IBM VisualAge® for Java™, Version 3.5

DB2® Stored Procedure Builder

Note!

Before using this information and the product it supports, be sure to read the general information under **Notices**.

Edition notice

This edition applies to Version 3.5 of IBM VisualAge for Java and to all subsequent releases and modifications until otherwise indicated in new editions.

Contents

Chapter 1. Overview of DB2 Stored Procedure Builder

DB2 Stored Procedure Builder assists you with creating a stored procedure that runs on a database server. You must write the client application separately. For information about writing a client application that calls a stored procedure, see the following guides:

- *IBM DB2 Universal Database™ Application Development Guide, for information about designing and programming DB2 applications*
- *IBM DB2 Universal Database Application Building Guide, for information about compiling and linking DB2 applications*
- *IBM DB2 Universal Database for OS/390® Application Programming and SQL Guide*
- *IBM DB2 Universal Database Application Development Guide—Embedded SQL*

RELATED CONCEPTS

What is DB2 Stored Procedure Builder?
How has DB2 Stored Procedure Builder changed the process of creating stored procedures?

RELATED TASKS

Managing Stored Procedure Builder projects
Ways to create stored procedures
Finding more information on DB2 Stored Procedure Builder

What are stored procedures?

All database access must go across the network which, in some cases, results in poor performance. For each SQL statement, a database manager application must initiate a separate communication with DB2(R).

Stored procedures are programs that have the following characteristics:

- Run on DB2 servers and access data in DB2 databases
- Contain blocks of procedural constructs and embedded SQL statements that are stored in databases and can be called by name
- Allow an application program to be run in two parts—-the calling application on the client and the stored procedure on the server

The following figure shows how a database manager application that does not use stored procedures accesses a database located on a database server.

To improve application performance, you can create stored procedures that run on your database server. A client application can call the stored procedure to obtain a result.

A typical stored procedure contains one or more SQL statements and some manipulative or logical processing in a programming language. The client application program uses the SQL CALL statement to invoke the stored procedure. When the stored procedure runs, it locally accesses the database on the database server and returns requested information to the client application.

The following figure shows how an application using a stored procedure accesses a database server.

To create a stored procedure, you write two separate programs: the stored procedure itself, which runs on a database server, and a client application, which runs on a client workstation or a middleware server (such as a Web server). The client application calls the stored procedure by using one of the available API methods.

A stored procedure follows certain conventions for exchanging data with a client application. A stored procedure does not connect to the database itself, but relies on the database connection already established by the client. The call across the network includes parameters required by the stored procedure. The stored procedure uses the parameters to complete its logic, runs, and can return a set of values, rows, or modified parameters to the calling client application.

RELATED CONCEPTS

What is DB2 Stored Procedure Builder?
What are the benefits of using stored procedures?
How has DB2 Stored Procedure Builder changed the process of creating stored procedures?

RELATED TASKS

Ways to create stored procedures
Managing Stored Procedure Builder projects
Finding more information on DB2 Stored Procedure Builder

What are the benefits of using stored procedures?

Applications using stored procedures have the following advantages:

- Reduced network traffic between clients and servers

  A client application passes control to a stored procedure on the database server. The stored procedure performs intermediate processing on the database server, without transmitting unnecessary data across the network. Only those records that are actually required by the client application are transmitted. Using a stored procedure can result in reduced network traffic and better overall performance.

  Also, applications that execute SQL statements one at a time typically generate two trips across the network for each SQL statement. A stored procedure can group SQL statements together and generate only two trips across the network for each group of SQL statements. The more SQL statements you group together in a stored procedure, the more you reduce network traffic and the time that database locks are held. Reducing network traffic and the time of database locks improves overall network performance and reduces lock contention problems.

  Applications that process large amounts of SQL-generated data, but present only a subset of the data to the user, can generate excessive network traffic because all of the data is returned to the client before final processing. A stored procedure can do the processing on the server, and transmit only the required data to the client, which reduces network traffic.

- Enhanced hardware and software capabilities

  Applications using stored procedures have access to increased memory and disk space if the server computer is so equipped and access to software installed only on the database server.

- Improved security by encapsulating database privileges for stored procedures that use static SQL

The database administrator (DBA) or developer who builds the stored procedure must have the database privileges the stored procedure requires, but the users of the client applications that call the stored procedure do not need such privileges.

- Reduced development cost and increased reliability from reusing common routines in multiple applications

  In a database application environment, many situations are recurrent, such as returning a fixed set of data, or performing the same set of multiple requests to a database. Stored procedures provide a highly efficient way to address these recurrent situations by reusing one common procedure.
- Centralized security, administration, and maintenance for common routines

  You can simplify security, administration, and maintenance by managing shared logic in one place at the server, instead of managing multiple copies of the same business logic on client workstations. Client applications can call stored procedures that run SQL queries with little or no additional processing. By using stored procedures, you gain centralized maintenance and authorization and can potentially encapsulate the database tables.

RELATED CONCEPTS

What are stored procedures?

RELATED TASKS

Ways to create stored procedures
Finding more information on DB2 Stored Procedure Builder

What is DB2 Stored Procedure Builder?

