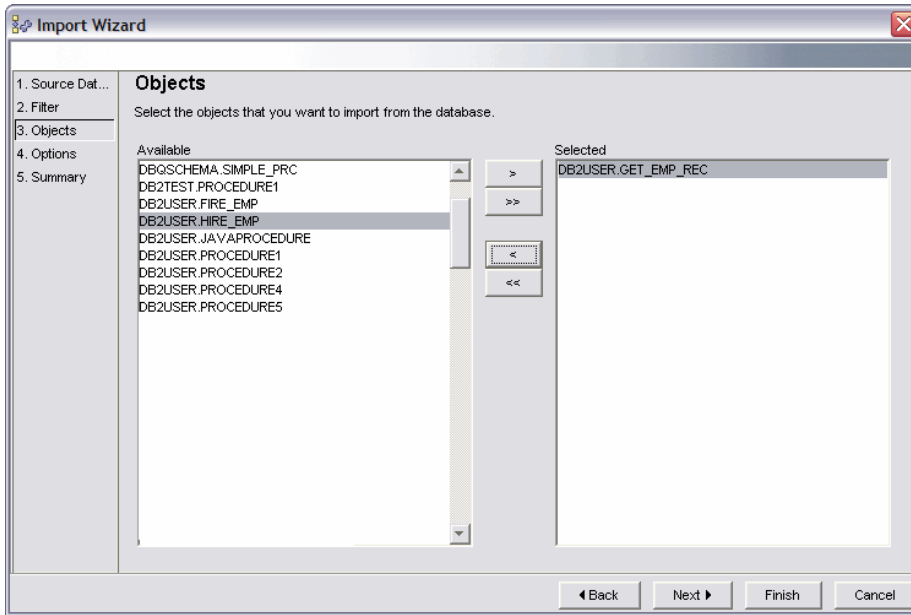



Figure 25. Import Objects window

- From the **Available** pane, select one or more procedures to import and click the  button to move them over to the **Selected** pane. Click **Next**.

The Options window is presented (see Figure 26. Import Options window).

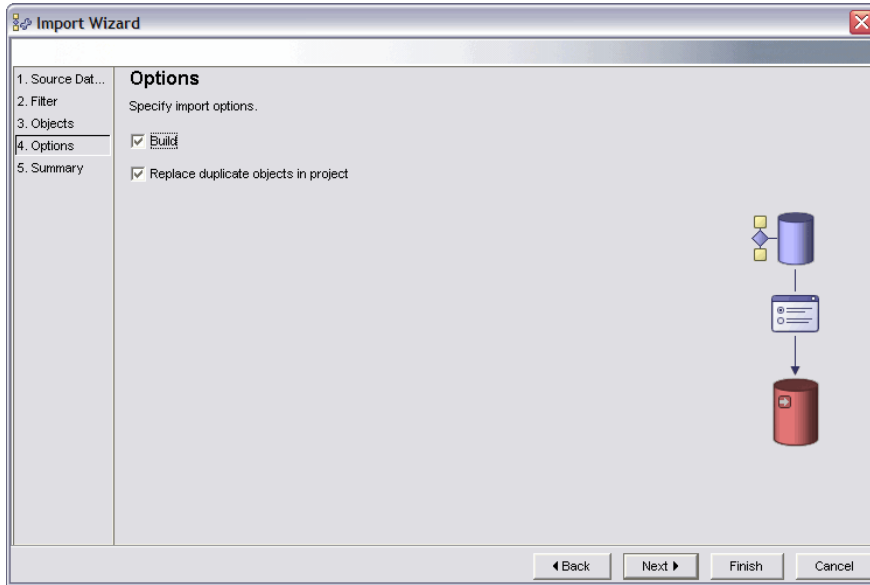


Figure 26. Import Options window

6. If you want to build the procedure during the import process, select **Build**.

The Summary window gives you an opportunity to confirm your import choices (see Figure 27. Import Summary window).

