

# IBM Data Studio Development Component

Tony Leung  
leungtk@us.ibm.com

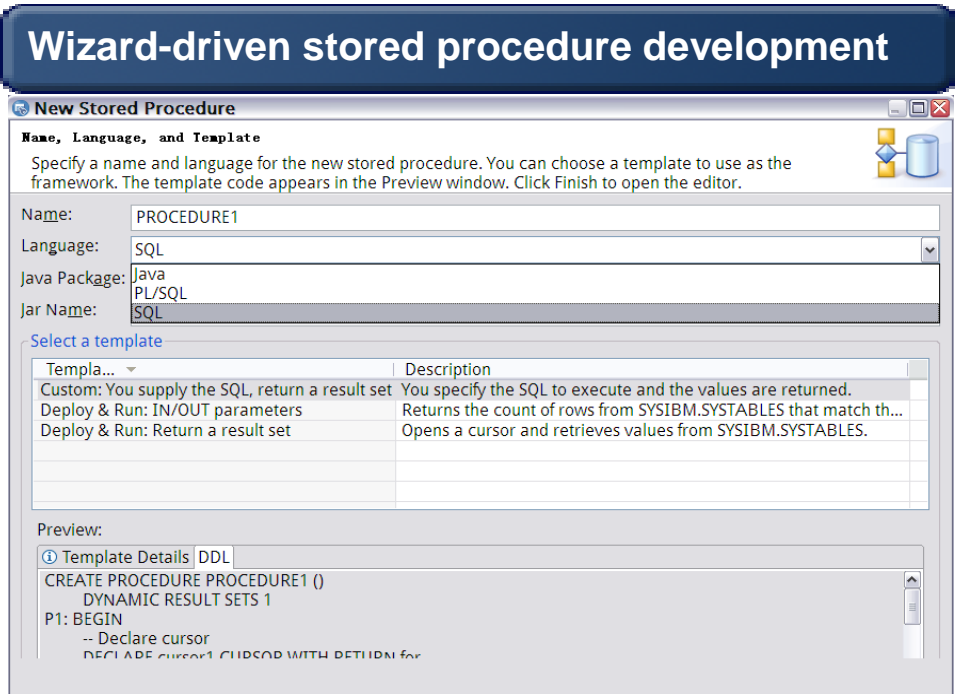
# Routine Development

- **Overview of IBM Data Studio Routine Development**
  
- **Routine Development lifecycle**
  - Create
  - Edit
  - Deploy
  - Run
  - Profiling
  - debug

## Overview of IBM Data Studio Routine Development

# Stored Procedures Development

- **Develop, debug and deploy**
  - SQL stored procedures
  - Java Stored Procedures in JDBC
  - Java Stored Procedures in SQLJ
  - Oracle PLSQL Stored Procedures
- **Wizard-driven**
  - Step by step creation
  - Enable debugging
  - Integrated deploy



## Using IBM Development Studio for Routines

- **Creating Database Connection**
- **Creating Database Development Project**
- **Creating a new SQL stored procedure from scratch**
- **Deploying a SQL stored procedure**
- **Running SQL stored procedure**
- **Working with an existing SQL stored procedure**
- **Managing stored procedure template**

# Creating Database Connection

- Automatically loading local database connection
- Remote database
  - Select driver type and connection parameters
  - Select driver

### Select driver type and connection parameters

**Connection Parameters**

Enter a user name.

**Connection identification**

Use default naming convention  
 Connection Name: SAMPLE1

Local Connection Config...  
 Select a database manager:  
 DB2 for i5/OS  
 DB2 for Linux, UNIX, and Windows  
 DB2 for z/OS  
 Derby  
 Generic JDBC  
 Informix  
 Oracle

JDBC driver: IBM 数据服务器 JDBC 和 SQL 驱动程序 (JDBC 4.0)

**Properties**

General Tracing Optional

Database: SAMPLE  
 Host: localhost  
 Port number: 50000

Use client authentication  
 User name:  
 Password:  
 Save password  
 Default schema:  
 Connection URL: jdbc:db2://localhost:50000/SAMPLE:retrieveMessagesFromServerOnGetMessage=true;

Test Connection

< Back Next > Finish Cancel

### Select driver

**Specify JAR List**

Specify the file path to the JAR or .zip files that are required by the JDBC driver. Some drivers are not included with this product and must be

Driver files:

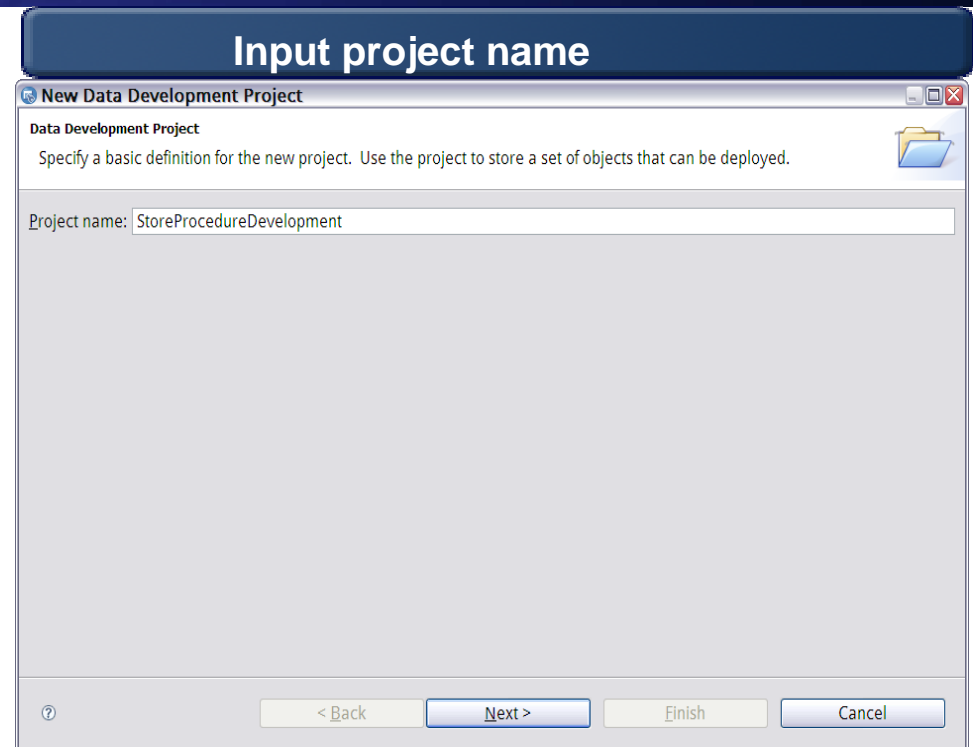
D:\IBM\optim\SDPShared\plugins\com.ibm.datatools.db2\_2.1.202  
 D:\IBM\optim\SDPShared\plugins\com.ibm.datatools.db2\_2.1.202