DB2 Stored Procedure Builder is a graphical application that supports the rapid development of DB2 stored procedures. Using Stored Procedure Builder, you can perform the following tasks:

- Create stored procedures
- Build stored procedures on local and remote DB2 servers
- Modify and rebuild existing stored procedures
- Run stored procedures for testing and debugging the execution of installed stored procedures

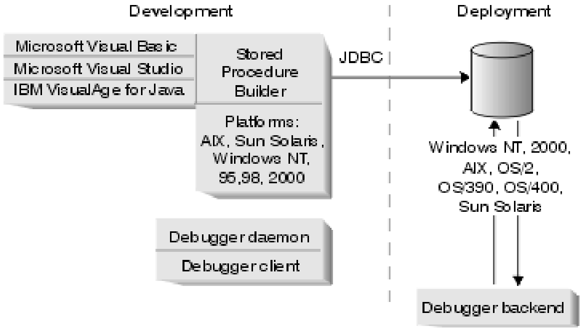
  Tip: Using Stored Procedure Builder, you can also run stored procedures that you created in another editor and that use a language other than Java or the SQL procedure language. You might want to test the execution of existing installed stored procedures.

Stored Procedure Builder provides a single development environment that supports the entire DB2 family ranging from the workstation to OS/390(R). You can launch Stored Procedure Builder as a separate application from the DB2 Universal Database program group or from any of the following development applications:

- IBM VisualAge(R) for Java Version 3.0 or later
- Microsoft(R) Visual C++(R) Version 5 and Version 6
- Microsoft Visual Basic(R) Version 5 and Version 6
- IBM DB2 Control Center

Stored Procedure Builder is implemented with Java and all database connections are managed by using a Java Database Connectivity (JDBC) API. Using a JDBC driver, you can connect to any DB2 database by using a local alias.

The following diagram illustrates the Stored Procedure Builder topology:



RELATED CONCEPTS

What are stored procedures?
How has DB2 Stored Procedure Builder changed the process of creating stored procedures?
Overview of debugging stored procedures

RELATED TASKS

Ways to create stored procedures
Managing Stored Procedure Builder projects
Modifying existing Java stored procedure source code
Modifying existing SQL stored procedure source code
Running stored procedures
Finding more information on DB2 Stored Procedure Builder

How has DB2 Stored Procedure Builder changed the process of creating stored procedures?

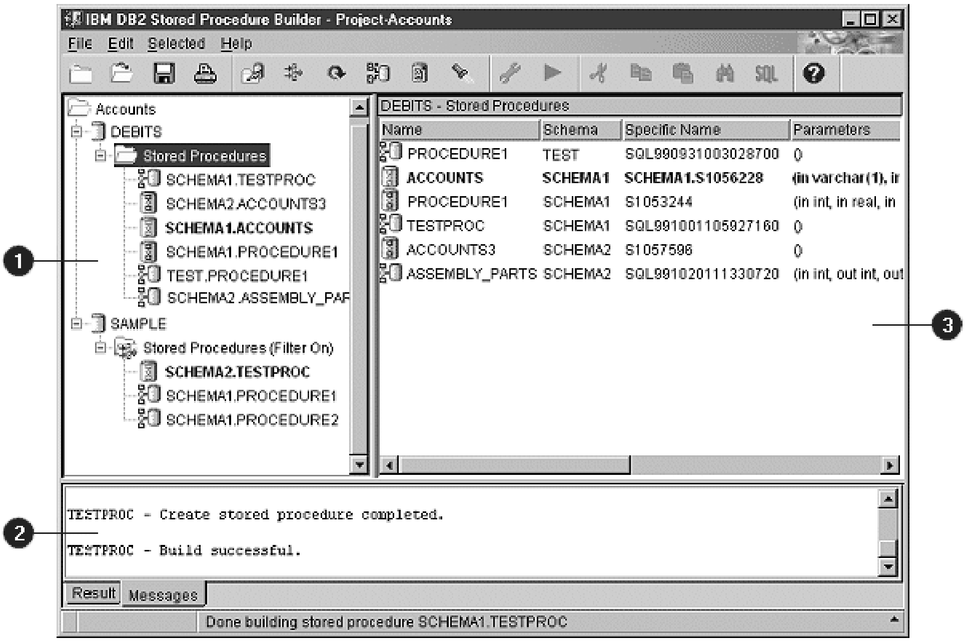
DB2 Stored Procedure Builder provides an easy-to-use development environment for creating, installing, and testing stored procedures. Using Stored Procedure Builder allows you to focus on creating your stored procedure logic rather than the details of registering, building, and installing stored procedures on a DB2 server. Additionally, with Stored Procedure Builder, you can develop stored procedures on one operating system and build them on other server operating systems.

Creating Stored Procedure Builder projects
Stored Procedure Builder manages your work by using projects. Each Stored Procedure Builder project saves the following information:

- Your connections to specific databases.
- The filters you created to display subsets of the stored procedures on each database. When opening a new or existing Stored Procedure Builder project, you can filter stored procedures so that you view stored procedures based on their name, schema, language, or collection ID (for OS/390 only).
- Stored procedure objects that have not been successfully built to a target database.

You can identify Stored Procedure Builder project files by their SPP file extension.

The Project window shows all of the stored procedures on the DB2 database to which you are connected. See the following diagram, which describes the parts of the Project window.