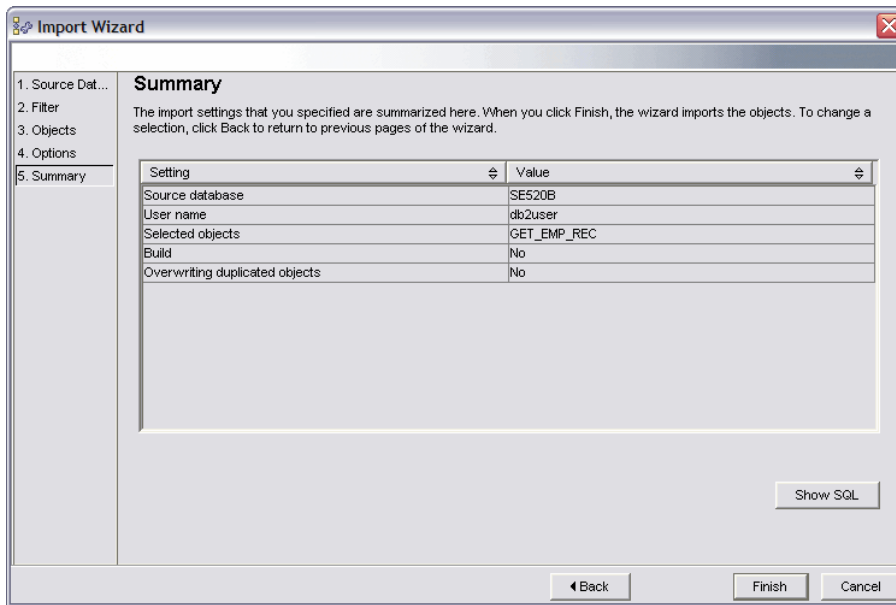


Figure 27. Import Summary window

7. Click **Finish** to begin the import process.



When the import finishes, the wizard closes and returns you to the DB2 Development Center main panel. In the Output View pane, click the **Messages** tab for feedback on results of the import and build operations (see Figure 28).

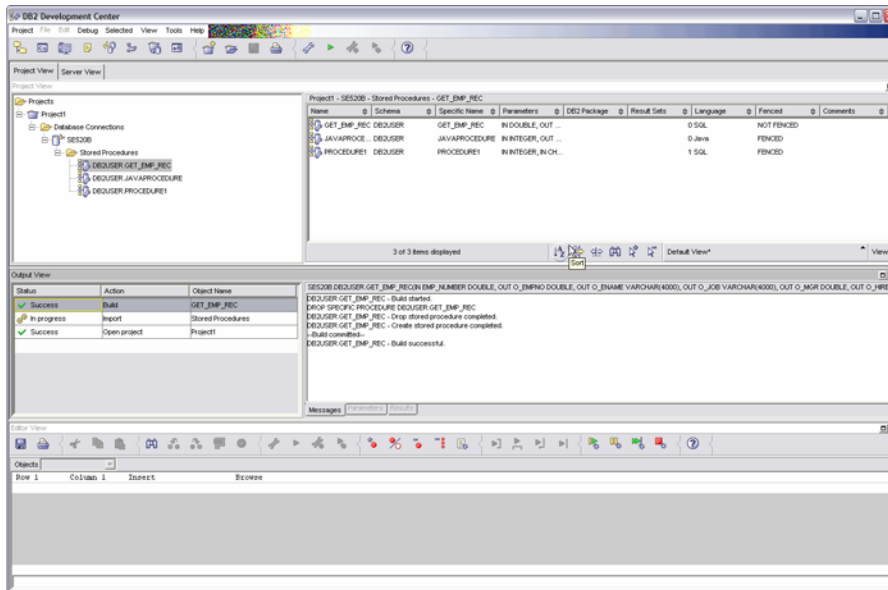


Figure 28. Messages tab showing that the import was successful

Running a procedure

After the procedure is successfully created or imported, use DB2 Development Center to run and test it. DB2 Development Center 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 29. Specify Parameter Values is displayed to prompt you for the needed input values.

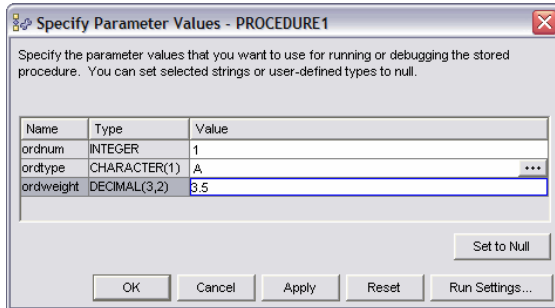


Figure 29. Specify Parameter Values window

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

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

DB2 Development Center 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 settings specified are persisted as part of your project settings for later reuse so it is not necessary to retype test parameters and options every time you test the procedures.