Add JAR/.zip...  
 Edit JAR/.zip...  
 Remove JAR/.zip  
 Clear All

OK Cancel

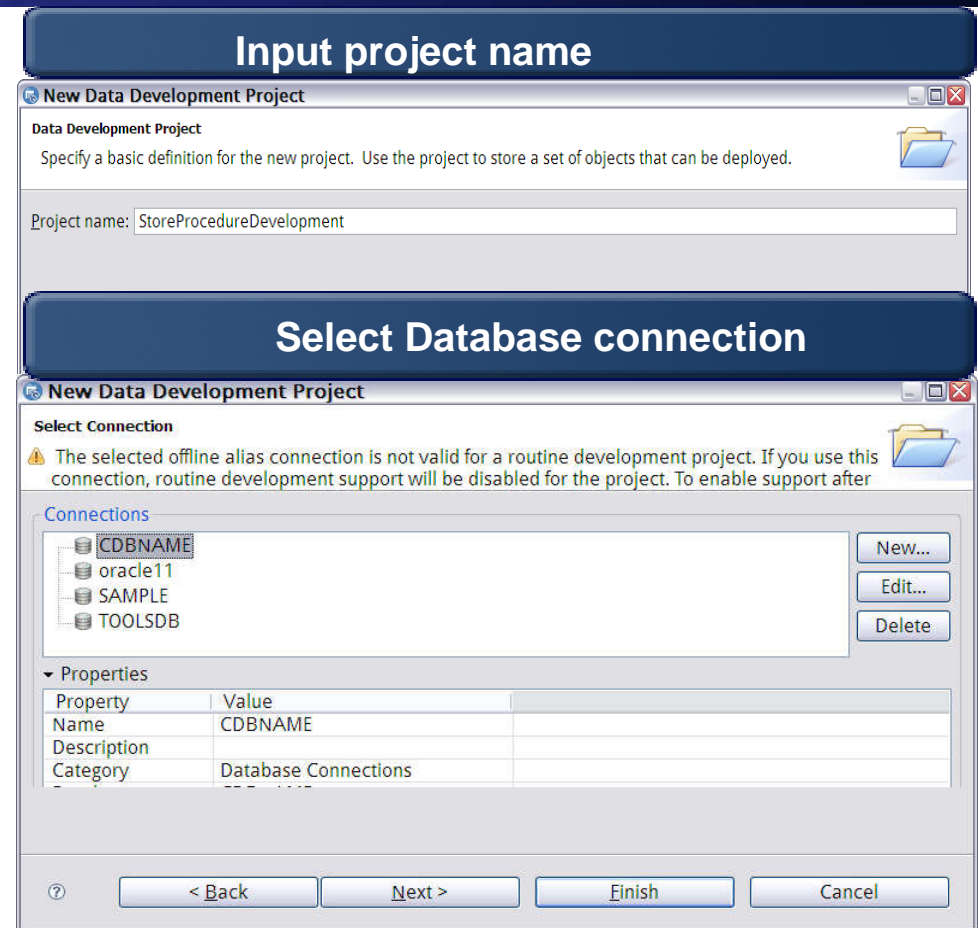
# Creating Database Development Project

- Input project name



# Creating Database Development Project

- Input project name
- Select database connection





# Creating Database Development Project

- Input project name
- Select database connection
- Customize the project properties

The image displays three sequential screenshots of the IBM Data Development Project wizard, each with a dark blue header bar indicating the current step.

**Step 1: Input project name**  
The window title is "New Data Development Project". The main heading is "Data Development Project". Below it, a text box contains the project name "StoreProcedureDevelopment".

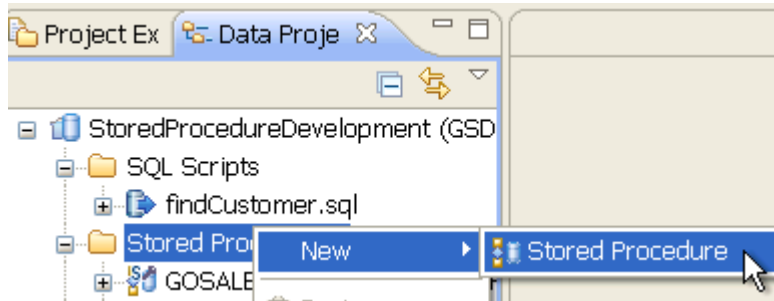
**Step 2: Select Database connection**  
The window title is "New Data Development Project". The main heading is "Select Connection". A warning icon and text state: "The selected offline alias connection is not valid for a routine development project. If you use this connection, routine development support will be disabled for the project. To enable support after".

**Step 3: Customize the project properties**  
The window title is "New Data Development Project". The main heading is "Default Application Process Settings". Below it, a text box contains the default path: "SYSIBM,SYSFUN,SYSPROC,SYSIBMADM,ADMINISTRATOR".

At the bottom of the third screenshot, there are four buttons: "?", "< Back", "Next >", "Finish", and "Cancel".

# Stored Procedure – Create New

## 1. New stored procedure



# Stored Procedure – Create New

## 1. New stored procedure

**2. Select routine language**

**Name, Language, and Template**

Specify a name and language for the new stored procedure. You can choose a template to use as the framework. The template code appears in the Preview window. Click Finish to open the editor.

Name: PROCEDURE1

Language: SQL

Java Package: Java

Jar Name: PL/SQL

Select a template

Templa...	Description
Custom: You supply the SQL, return a result set	You specify the SQL to execute and the values are returned.
Deploy & Run: IN/OUT parameters	Returns the count of rows from SYSIBM.SYSTABLES that match th...
Deploy & Run: Return a result set	Opens a cursor and retrieves values from SYSIBM.SYSTABLES.

Preview:

Template Details DDL

```
CREATE PROCEDURE PROCEDURE1 ()
  DYNAMIC RESULT SETS 1
P1: BEGIN
  -- Declare cursor
  DECLARE cursor1 CURSOR WITH RETURN FOR
```