- Accounts
  - DEBITS
    - Stored Procedures
      - SCHEMA1.TESTPROC
      - SCHEMA2.ACCOUNTS3
      - **SCHEMA1.ACCOUNTS**
      - SCHEMA1.PROCEDURE1
      - TEST.PROCEDURE1
      - SCHEMA2.ASSEMBLY_PAF
  - SAMPLE
    - Stored Procedures (Filter On)
      - **SCHEMA2.TESTPROC**
      - SCHEMA1.PROCEDURE1
      - SCHEMA1.PROCEDURE2

DEBITS - Stored Procedures

Name	Schema	Specific Name	Parameters
PROCEDURE1	TEST	SQL990931003028700	()
ACCOUNTS	**SCHEMA1**	**SCHEMA1.S1056228**	**(in varchar(1), ir**
PROCEDURE1	SCHEMA1	S1053244	(in int, in real, in
TESTPROC	SCHEMA1	SQL991001105927160	()
ACCOUNTS3	SCHEMA2	S1057596	()
ASSEMBLY_PARTS	SCHEMA2	SQL991020111330720	(in int, out int, out

```
TESTPROC - Create stored procedure completed.

TESTPROC - Build successful.
```

Done building stored procedure SCHEMA1.TESTPROC

1 The tree pane, which shows all the objects in your project

2 The messages and result pane, which shows the messages that Stored Procedure Builder generates when you build or run a stored procedure and the result sets returned when running a stored procedure

3 The contents pane, which shows detailed information for each stored procedure in a connection

In the tree pane, the following objects appear in this order: the project folder, the database connection, the **Stored Procedures** folder, and stored procedures. Depending on their state and language, stored procedures have different icons:

Java stored procedure

SQL stored procedure

Stored procedure created in a language other than Java or SQL

Java stored procedure that has not been built to the database or that has changed since it was last built to the database

SQL stored procedure that has not been built to the database or that has changed since it was last built to the database

Creating stored procedures

Stored Procedure Builder greatly simplifies the process of creating and installing stored procedures on a DB2 database server. When you build a stored procedure on a target database, you no longer have to manually register the stored procedure with DB2 by using the CREATE PROCEDURE statement.

With Stored Procedure Builder, you create stored procedures in Java and the SQL procedure language. Creating stored procedures in Java and the SQL procedure language produces stored procedures that are highly portable among operating systems.

Using a stored procedure wizard and SQL Assistant facilitate the development of stored procedures. You launch a stored procedure wizard from the Project window by selecting the **Stored Procedures** folder; then, use the Selected menu to create a new stored procedure. You can use a stored procedure wizard to create your basic SQL structure; then, you can use the code editor to add highly sophisticated stored procedure logic.

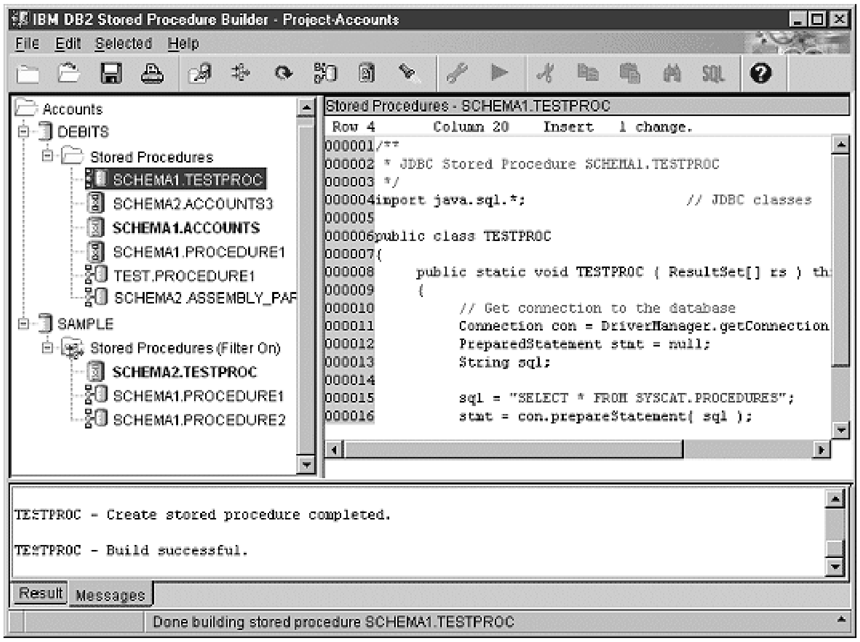
You can also create stored procedures without using a stored procedure wizard. You can create stored procedures by using a template that creates the stored procedure structure but does not include any SQL statements. After generating the basic stored procedure structure, you can use the code editor to further modify the generated code to add logic and SQL statements.

See Ways to create stored procedures for more information.

When creating a stored procedure, you can choose to return a single result set, multiple result sets, or output parameters only. You might choose not to return a result set or output parameters when your stored procedure creates or updates database tables.

You can use a stored procedure wizard to define input and output parameters for a stored procedure so that it receives values for host variables from the client application. Additionally, you can create multiple SQL statements in a stored procedure; the stored procedure receives a case expression and selects one of a number of queries.