Editing procedures

Procedure or new functionality testing often reveals the need to change a stored procedure. You can do this by double-clicking the stored procedure object. In the procedure editor window (see Figure 30. Edit stored procedure), where you can manually make changes or use the editor pull-down menus to help code conditional tests and loops.

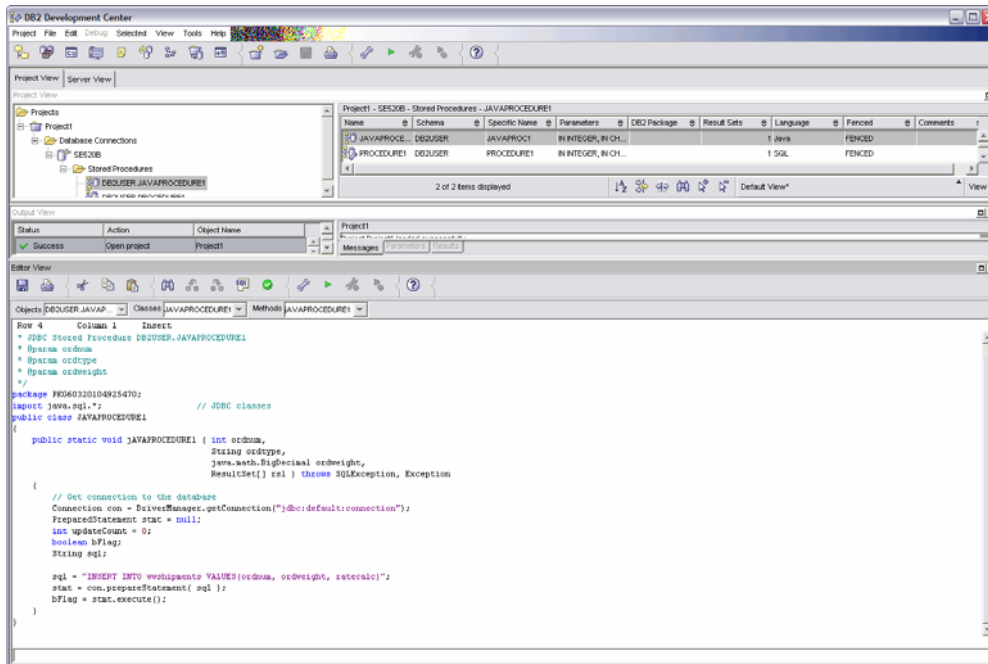


Figure 30. Edit stored procedure

As mentioned previously, you can import stored procedures that were not created with DB2 Development Center into the project. After importing, you can use the DB2 Development Center editor to modify the procedure. Another method of bringing such procedures into the project is by adding them to the project. This is a multistep process that does not use a wizard interface as the import process does.

1. In DB2 Development Center, click **Server View** (see Figure 31. Server View).

Note: The primary difference between importing and adding to the project is in the function of the Import wizard. Within the Import wizard, you can import multiple procedures and optionally initiate the build for each of them. In addition, you have the ability to import from a file, project, or database, and you do not need an existing connection to the source database.

2. Double-clicking the **Stored Procedures** folder to navigate through existing stored procedures. To bring a stored procedure into the current project, right-click the stored procedure object and click **Add to Project**. That procedure appears in your **Project View** tab where you can edit and build it with the DB2 Development Center tools.