# Stored Procedure – Create New

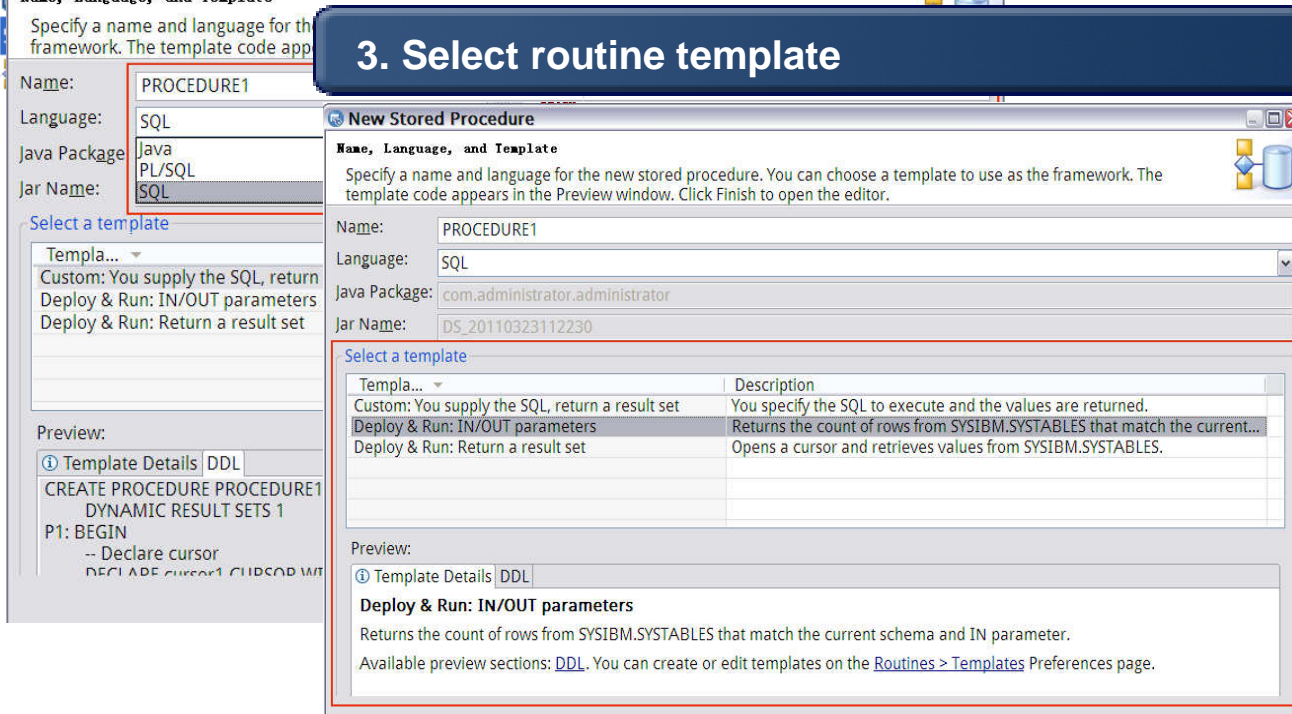
## 1. New stored procedure



## 2. Select routine language



## 3. Select routine template



# Stored Procedure – Edit / Customize

- Context sensitive editor
- Content assist for SQL statements
- Formatted SQL
- Optional validation

```

CREATE PROCEDURE USER01.GETCUSTOMERORDERS ( )
VERSION VERSION1
ISOLATION LEVEL CS
RESULT SETS 1
LANGUAGE SQL

-----
-- SQL Stored Procedure
-----

P1: BEGIN
-- Declare cursor
DECLARE cursor1 CURSOR WITH RETURN FOR
SELECT GOSALEST.CUST_ORDER_DETAIL.CUST_ORDER_DETAIL_CODE, GOSALEST.CUST_ORDE
GOSALEST.CUST_ORDER_DETAIL.CUST_SHIP_DATE, GOSALEST.CUST_ORDER_DETAIL.CUST
FROM
GOSALEST.CUST_ORDER_DETAIL J
WHERE GOSALEST.CUST_ORDER_HEADER.

-- Cursor left open for client applicati
OPEN cursor1;
END P1

```

Content Assist	Ctrl+Space
Content Tip	Ctrl+Shift+Space
Format SQL	Ctrl+Shift+F
Toggle Comment	Ctrl+ /
<input checked="" type="checkbox"/> Validate Statement Syntax	
Deploy Source	
Deploy...	
Start Tuning...	
Find reference in SQL Outline	
Generate pureQuery Code...	
Open Visual Explain	

```

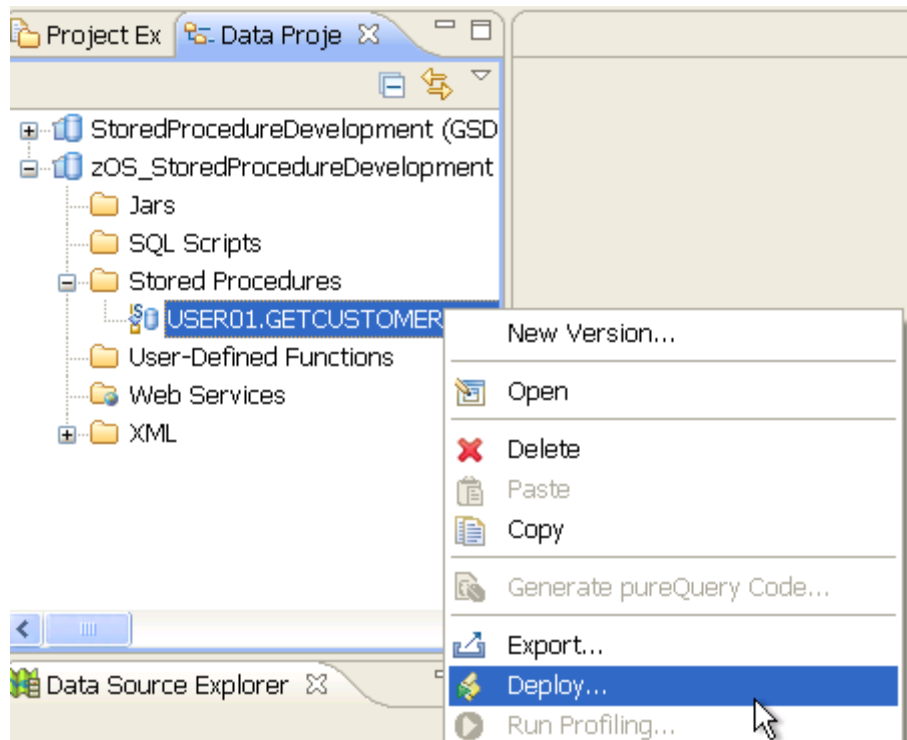
P1: BEGIN
-- Declare cursor
DECLARE cursor1 CURSOR WITH RETURN FOR
SELECT GOSALEST.CUST_ORDER_DETAIL.CUST_ORDER_DETAIL_CODE,
GOSALEST.CUST_ORDER_DETAIL.CUST_SHIP_DATE, GOSALEST.CU
FROM
GOSALEST.CUST_OR JOIN GOSALEST.CUST_ORDER_HEADER
WHERE GOSALEST
-- Cursor left open f
OPEN cursor1;

```

	CUST_ORDER_DETAIL
	CUST_ORDER_HEADER
	CUST_ORDER_STATUS

# Stored Procedure – Deployment

## 1. Select stored procedure for deployment



# Stored Procedure – Deployment

## 2. Customize deployment options

### Deploy Routines

#### Deploy Options

Database: You must specify a value.

**Target database**

Use current database

Use different database

Database:  Connection...

**Target schema and default path for deploying an unqualified routine**

Target schema:

Default path:

**Duplicate handling**

Drop duplicates

Treat duplicates as errors

Ignore duplicates and continue to the next routine

Deploy by building the source

Deploy using binaries if available in the database

Deploy source to the database

**Specify target database**

**Specify target schema**

**Specify default path**

**Specify how to handle duplicates**

## Enhanced Duplicate Handling in DB2 for Z

### Duplicate handling

- Alter duplicates
- Drop duplicates
- Treat duplicates as errors
- Ignore duplicates and continue to the next routine

**Drop duplicates used to generate Alter procedure statement.  
We will separate the two options to handle both Alter and Drop cases**



# Stored Procedure – Deployment

## 3. . Customize compiled options

**Deploy Routines**

**Routine Options**  
Specify routine options.

- SQL Stored Procedure: 
  - USER01.GETCUST:
- WLM environment: WLMENV1
- Procedure options: ISOLATION LEVEL CS
- Build owner: USER10
- ASU time limit: 0
- Enable debugging
- Apply Folder Settings

Options for z/OS

**Deploy Routines**

**Routine Options**  
Specify routine options.

- SQL Stored Procedure: 
  - GOSALESCCT.GET:
- Precompile options:
- Enable debugging
- Apply Folder Settings

Options for LUW

## Stored Procedure - Run

- Run executes a call of the stored procedure
- Asks for input parameters values
- View result set
- Customize run settings

**Run Settings - GET\_CUSTOMER\_NAME**

Before Run | Parameter Values | After Run | Options

Specify SQL statements to run before you run or debug the stored procedure. Separate statements with semi-colons (;).

```
SET CURRENT SCHEMA = 'GOSALESCT'
```

Create SQL...  
Open Script...  
Save Script...