Editing stored procedures
Stored Procedure Builder includes a fully-functional code editor that is language sensitive depending on whether the stored procedure was written in Java or the SQL procedure language. Once you have created a stored procedure, you can easily modify it by opening the source code in the Stored Procedure Builder code editor. When you open a stored procedure, the source code is displayed in the editor pane on the right side.



```
 Row 4       Column 20    Insert    1 change.
/**
 * JDBC Stored Procedure SCHEMA1.TESTPROC
 */
import java.sql.*;                  // JDBC classes

public class TESTPROC
{
    public static void TESTPROC ( ResultSet[] rs ) th
    {
        // Get connection to the database
        Connection con = DriverManager.getConnection
        PreparedStatement stmt = null;
        String sql;

        sql = "SELECT * FROM SYSCAT.PROCEDURES";
        stmt = con.prepareStatement( sql );
```

```
TESTPROC - Create stored procedure completed.

TESTPROC - Build successful.
```

Done building stored procedure SCHEMA1.TESTPROC

Using the code editor, you can add parameters, result sets, and additional stored procedure logic. By modifying a Java stored procedure, you can add methods to the code to include sophisticated stored procedure logic. For Java stored procedures, you can add logic to the method that is called when the stored procedure starts. In Java stored procedures and SQL stored procedures, you can add calls to other stored procedures and run other SQL statements.

See the following topics for detailed information about how you can use the code editor in Stored Procedure Builder:

- Modifying existing Java stored procedure source code
- Modifying existing SQL stored procedure source code
- Editing generated code

Working with existing stored procedures

After you successfully build a stored procedure on a database server, you are ready to rebuild, run, and test the procedure. Running a stored procedure within Stored Procedure Builder allows you to ensure that the stored procedure is correctly installed and that the stored procedure logic is working as expected. When you run a stored procedure, depending on how you set it up, it can return result sets and output parameters based on test input parameter values that you enter. When you run a stored procedure that has input parameters, you are prompted for any required parameters.

For Java stored procedures, you can view the Data Definition Language (DDL) statements by right-clicking a stored procedure in the tree and selecting **Show SQL**.

From the Project window, you can drop a stored procedure or copy it to another database connection.

You can search for specific text in the contents page of the Stored Procedure Builder Project window by selecting the **Stored Procedures** folder and clicking the **Search** toolbar button. Stored Procedure Builder notifies you when no matches are found.

See the following topics for detailed information about how you can work with existing stored procedures:

- Running stored procedures
- Copying and pasting stored procedures to another database
- Dropping stored procedures

Remotely debugging stored procedures

Using Stored Procedure Builder and the IBM Distributed Debugger, you can remotely debug a stored procedure installed on a DB2 server. When you have enabled debugging for a stored procedure, the IBM Distributed Debugger starts automatically when you run the stored procedure. You are not required to debug the stored procedure from within an application program. You can separate testing your stored procedure from testing the application program.

When a stored procedure is built to a workstation database, you can view who has the authority to change, add, or remove debug records in the stored procedures debug table. When a stored procedure is built to an OS/390 server, the debugging options are stored in a DB2 catalog table.

The Stored Procedure Properties notebook contains information about the debug options set for a stored procedure. If you are a database administrator (DBA) or you created the selected stored procedure, you can grant database authorization to other users to debug the stored procedure.

For information about debugging stored procedures, see Overview of debugging stored procedures.

RELATED CONCEPTS

What is DB2 Stored Procedure Builder
What are stored procedures?

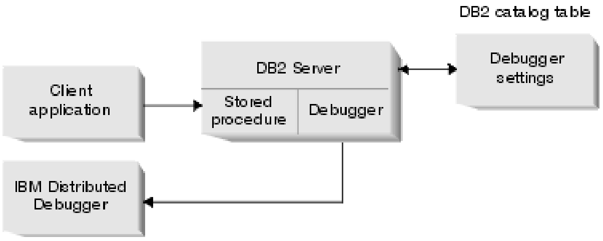
RELATED TASKS

Browsing and modifying stored procedure properties
Setting project properties
Setting environment properties
Copying and pasting stored procedures to another database
Finding more information on DB2 Stored Procedure Builder

Overview of debugging stored procedures

Using DB2 Stored Procedure Builder, the IBM Distributed Debugger (available separately as part of VisualAge for Java), and the debugger component running on the DB2 server, you can remotely debug a stored procedure installed on a DB2 server. You are not required to debug a stored procedure from within an application program because you can run a stored procedure in debug mode within Stored Procedure Builder. Therefore, you can separate debugging your stored procedure from debugging the client application program that calls the stored procedure.

When you build a stored procedure for debugging and run it in Stored Procedure Builder, DB2 initializes the debugger component on the database server. The debugger component on the DB2 server uses the IP address and port that you set in Stored Procedure Builder to launch the IBM Distributed Debugger on your workstation.



The IBM Distributed Debugger reviews each programming instruction and displays the states of the variables in the executing stored procedure as it runs on the DB2 server.

For information about how to use the IBM Distributed Debugger, see the VisualAge for Java documentation.

To set up a stored procedure in Stored Procedure Builder for debugging, you must:

- Build the stored procedure in debug mode

When you create a stored procedure with a stored procedure wizard, select **Enable debugging** from the Options page.

For existing stored procedures, you can enable debugging by selecting **Build the stored procedure for debugging** on the Build and Debug page of the Stored Procedure Properties notebook.

- When you are building stored procedures to a workstation server, enter debug records for the stored procedure in the debug table to grant debugging authorization to particular users. These records are stored in DB2DBG.ROUTINE_DEBUG on the DB2 server. If you are a DBA or the creator of a selected stored procedure, you can authorize other users to debug the stored procedure by entering debug records for the stored procedure in the Stored Procedure Properties notebook.
  You can also enable or disable debugging for several stored procedures within the same connection by right-clicking the database connection in the tree and selecting **Debug Records**.