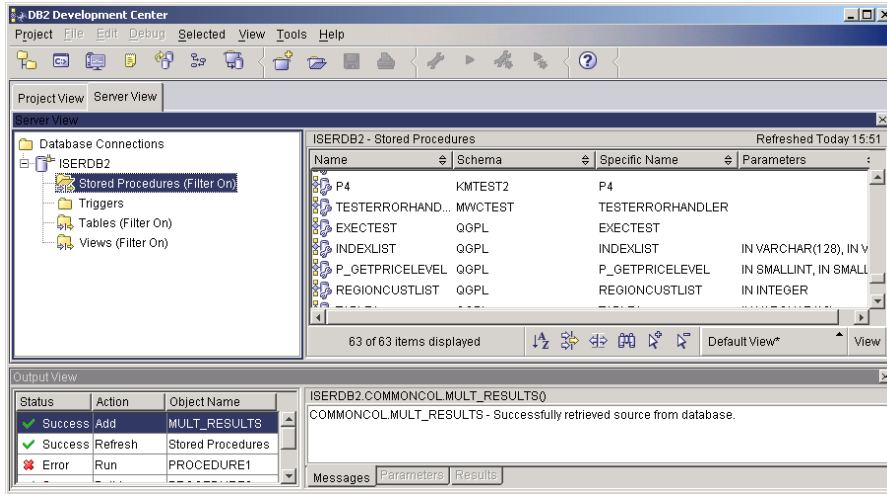


Figure 31. Server View tab

The **Server View** tab allows you to specify filters for each of the object folders. This makes it easy to list only those objects that are associated with a specific application or database. You can only view the source and properties for *trigger*, *view*, and *table* objects; you cannot add the source for these objects to a project.

Exporting and deploying procedures

DB2 Development Center provides three methods of deploying your stored procedures: exporting to a deployment project, exporting to a deployment script, and deploying directly to a target database.

Exporting to a deployment project

Exporting to a deployment project allows later redeployment into production using the Development Center Deployment Tool or command. This process is the easiest way to export and deploy, because it requires no user modification to the export files.

1. Access the Export task by right-clicking a project's **Stored Procedures** folder as shown in Figure 32. Exporting stored procedures.

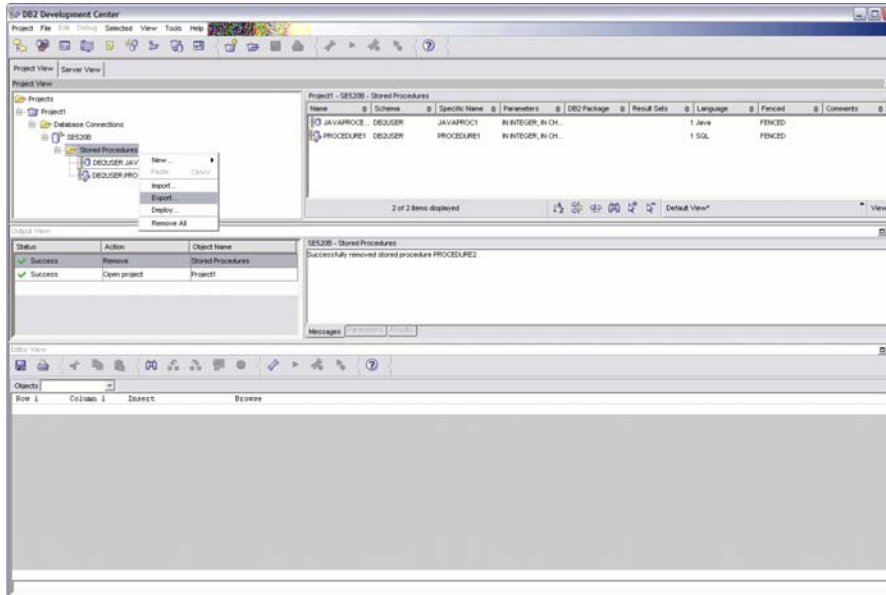


Figure 32. Exporting stored procedures

The **Export Wizard Objects** window is displayed (see Figure 33).