**Specify Parameter Values - GET\_CUSTOM...**

Specify the parameter values that you want to use for running or debugging the stored procedure. You can set selected strings or user-defined types to null.

Name	Type	Value
CUSTOMERID	INTEGER	126920

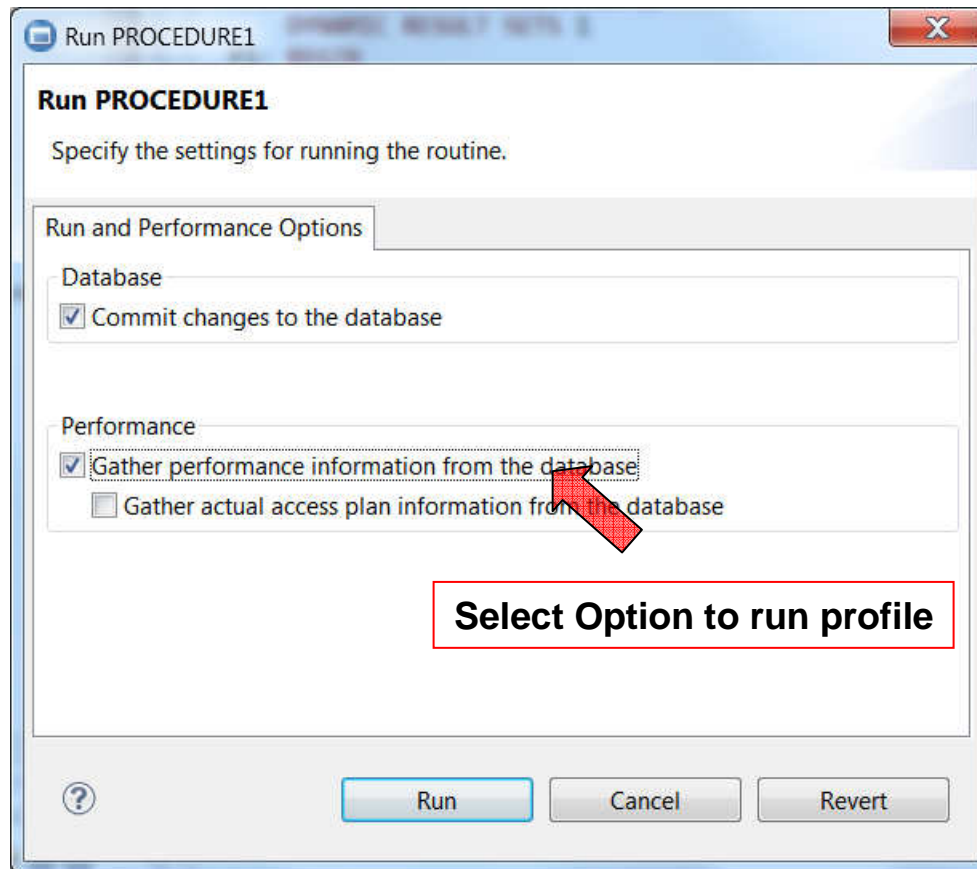
OK | Cancel | Reset

Workload List | Monitor List | SQL Results

Name	Type	Data type	Value	Value (OUT)
CUSTO...	INPUT	INTEGER	126920	
FIRST_...	OUTPUT	VARCHAR		Vikentiy
LAST_N...	OUTPUT	VARCHAR		Prokofiyov
PHONE_...	OUTPUT	VARCHAR		+7 8455 555 6045

# DB2 for LUW stored procedures – run profiling

Invoke Run Procedure



# Profiling Data for Stored Procedures

Status Profiling Data

Profile data for DB2ADMIN.PROCEDURE1

Lin...	SQL	Number of Executions	Elapsed Time	User CPU Time	Rows Returned
1	CREATE PROCEDURE PROCEDURE1 ()				
2	DYNAMIC RESULT SETS 1				
3	P1: BEGIN				
4	-- Declare cursor				
5	DECLARE cursor1 CURSOR WITH RETUR...	1	61952000	-	796
6	-- #####...				
7	-- # Replace the SQL statement with your...				
8	-- # Note: Be sure to end statements wit...				
9	-- #				
10	-- # The example SQL statement SELECT ...				
11	-- # returns all names from SYSIBM.SYST...				