  Restriction: For SQL stored procedures, the RUNOPTS column of the table SYSIBM.SYSPROCEDURES stores the port and IP address that the debugger uses to connect to your client workstation. Stored procedures built on a DB2 for OS/390 database server do not use the debug table. Therefore, *only* one user at a time can debug a stored procedure.
- **Optional:** Change the global settings (port and IP address) that Stored Procedure Builder uses for the connection between the IBM Distributed Debugger and stored procedures by using the Debug page in the Environment Properties notebook. When you change these global settings, they affect stored procedures that you subsequently create. The IP address and port setting for existing stored procedures are not affected.

RELATED CONCEPTS

What is DB2 Stored Procedure Builder?
What are stored procedures?

RELATED TASKS

Ways to create stored procedures
Finding more information on DB2 Stored Procedure Builder

Chapter 2. Managing Stored Procedure Builder projects

DB2 Stored Procedure Builder manages your work by using projects. Stored Procedure Builder projects save connection information and stored procedure objects that have not yet been built to a database.Specifically, each Stored Procedure Builder project saves the following information:

- Your connections to specific databases.
- The filters you created to display subsets of the stored procedures on each database. For a Stored Procedure Builder project, you can filter stored procedures so that you view stored procedures based on their name, schema, language, or collection ID (for OS/390 only).
- Stored procedure objects that have not been successfully built to a target database.

Tip: You can identify Stored Procedure Builder project files by their SPP file extension.

The Project window shows all of the stored procedures on the DB2 database to which you are connected. See the following diagram, which describes the parts of the Project window.

Once you have created a Stored Procedure Builder project, you can create stored procedures, add connections to databases, set project and environment properties, run stored procedures, debug stored procedures, and drop stored procedures.

RELATED CONCEPTS

What is DB2 Stored Procedure Builder?
How has DB2 Stored Procedure Builder changed the process of creating stored procedures?

RELATED TASKS

Managing multiple database connections in a project
Setting project properties
Finding more information on DB2 Stored Procedure Builder

Setting project properties

To edit the characteristics of a project:

1. Select **File —> Project Properties**. The Project Properties notebook opens to the Project page.
2. Edit the current project name by typing over the name in the **Project name** field. The project name can be up to 255 characters, including the project path and filename extension, for example, if your project path contains 50 characters, then your project name can contain a maximum of 205 characters. The project name can contain alphanumeric, underscore (_), and space characters. If you give your project the same name as an existing project, you are prompted to overwrite the existing project when you save.

   When you change the project name, Stored Procedure Builder changes the current file name of the project to match your new project name. You are prompted to save the project with the new file name when you save or exit the project.

3. Edit the project path by typing a new path in the **Project path** field. When you change the project path, you are prompted to save the project in the new location when you save or exit the project. Click the ... button to open the Choose Directory window and select a directory for the project.
4. **Optional:** Type comments about the project in the **Description** field.
5. **Optional:** Click the **Copyright** tab to enter copyright information for this project.
6. Click **OK** to apply your changes.

RELATED CONCEPTS

What is DB2 Stored Procedure Builder?
How has DB2 Stored Procedure Builder changed the process of creating stored procedures?

RELATED TASKS

Setting environment properties
Browsing and modifying stored procedure properties
Finding more information on DB2 Stored Procedure Builder

Setting environment properties

Use the Environment Properties notebook to specify default connection, result, debugging, and SQL type synonym and mapping information. Select **File —> Environment Properties** to display the Environment Properties notebook.

Tip: If you have changed the default settings from the Environment Properties notebook and want to restore the default settings for your environment, delete the DB2SPB.INI file. When you open the Environment Properties notebook, Stored Procedure Builder recreates the INI file with the default settings. You can also delete portions of the INI file to restore only selected settings to the defaults.

RELATED CONCEPTS

What is DB2 Stored Procedure Builder?
How has DB2 Stored Procedure Builder changed the process of creating stored procedures?

RELATED TASKS

Setting project properties
Browsing and modifying stored procedure properties
Finding more information on DB2 Stored Procedure Builder

Ways to create stored procedures

DB2 Stored Procedure Builder greatly simplifies the process of creating and installing stored procedures on a DB2 database server. With Stored Procedure Builder, you create stored procedures in Java and the SQL procedure language. Stored Procedure Builder provides different ways to create stored procedures depending on the language that you choose to create a stored procedure.

Create stored procedures	Languages	Descriptions and benefits

Using a wizard	Java and SQL	Using a stored procedure wizard and SQL Assistant facilitate the development of stored procedures. To start a stored procedure wizard: 1. From the Project window, select the **Stored Procedures** folder. 2. Select **Selected —> Insert —> Java Stored Procedure using Wizard** or **SQL Stored Procedure using Wizard** from the menu. Using a stored procedure wizard, you create your basic SQL structure; then, you can use the code editor to add highly sophisticated stored procedure logic.
From a class by using a wizard	Java	Using the Inserting Java Stored Procedure from class wizard allows you to create a new Java stored procedure from an existing Java class. 1. From the Project window, select the **Stored Procedures** folder. 2. Select **Selected —> Insert —> Java Stored Procedure from Class using Wizard** from the menu.