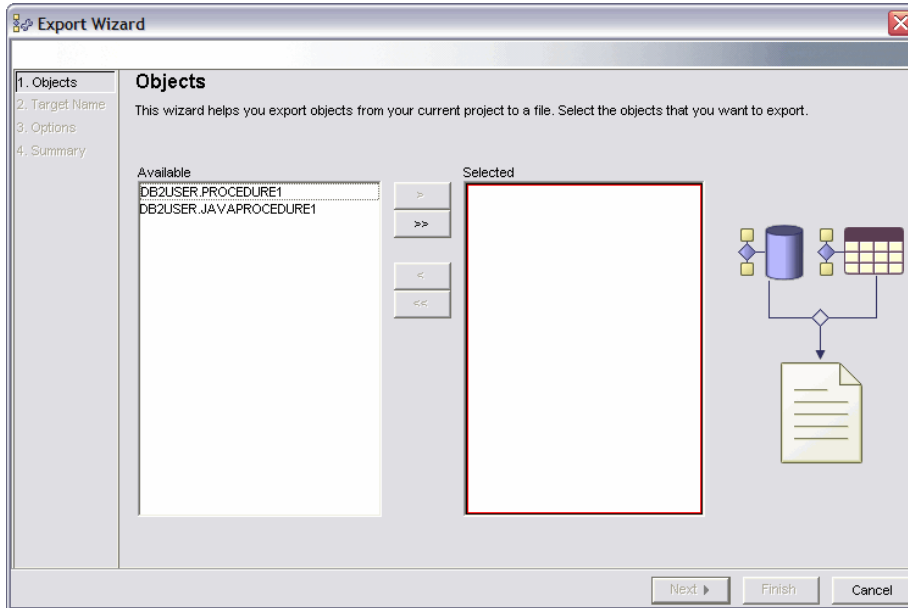


Figure 33. Export Wizard - Objects window

2. Select the objects to export and click the  button (or click the  button to select all available objects). When all of the objects to export appear in the **Selected** pane, click **Next**.

The Export Wizard Target Name window appears (see Figure 34: Export Wizard - Target Name).

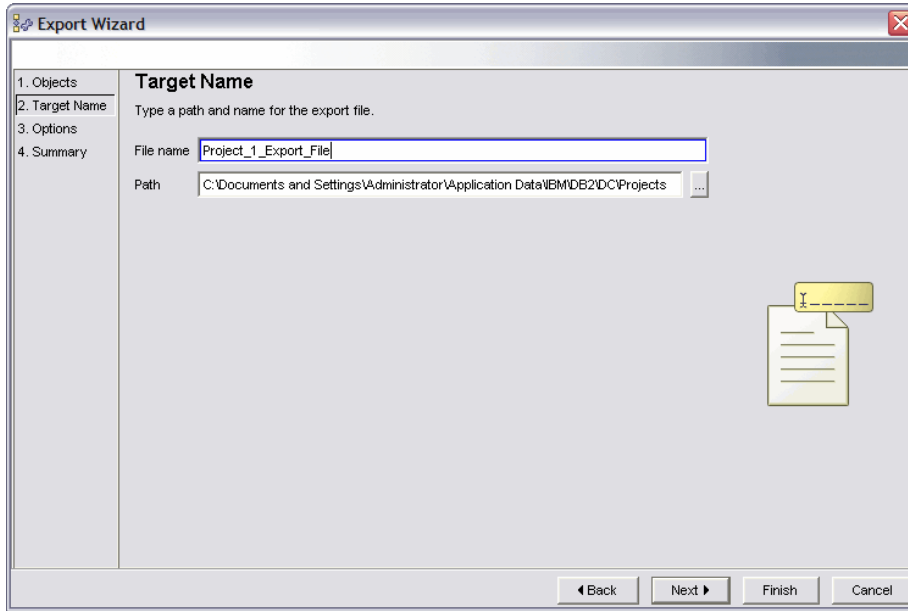


Figure 34: Export Wizard - Target Name window

3. Specify the path and file name of the export file and click **Next**.

The Export Wizard Options window is displayed (see Figure 35. Export Wizard – Options). This is where you select exporting to a project or a script.

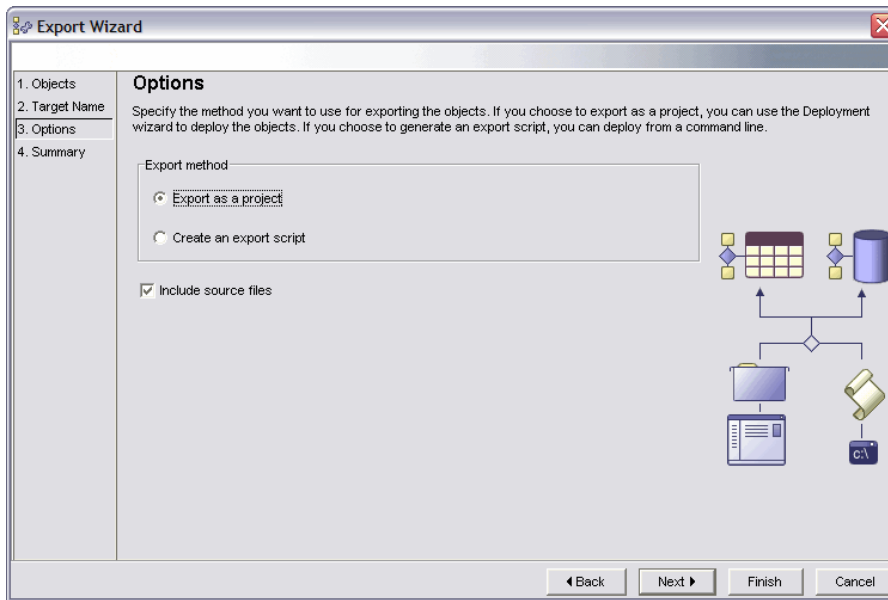


Figure 35. Export Wizard – Options window (Export as a project selection)