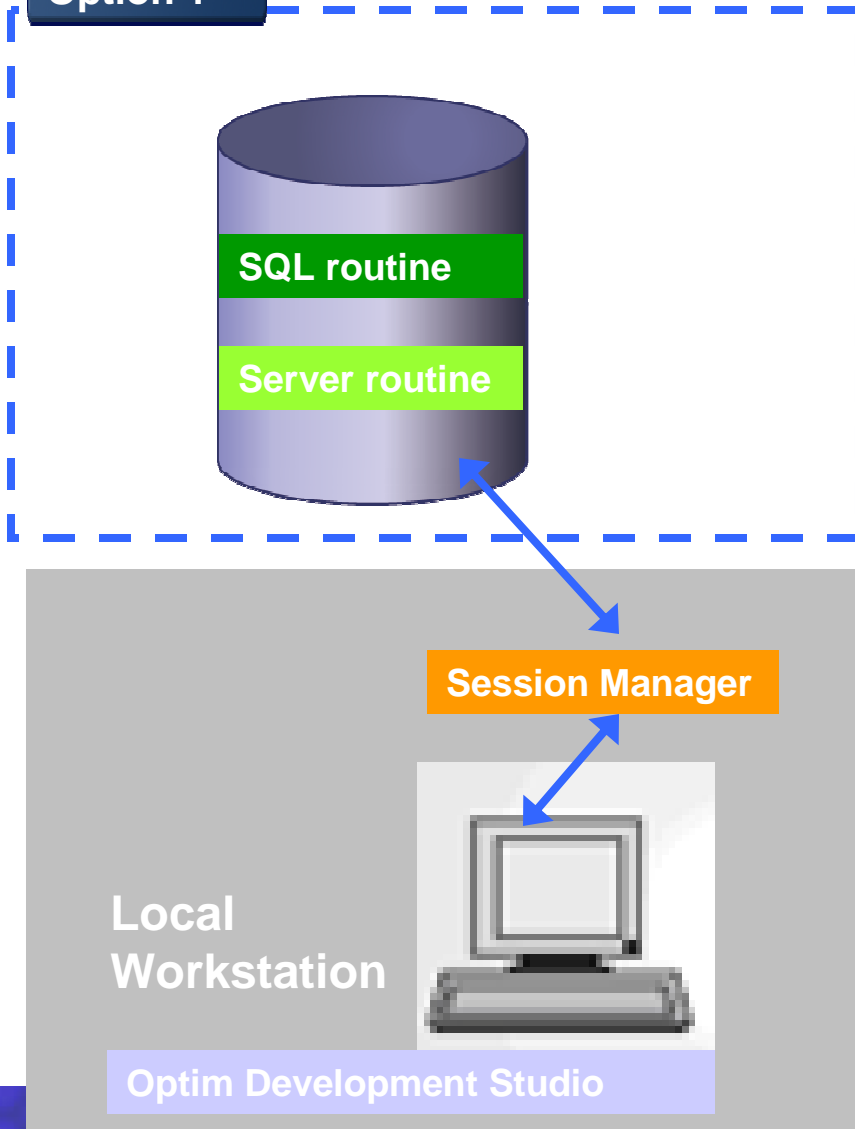
# Debugging Stored Procedures

# DB2 Unified Debugger – Server Side Debug Support

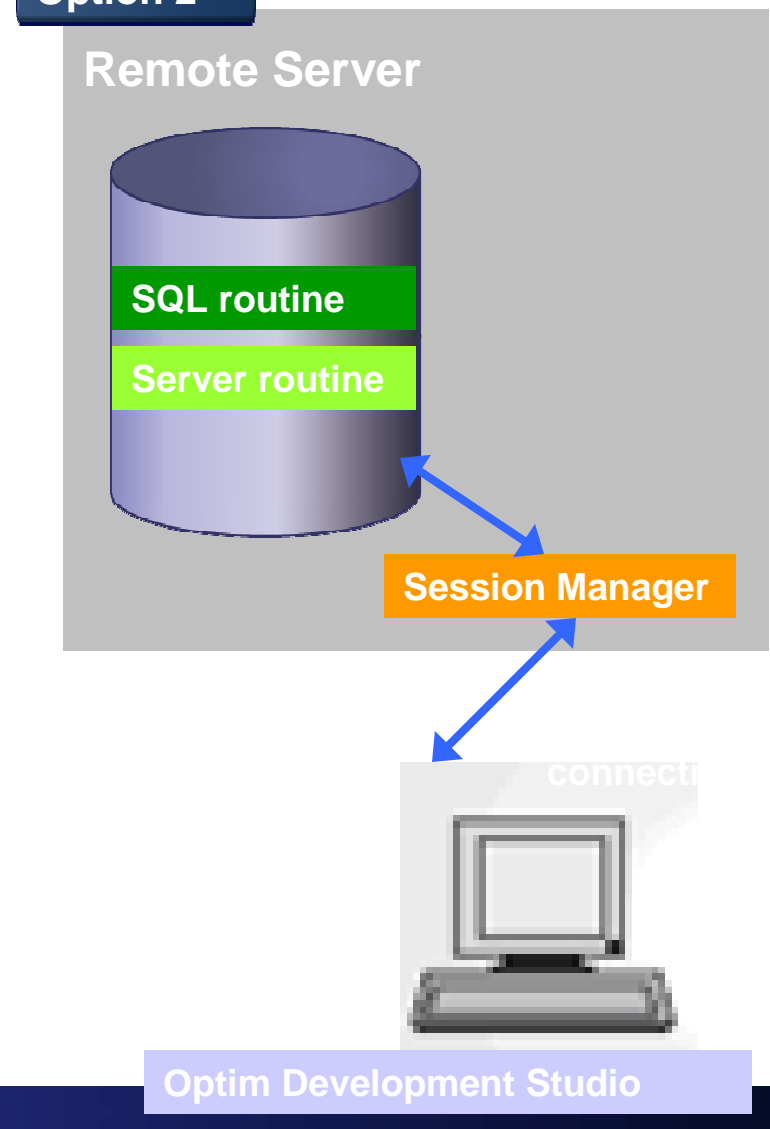
- **One debugger for multiple platforms**
- **Allows users to remotely debug server side SQL and Java stored procedures**
- **Debug Target servers**
  - DB2 for LUW V9,9.7,10,10.5 and DB2 for LUW V8 FP15
  - DB2 for zSeries V9~V11,
  - DB2 for zSeries V8 (with PTF number UK25860 applied)
  - DB2 for iSeries V5R4, V6.1
- **Debug native SQL stored procedures against a DB2 for z/OS V9 server**

# Unified Debugger framework

Option 1



Option 2



# Unified Debugger – Session Manager

- **Session Manager can reside in server or a client machine**

Note: these settings will take effect at the start of the next session.

Number of seconds of inactivity before the procedure runs to completion:

Number of characters per line in Variables View Details Pane:

Diagnostic error trace level on database server:

**Routine Debug Session Manager Location**

Use the built-in session manager:

Port:

Server port use for debugging multiple Java routines when applicable

Port:

Use the session manager on each connected server:

Port:

Session manager timeout in minutes:

Use an already running session manager:

Host:

Port:

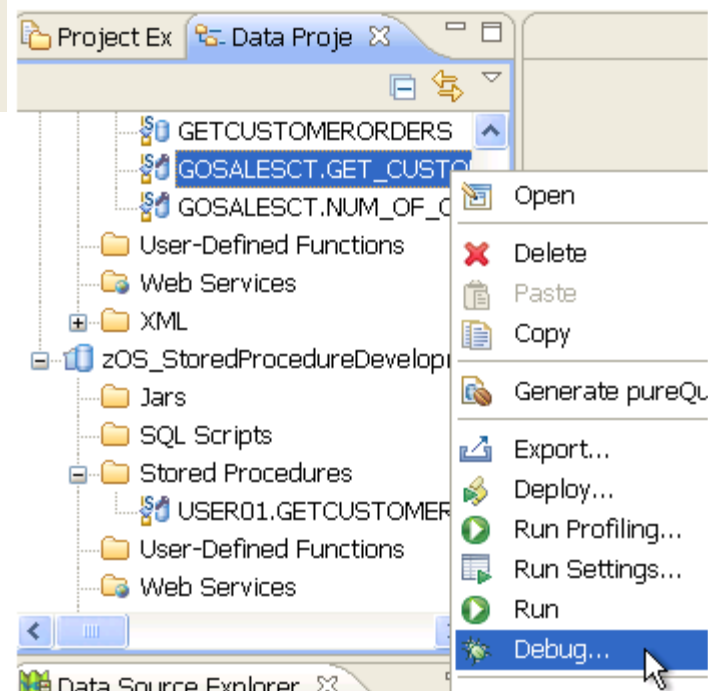


# Preparing for Debugging

## 1. Deploy stored procedure with "Enable debugging"



## 2. Select stored procedure for Debug...



# Debug Perspective

The screenshot shows the IBM Debug Perspective interface. The top toolbar contains several icons with callouts: **terminate**, **resume**, **step into**, **step over**, and **step return**. The main editor displays a SQL script with a callout for **current location** pointing to the current execution point. A **Valid break point location** callout points to a green icon in the left margin, and a **break point** callout points to a yellow icon. The **Variables** panel on the right shows a table of variables with a callout for **current value**. A right-click context menu is open over the variables panel, with a callout for **right click change value** pointing to the **Change Value...** option.