Using a wizard to generate stored procedures that do not contain SQL statements	Java and SQL	Using a template provides a quick way to create your basic SQL structure for a stored procedure. When you select this option in a stored procedure wizard, Stored Procedure Builder creates the stored procedure structure but does not include any SQL statements or an open cursor in the stored procedure. 1. From the Project window, select the **Stored Procedures** folder. 2. Select **Selected —> Insert —> Java Stored Procedure using Wizard** or **SQL Stored Procedure using Wizard** from the menu. 3. Click the **Pattern** tab. 4. Select the **Generate no SQL statement** radio button.

Using a template that contains the default stored procedure structure	SQL	Using a temple creates the default structure for an SQL stored procedure without parameters or SQL statements. The stored procedure wizards create stored procedures that have simple stored procedure logic; for example, the wizards create stored procedures with one SQL statement and all variables become parameters. If you want your stored procedure to contain sophisticated business logic or you want to reuse logic from another stored procedure, you can use an SQL template and add the logic in the Stored Procedure Builder code editor. 1. From the Project window, select the **Stored Procedures** folder. 2. Select **Selected —> Insert —> SQL Stored Procedure**. The SQL stored procedure code appears in the code editor. 3. **Optional**: Use the Stored Procedure Properties notebook to add comments to the stored procedure or to change the default settings for the build and debug options.

RELATED CONCEPTS

What is DB2 Stored Procedure Builder?
How has DB2 Stored Procedure Builder changed the process of creating stored procedures?

RELATED TASKS

Modifying existing Java stored procedure source code
Modifying existing SQL stored procedure source code
Editing generated code
Finding more information on DB2 Stored Procedure Builder

Modifying existing Java stored procedure source code

If you open a project that contains existing Java stored procedures, you can modify the source code (including SQL statements) within DB2 Stored Procedure Builder. Stored Procedure Builder drops the old stored procedure from the database and creates a stored procedure that reflects the changes you made.

Requirement: Java stored procedures built by Stored Procedure Builder conform to the *SQLJ Routines* specification. Java stored procedures are defined in the catalog table with LANGUAGE JAVA and PARAMETER STYLE JAVA.
Java stored procedures must follow these rules:

- The method mapped to the stored procedure must be defined as a public static void method.
- The stored procedure must receive input parameters as host variables.
- Output and InOut parameters must be set up as single element arrays.

To modify existing Java stored procedure source code:

1. In the tree, right-click the Java stored procedure that you want to modify and select **Get Source**. The source code of the stored procedure displays in the editor pane.
   Tip: You can determine which stored procedures contain Java by clicking the **Stored Procedures** folder in the tree and viewing the **Language** column in the Contents pane. Java stored procedures are also represented in the tree with this icon:
2. Refer to Editing generated code for information about editing the source code of the stored procedure.
3. Add SQL statements to the source code by using the editor or launch the SQL Assistant by right-clicking in the editor and selecting **Insert SQL**. The placement of the cursor in the editor determines where SQL Assistant places the SQL statements.

   In SQL Assistant, the tables that you can view and build queries from are those of the current database alias.
   Tip: To replace an existing SQL statement with an SQL statement created in SQL Assistant, select the entire SQL statement in the editor and then launch SQL Assistant to create a new query. SQL Assistant replaces the selected SQL statements in the editor with the new SQL statement. Note that you cannot modify an existing SQL statement by using SQL Assistant.
4. When you are finished modifying the code, you can build the stored procedure to the database by right-clicking the stored procedure in the tree and selecting **Build**. Note that a build must be performed before changes to the stored procedure are reflected in the database.
   Tip: If the build fails and build errors display in the Messages pane, you can double-click an error message to see the referenced line highlighted in the editor.

RELATED CONCEPTS

What is DB2 Stored Procedure Builder?
How has DB2 Stored Procedure Builder changed the process of creating stored procedures?

RELATED TASKS

Editing generated code
Modifying existing SQL stored procedure source code
Finding more information on DB2 Stored Procedure Builder

Modifying existing SQL stored procedure source code

If you open a project that contains existing SQL stored procedures, you can modify the source code (including SQL statements) within DB2 Stored Procedure Builder. Stored Procedure Builder drops the old stored procedure from the database and creates a stored procedure that reflects the changes you made.

To modify existing SQL stored procedure source code:

1. In the tree, right-click the SQL stored procedure that you want to modify and select **Get Source**. The source code of the stored procedure displays in the editor pane.
   Tip: You can determine which stored procedures are written in SQL by clicking the **Stored Procedures** folder in the tree and viewing the **Language** column in the Contents pane. SQL stored procedures are also represented in the tree with this icon:
2. Refer to Editing generated code for information about editing the source code of the stored procedure.
3. Add SQL statements to the source code by using the editor or launch the SQL Assistant by right-clicking in the editor and selecting **Insert SQL**. The placement of the cursor in the editor determines where SQL Assistant places the SQL statements.

   In SQL Assistant, the tables that you can view and build queries from are those of the current database alias.
   Tip: To replace an existing SQL statement with an SQL statement created in SQL Assistant, select the entire SQL statement in the editor and then launch SQL Assistant to create a new query. SQL Assistant replaces the selected SQL statement in the editor with the new SQL statement. Note that you cannot modify an existing SQL statement by using SQL Assistant.
4. Insert SQL code fragments that provide the basic structure of SQL statements such as IF, CASE, LOOP, REPEAT, and WHILE by right-clicking in the editor and selecting **Insert Code Fragment**. Add logic to these SQL statements by using the code editor.
5. When you are finished modifying the code, you can build the stored procedure to the database by right-clicking the stored procedure in the tree and selecting **Build**. Note that a build must be performed before changes to the stored procedure are reflected in the database.