4. Select **Export to a project** and **Include source files**. Click **Finish** to complete the export request.

Exporting to a deployment script

Another deployment option is exporting to a script file. This method essentially puts your stored procedure source code into a text file. It 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 rerunning the script. Depending on the target database, you might have to modify the exported script files.

1. To specify exporting to a deployment script, follow steps 1-3 in the previous section (“Exporting to a deployment project” on page 31).
2. At the Export Wizard Options window, select **Create an export script and Include source files** as shown in Figure 36: Export Wizard - Options window (Create an export script. Click **Finish** to complete the export request.

Comment: Changed to make correspond to screen capture – please confirm that the text was wrong/outdated, NOT the screen cap.

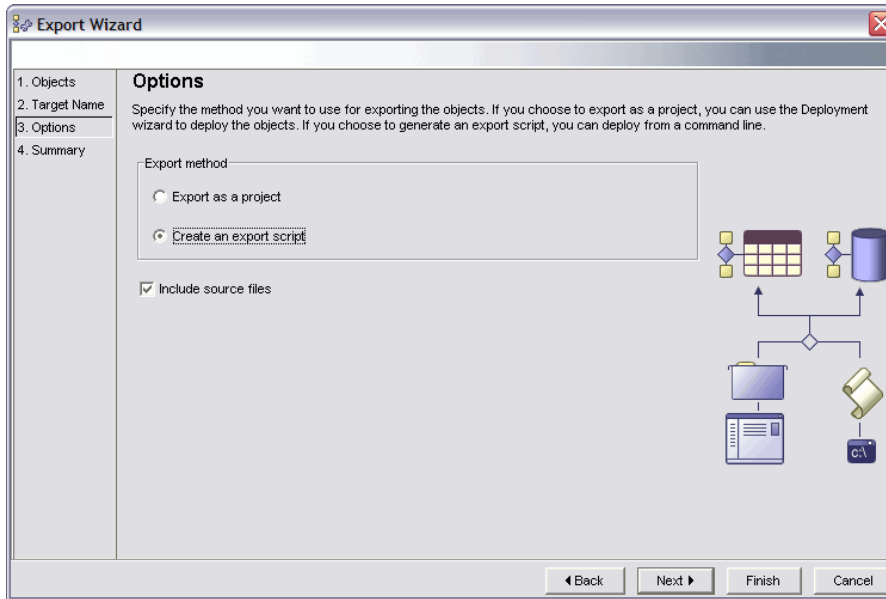


Figure 36: Export Wizard - Options window (Create an export script selection)

Deploying to target database

The last option available to you is deploying the objects immediately to a target production database. This method is appropriate if you have determined that an intermediate export step is not required and the objects and the 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 as shown in Figure 37. Deploying stored procedures.