Name	Value
Diagnostic Information	
NUM_LINES	null
TBLROUTINE	
SPECIFIC	
ROUTINES	

```
Script12.sql V9SPB003 *Script18.sql PROCEDURE7 PR
IF(TBLROUTINE=1) THEN
  SELECT ROUTINENAME, TEXT, ROUTINEID
  INTO ROUTINENAME, TEXT, ROUTINEID
  FROM REMOTE ROUTINES
  WHERE ROUTINESCHEMA = SHRED.ROUTINESCHEMA
  AND SPECIFICNAME = SHRED.SPECIFICNAME;
END IF;
IF SQLCODE = 100 THEN RETURN 1; END IF;
SET NEXT_CR = 1;
SET LINE_NUMBER = 1;
WHILE NEXT_CR <> 0 DO
  SELECT POSSTR(TEXT, CHR(10)), SUBSTR(TEXT, 1, ABS(POSSTR(TEXT, CHR(10)) - 1))
  INTO NEXT_CR, LINE
  FROM TABLE(VALUE TEXT) AS T(TEXT);
  IF NEXT_CR <> 0 THEN
    INSERT INTO ROUTINE TEXT
```

# Debug Perspective— Breakpoints

Clicking resume will stop at next breakpoint

ProcedureDevelopment/GET\_CUSTOMER\_NAME.spxmi - IBM Optim Development Studio

Project Script Run Data Window Help

Debug Data

Variables Breakpoints

- GET\_CUSTOMER\_NAME:Line 11
- GET\_CUSTOMER\_NAME:Line 15

StoredProcedureDevelopment.GSDB [DB2 SQL and Java Stored Procedure Debugger]

- jdbc:db2://localhost:50000/GSDB:retrieveMessagesFromServerOnGetMessage=true
- GET\_CUSTOMER\_NAME (Suspended (breakpoint at line 11))
- GET\_CUSTOMER\_NAME [Line: 11]

Debugger stops at line 11

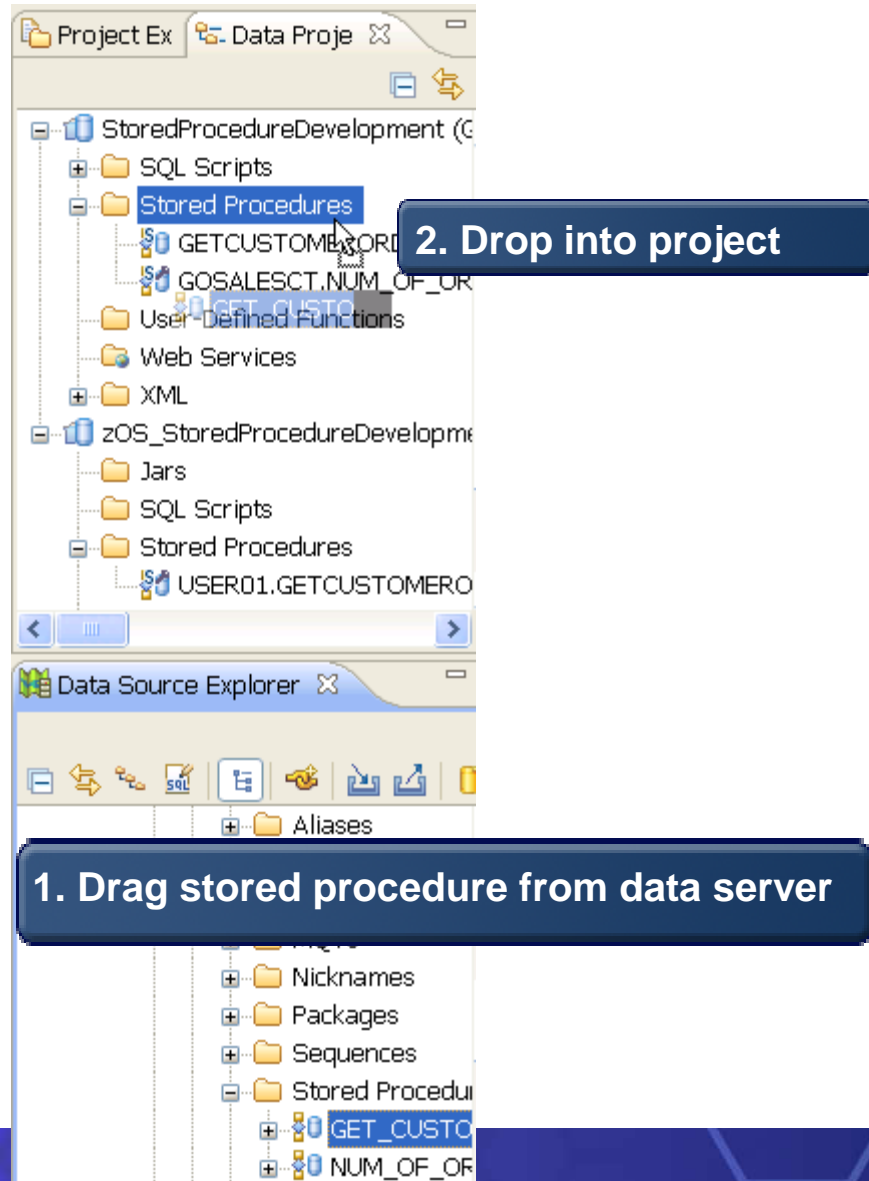
```
CREATE PROCEDURE GOSALESCT.GET_CUSTOMER_NAME (IN CUSTOMERID INTEGER,
OUT FIRST_NAME VARCHAR(128),
OUT LAST_NAME VARCHAR(128),
OUT PHONE_NUMBER VARCHAR(128))
SPECIFIC GOSALESCT.GET_CUSTOMER_NAME
BEGIN
  SELECT CUST_FIRST_NAME INTO FIRST_NAME
  FROM GOSALESCT.CUST_CUSTOMER
  WHERE CUST_CODE = CUSTOMERID;
  SELECT CUST_LAST_NAME INTO LAST_NAME
```

# Objects Supported for debugging

- **SQL/Java Stored Procedures and User-defined Functions**
  - DB2 for LUW
  - DB2 for I
  - DB2 for zOS
  - Infomix
  