RELATED CONCEPTS

What is DB2 Stored Procedure Builder?
How has DB2 Stored Procedure Builder changed the process of creating stored procedures?

RELATED TASKS

Editing generated code
Modifying existing Java stored procedure source code
Finding more information on DB2 Stored Procedure Builder

Browsing and modifying stored procedure properties

Use the Stored Procedure Properties notebook to view and modify the parameters and other characteristics of an existing stored procedure. The stored procedure properties that you can modify with this notebook vary depending on the language (Java or SQL) of the stored procedure and the database in which it is built.

The characteristics that you cannot modify in the Stored Procedure Properties notebook appear as read-only fields. For Java and SQL stored procedures, you can change the value of many of these read-only fields by changing the values in the stored procedure source code.

Restriction: For stored procedures written in languages other than Java or SQL, you cannot modify the characteristics of the stored procedures.

For information about creating stored procedures, see Ways to create stored procedures. For information about modifying stored procedure source code, see Modifying existing Java stored procedure source code and Modifying existing SQL stored procedure source code.

RELATED CONCEPTS

What is DB2 Stored Procedure Builder?
How has DB2 Stored Procedure Builder changed the process of creating stored procedures?

RELATED TASKS

Setting environment properties
Setting project properties
Finding more information on DB2 Stored Procedure Builder

Copying and pasting stored procedures to another database

DB2 Stored Procedure Builder makes moving stored procedures from one database to another an easy process with menu-driven copy and paste commands. Copying and pasting stored procedures between databases is useful if you want a stored procedure to reside on multiple databases, or if you need to migrate a stored procedure from a test to a production database.

Restriction: You can only copy and paste stored procedures to other databases that have the same operating system; for example, you can copy a stored procedure from a workstation server to another workstation server. You cannot copy a stored procedure from a workstation server to an OS/390 server.

To copy and paste a stored procedure to another database:

1. Right-click the stored procedure you want to copy and select **Copy Procedure**.
2. **Optional**: Add a database connection by right-clicking the project folder and selecting **Insert Connection**. The new connection appears in the tree.
3. Right-click the **Stored Procedures** folder under the new database connection or an existing connection and select **Paste** or **Paste and Build**. **Paste** puts the stored procedure in the tree. **Paste and Build** puts the stored procedure in the tree and builds the stored procedure to the database.

RELATED CONCEPTS

What is DB2 Stored Procedure Builder?
How has DB2 Stored Procedure Builder changed the process of creating stored procedures?

RELATED TASKS

Managing multiple database connections in a project
Finding more information on DB2 Stored Procedure Builder

Editing generated code

DB2 Stored Procedure Builder includes an editor for editing and manipulating source code. The default editor includes standard cut, copy, and paste capabilities, find and replace, support for menus and key combinations, colored text, and more. You can change the default key behavior of the editor to match vi or Emacs

keyboard commands by using the Environment Properties notebook.
Right-clicking in the editor pane when source code is displayed opens the context menu for the editor.

RELATED CONCEPTS

What is DB2 Stored Procedure Builder?
How has DB2 Stored Procedure Builder changed the process of creating stored procedures?

RELATED TASKS

Modifying existing Java stored procedure code
Modifying existing SQL stored procedure code
Finding more information on DB2 Stored Procedure Builder

Managing multiple database connections in a project

With DB2 Stored Procedure Builder you can maintain connections to multiple databases.

To manage multiple database connections:

If you want to:	Then:
Insert a database connection	Right-click the project folder in the tree and select **Insert Connection**.
Remove a database connection	Right-click a database connection in the tree and select **Delete**. If you have stored procedures that have not been built to the database, you are prompted to delete these stored procedures from the project or cancel the removal of the connection.
Refresh a database connection	Right-click a database connection in the tree and select **Refresh**. You might want to refresh your connection periodically to view changes that you or other users of the database have made. Newly added stored procedures display in the tree when a connection is refreshed.
View the properties of a database connection	Right-click a database connection in the tree and select **Properties**. The Database Connection Properties notebook opens, showing you the properties for the current connection.
Filter stored procedures for a connection	Right-click a database connection in the tree and select **Filter**. The Database Connection Properties notebook opens and displays the Filter page. Enter the search criteria and values on which you want to filter the stored procedures for that connection. The Project windows indicates when a filter is set for a connection.

Add or modify debug records for stored procedures	Right-click a database connection in the tree and select **Debug Records**. The Database Connection notebook opens. The Database Connection notebook allows you to create debug records for all stored procedures on a database without having to rebuild individual stored procedures. **Restriction**: You can *only* add or modify debug records for stored procedures built to workstation databases.

RELATED CONCEPTS

What is DB2 Stored Procedure Builder?
How has DB2 Stored Procedure Builder changed the process of creating stored procedures?
Overview of debugging stored procedures

RELATED TASKS

Managing Stored Procedure Builder projects
Copying and pasting stored procedures to another database
Finding more information on DB2 Stored Procedure Builder

Running stored procedures

DB2 Stored Procedure Builder allows you to run a stored procedure for testing purposes. When you use Stored Procedure Builder to run stored procedures, you can determine whether the build was successful and if a result set exists. Running a stored procedure will help you test stored procedure logic and the accuracy of output arguments and result sets.