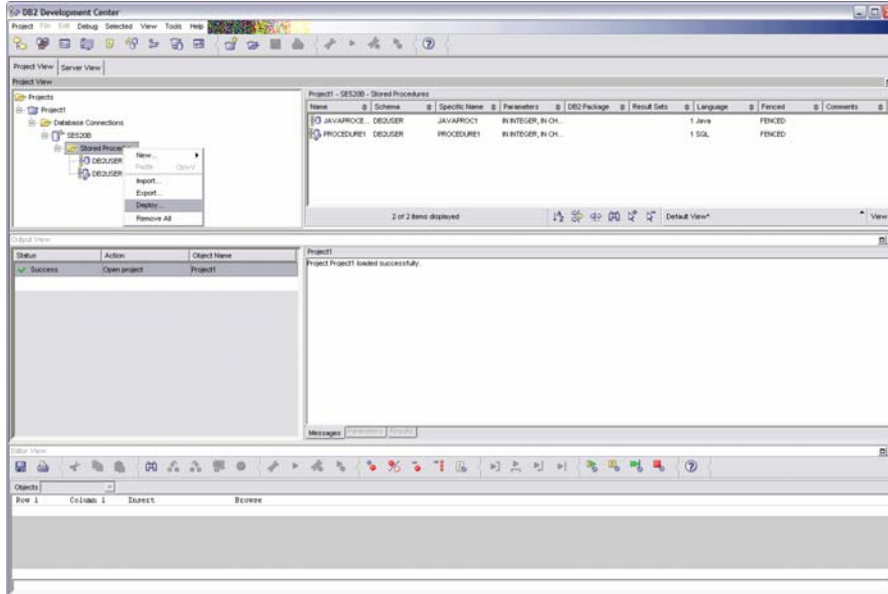


Figure 37. Deploying stored procedures

The Deployment Wizard - Target Database window is displayed (see Figure 38. Deployment Wizard - Target Database).

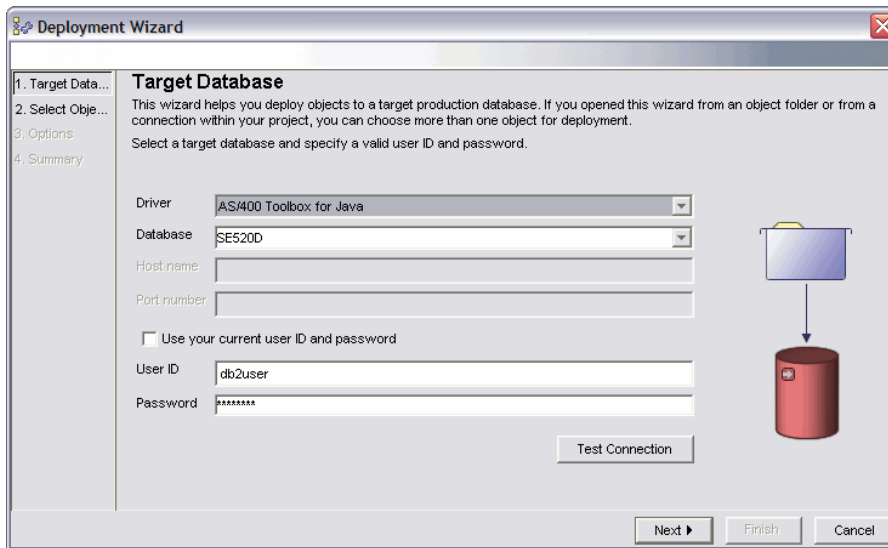


Figure 38. Deployment Wizard - Target Database window

2. From the **Driver** pull-down list, select **AS/400 Toolbox for Java**. Then, type the name of the database, the user ID, and password. Click **Next**.
3. At the Select Objects window, select all the objects to deploy and click **Next**.
4. At the Options window, click **Finish** to submit the deployment request.
5. Back at the Development Center main panel, check the status in the **Output View** pane to monitor the progress of the deployment request.



Summary

The intent of this white paper is to help you understand the DB2 Development Center capabilities and how you can use them to create, test, and deploy SQL stored procedures on DB2 for i5/OS. This iterative development toolset provides an environment that simplifies your efforts to develop 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 eServer™ i5 Information Center
<http://publib.boulder.ibm.com/series/>
- DB2 for i5/OS online manuals
ibm.com/series/db2/books.html
- DB2 Information Center
<http://publib.boulder.ibm.com/infocenter/db2help/>
- IBM Publications Center
www.elink.ibm.com/public/applications/publications/cgi-bin/pbi.cgi?CTY=US
- IBM Redbooks™
www.redbooks.ibm.com/
- IBM Toolbox for Java driver download site
www.iseries.ibm.com/toolbox

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 IBM System i5™ relational database topics.



Gene Cobb is a DB2 technology specialist in eServer Solutions Enablement. He has worked on IBM midrange systems since 1988, with 10 years in the IBM Client Technology Center (CTC), IBM Rochester. While in the CTC, he assisted IBM clients with application design and development using RPG, DB2 for i5/OS, 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 1994-2006. 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.

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

DB2	i5/OS	Lotus	System i5
DB2 Connect	IBM	OS/400	
Domino	the IBM logo	Redbooks	
eServer	iSeries	System i	

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.