- **PL/SQL**
  - DB2 for LUW V9.7
  
- **Anonymous block and nested routines**
  - DB2 for LUW V10fp2
  
- **Trigger debugger**
  - DB2 for LUW V10fp3

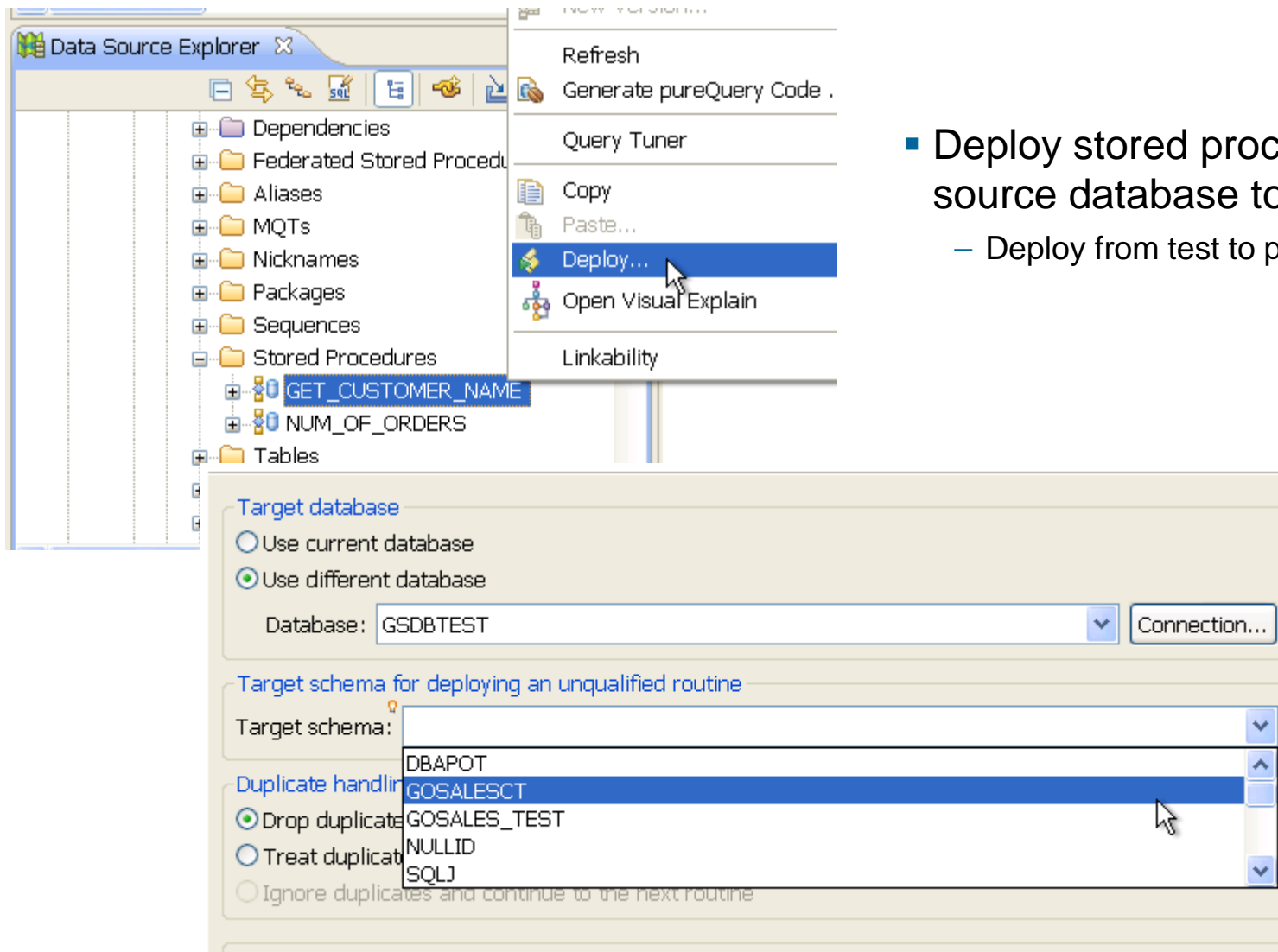
## **Working with existing stored procedures**

# Work Existing Stored Procedures



- Work with stored procedures that have been already deployed
  - Make a local copy by drag and drop or copy and paste
  - Modify / work with local copy
  - Deploy onto server
  - Debug stored procedure

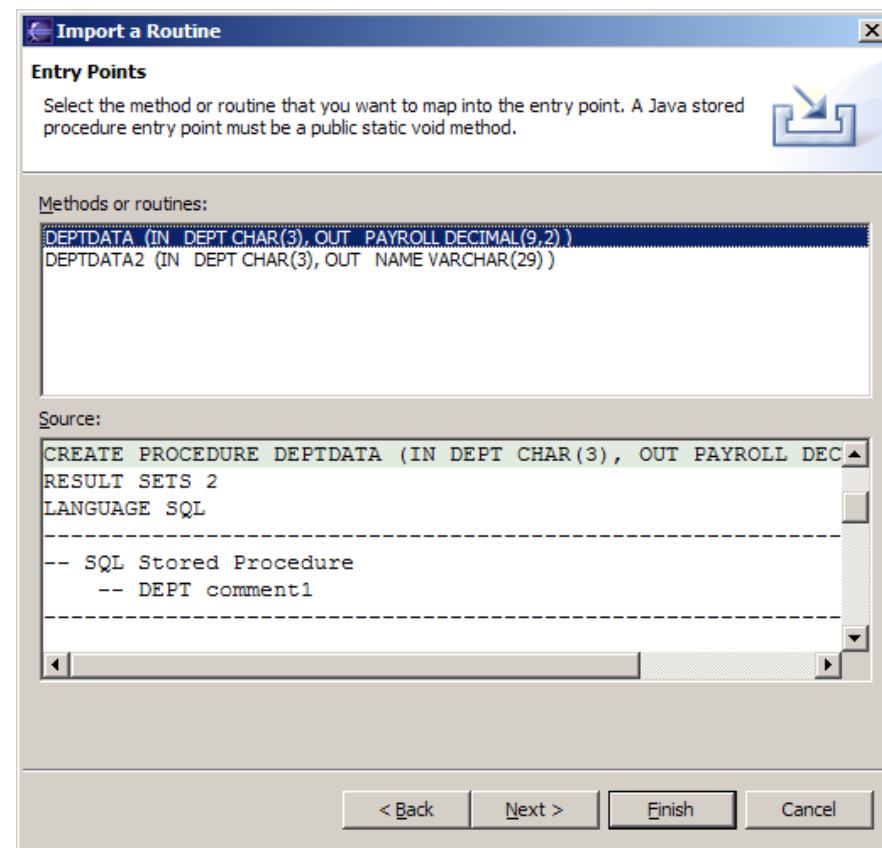
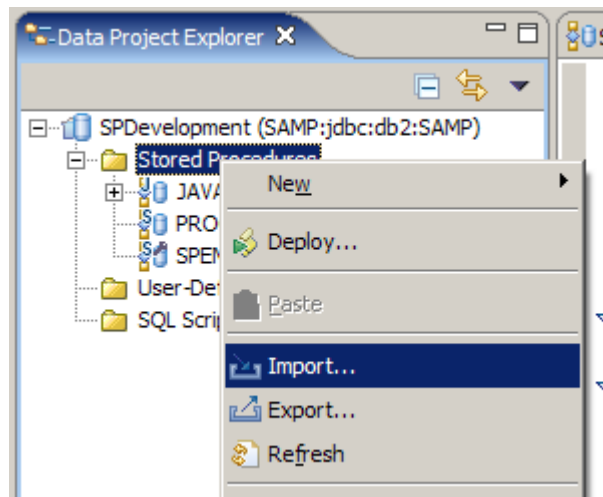
# Deploy stored procedures from server to server



The screenshot shows the IBM Data Source Explorer interface. The left pane displays a tree view of database objects, including 'Stored Procedures'. The 'GET\_CUSTOMER\_NAME' procedure is selected. A context menu is open over it, with the 'Deploy...' option highlighted. Below the context menu, a dialog box is shown for configuring the deployment. The 'Target database' section has 'Use different database' selected, with 'GSDBTEST' chosen in the dropdown. The 'Target schema for deploying an unqualified routine' section has a dropdown menu open, showing a list of schemas: DBAPOT, GOSALESCT (highlighted), GOSALES\_TEST, NULLID, and SQLJ. The 'Duplicate handling' section has 'Drop duplicate' selected.