To run a stored procedure:

1. Right-click the stored procedure and select **Run**. If you have made changes to the stored procedure since the last build, the Run Confirmation window opens.
2. Select to build the stored procedure to the database before running to reflect the latest changes, or select to run the stored procedure that exists in the database, without the latest changes.
3. Click **OK** after you have made your selection in the Run Confirmation window. If the stored procedure requires user-supplied variables, the Specify Variable Values window opens, listing the name and type of all input parameters.
4. Type a value in the **Value** field. The value must conform to the parameter type specified in the **Type** column. String parameters can be empty or null.
5. If you do not intend to specify an input string and your stored procedure is not written to handle null inputs, deselect the **Treat empty strings as nulls** check box.
6. Click **OK** to run the stored procedure. Input variables display in the Messages pane; output variables and result sets display in the Result pane.

RELATED CONCEPTS

Overview of debugging stored procedures
How has DB2 Stored Procedure Builder changed the process of creating stored procedures?

RELATED TASKS

Modifying existing Java stored procedure source code
Modifying existing SQL stored procedure source code
Finding more information on DB2 Stored Procedure Builder

Dropping stored procedures

DB2 Stored Procedure Builder allows you to drop a stored procedure from a project and a database simultaneously.

To drop a stored procedure:

1. **Optional**: To save the source code from your stored procedure before dropping the stored procedure, click the stored procedure in the tree and select **File —> Save as**.
2. Right-click the stored procedure and select **Drop**. If the stored procedure has already been built to the database, the Drop Item window opens.
3. In the Drop Item window, select whether to drop the stored procedure from the database and the project or just from the database.
4. Click **Yes** to drop the stored procedure.

Requirement: To drop a stored procedure from the database, you must have ownership of the stored procedure and the proper database privileges, which include at least one of the following privileges or authorities:

- The DELETE privilege
- The DROPIN privilege for the schema or all schemas
- SYSADM or SYSCTRL authority

Contact your database administrator for more information.

RELATED CONCEPTS

What is DB2 Stored Procedure Builder?
How has DB2 Stored Procedure Builder changed the process of creating stored procedures?

RELATED TASKS

Copying and pasting stored procedures to another database
Finding more information on DB2 Stored Procedure Builder

Finding more information on DB2 Stored Procedure Builder

After starting DB2 Stored Procedure Builder, you can obtain full procedural information about Stored Procedure Builder by selecting **Help —> General Help** in the Project window. You can also get help for the current window.

To get help for the current window:

- Click the Help button.
- Press F1. In most windows, pressing F1 is the same as clicking **Help**.
- Select a field and press F1 to display the help for the field or control. The wizards and notebooks have more detailed help known as infopops. If the Show pop-up information for interface controls check box in the Environment Properties notebook is selected, you can also see an infopop by holding the mouse cursor over a field or control. The infopops appears within two seconds.

The Stored Procedure Builder help has these features:

- Notebook overviews. A notebook is a window that has multiple pages. You can perform multiple tasks from a notebook. Therefore, when you click Help in a notebook, an overview page appears that shows all the tasks available from the notebook.
- Task steps. Choose from a list of tasks if more than one can be performed from the window.
- Getting started information in the table of contents section "Before you create stored procedures." This section of help provides information that you should read before creating stored procedures.

RELATED CONCEPTS

What is DB2 Stored Procedure Builder?

RELATED TASKS

Managing Stored Procedure Builder projects

Notices

Note to U.S. Government Users Restricted Rights — Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.
IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:
IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:
INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OR CONDITIONS OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.
This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

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.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

Lab Director
IBM Canada Ltd.
1150 Eglinton Avenue East
Toronto, Ontario M3C 1H7
Canada

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.
The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples may include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:
This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM’s application programming interfaces.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:
© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. 1997, 2000. All rights reserved.

Programming interface information

Programming interface information is intended to help you create application software using this program.

General-use programming interfaces allow the customer to write application software that obtain the services of this program's tools.

However, this information may also contain diagnosis, modification, and tuning information. Diagnosis, modification and tuning information is provided to help you debug your application software.

Warning: Do not use this diagnosis, modification, and tuning information as a programming interface because it is subject to change.

Trademarks and service marks

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

- AIX
- AS/400
- DB2
- CICS
- CICS/ESA
- IBM
- IMS
- Language Environment
- MQSeries
- Network Station
- OS/2
- OS/390
- OS/400
- RS/6000
- S/390
- VisualAge
- VTAM
- WebSphere

Lotus, Lotus Notes and Domino are trademarks or registered trademarks of Lotus Development Corporation in the United States, or other countries, or both.

Tivoli Enterprise Console and Tivoli Module Designer are trademarks of Tivoli Systems Inc. in the United States, or other countries, or both.

Encina and DCE Encina Lightweight Client are trademarks of Transarc Corporation in the United States, or other countries, or both.

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

ActiveX, Microsoft, SourceSafe, Visual C++, Visual SourceSafe, Windows, Windows NT, Win32, Win32s and the Windows logo are trademarks or registered trademarks of Microsoft Corporation in the United States, or other countries, or both.

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.

Intel and Pentium are trademarks of Intel Corporation in the United States, or other countries, or both.

Other company, product, and service names, which may be denoted by a double asterisk(**), may be trademarks or service marks of others.