- Deploy stored procedures from source database to other servers
  - Deploy from test to production

# Stored procedure – import from file system





## Stored procedure - export and deploy

- **Export to file system**
  - Generates DB2 script file, for Java an ANT script and a properties file for customization
  - Allows deployment of the stored procedure independent of the IDE
- **Binary Deploy for SQL and Java stored procedures**
  - Deploy using binaries only to a target server other than the current server
  - Optionally include the source when deploying (not used to build)
- **Batch Deploy for Java routines**
  - Properties file for customized settings
    - Target database, username and password, procedure options like WLM environment
    - To deploy run from command prompt: **ant -buildfile [filename].xml**
- **Batch deploy for SQL routines:**
  - Customize the DB2 script file directly

# Customize Routine Template

**Specify Template Library**

**Review Template**

**Templates**

Master Template  
Specify a file to use as the master template:  
File:     
 Load at Startup  
**Note:** Master templates take precedence and will replace any local instance of the template.

Create, edit, or remove templates:

Name	Context	Description	Auto In...
<input checked="" type="checkbox"/> Deplo...	oracle_package_plsql	Sample procedur...	
<input checked="" type="checkbox"/> Deplo...	informix_sp_spl	Returns the cou...	
<input checked="" type="checkbox"/> Deplo...	db2i5_sp_sql	Returns the cou...	
<input checked="" type="checkbox"/> Deplo...	db2luw_sp_sql	Returns the cou...	
<input checked="" type="checkbox"/> Deplo...	db2luw_sp_plsql	Opens a cursor ...	
<input checked="" type="checkbox"/> Deplo...	informix_udf_spl	Opens a cursor ...	
<input checked="" type="checkbox"/> Deplo...	db2i5_sp_sql	Opens a cursor ...	
<input checked="" type="checkbox"/> Deplo...	oracle_sp_plsql	Opens a cursor ...	
<input checked="" type="checkbox"/> Deplo...	db2luw_sp_sql	Opens a cursor ...	
<input checked="" type="checkbox"/> Deplo...	db2luw_udf_plsql	Opens a cursor ...	
<input checked="" type="checkbox"/> Deplo...	db2luw_package_plsql	Sample procedur...	
<input checked="" type="checkbox"/> Deplo...	oracle_package_plsql	Sample procedur...	
<input checked="" type="checkbox"/> Deplo...	informix_sp_spl	Returns the cou...	
<input checked="" type="checkbox"/> Deplo...	oracle_udf_plsql	Returns the cou...	
<input checked="" type="checkbox"/> Deplo...	db2zos_udf_sql	Returns the curr...	
<input checked="" type="checkbox"/> Deplo...	informix_udf_spl	Returns rows fr...	

Buttons: New..., Edit..., Remove, Restore Removed, Revert to Default, Import...

Preview:  
CREATE PROCEDURE \${name} ()  
RESULT SETS 1  
LANGUAGE SQL  
P1: BEGIN  
-- #####  
-- # Returns all tables created by \${schemaName}  
-- #####  
-- Declare cursor  
DECLARE cursor 1 CURSOR WITH RETURN FOR

Buttons: Restore Defaults, Apply, OK, Cancel

# Compare Simplification

# Data Studio 3.2

Compare DEMOPROC.spxmi Current and Local Revision

Text Compare

Local: DEMOPROC.spxmi	Local history: DEMOPROC.spxmi Sep 13, 2013 11:02:21 AM
<pre> xmlns:xsi="http://www.w3.org/2001/XMLSchema- xmlns:DB2Model="http://com.ibm/db/models/db name="DEMOPROC" specificName="" language="SQL" sqlDataAccess="MODIFIES_SQL_DATA" lastAlteredTS="" security="SECURITY DB2" maxResultSets="1" implicitSchema="true" changeState="2"&gt; &lt;source xsi:type="DB2Model:DB2Source"&gt; &lt;clearBody&gt;&lt;![CDATA[CREATE PROCEDURE DemoProc DYNAMIC RESULT SETS 1 P1: BEGIN -- Declare cursor DECLARE cursor1 CURSOR WITH RETURN for -- ##### -- # Replace the SQL statement with your sta -- # Note: Be sure to end statements with -- # -- # The example SQL statement SELECT NAME f -- # returns all names from SYSIBM.SYSTABLES -- ##### SELECT NAME FROM SYSIBM.SYSVIEWS;  -- Cursor left open for client application OPEN cursor1; END P1]]&gt;&lt;/clearBody&gt; &lt;/source&gt; </pre>	<pre> xmi:version="2.0" language="SQL" sqlDataAccess="MODIFIES_SQL_DATA" security="SECURITY DB2" maxResultSets="1" implicitSchema="true" changeState="2"&gt; &lt;source xsi:type="DB2Model:DB2Source"&gt; &lt;clearBody&gt;&lt;![CDATA[CREATE PROCEDURE Dem DYNAMIC RESULT SETS 1 P1: BEGIN -- Declare cursor DECLARE cursor1 CURSOR WITH RETURN for -- ##### -- # Replace the SQL statement with your -- # Note: Be sure to end statements wi -- # -- # The example SQL statement SELECT NA -- # returns all names from SYSIBM.SYSTA -- ##### SELECT NAME FROM SYSIBM.SYSTABLES;  -- Cursor left open for client applicati OPEN cursor1; END P1]]&gt;&lt;/clearBody&gt; &lt;/source&gt; </pre>

**XML attributes show up in Routine Version Compare**

# Data Studio 4.1

Compare DEMOPROC.spsql Current and Local Revision

SQL Routine Compare

Local: DEMOPROC.spsql	Local history: DEMOPROC.spsql Sep 13, 2013 10:58:01 AM
<pre>CREATE PROCEDURE DemoProc ()   DYNAMIC RESULT SETS 1 P1: BEGIN   -- Declare cursor   DECLARE cursor1 CURSOR WITH RETURN for   -- #####   -- # Replace the SQL statement with your   -- # Note: Be sure to end statements with   -- #   -- # The example SQL statement SELECT NAME   -- # returns all names from SYSIBM.SYSTAB   -- #####   SELECT NAME FROM SYSIBM.VIEWS;    -- Cursor left open for client application   OPEN cursor1; END P1</pre>	<pre>CREATE PROCEDURE DemoProc ()   DYNAMIC RESULT SETS 1 P1: BEGIN   -- Declare cursor   DECLARE cursor1 CURSOR WITH RETURN for   -- #####   -- # Replace the SQL statement with your   -- # Note: Be sure to end statements   -- #   -- # The example SQL statement SELECT   -- # returns all names from SYSIBM.SYS   -- #####   SELECT NAME FROM SYSIBM.SYSTABLES;    -- Cursor left open for client applica   OPEN cursor1; END P1</pre>

Much easier to spot the differences

Thank You