

# IBM DB2 for i: Code example

## DB2 Example: SQL StoredProcedure with Result Sets

---

### Section I - The Stored Procedure.

DB2 for i supports two methods for returning stored procedure result sets. The stored procedure can use the standard SQL as shown in the first example or non-standard (proprietary) SQL as shown in the second example. The easiest ways to create the stored procedure would be to use the SQL Script Center in Operations Navigator or to move this code into a source file member and use the RUNSQLSTM CL command to execute this SQL statement.

#### Standard SQL:

```
CREATE PROCEDURE SPLIB/GETNAMESET ( IN CNUM INTEGER )
  RESULT SET 1
  LANGUAGE SQL
  BEGIN

  --Take the inputted customer number, CNUM and
  --return the record(s) for that customer from the
  --customer table via SQL result set(s)
  DECLARE C1 CURSOR WITH RETURN FOR
    SELECT * FROM TESTLIB/CUSTTABLE WHERE CUSTNUM = CNUM;

  OPEN C1;

END;
```

#### Proprietary SQL:

```
CREATE PROCEDURE SPLIB/GETNAMESET ( IN CNUM INTEGER )
  RESULT SET 1
  LANGUAGE SQL
  BEGIN

  --Take the inputted customer number, CNUM and
  --return the record(s) for that customer from the
  --customer table via SQL result set(s)
  DECLARE C1 CURSOR FOR
    SELECT * FROM TESTLIB/CUSTTABLE WHERE CUSTNUM = CNUM;

  OPEN C1;

  SET RESULT SETS CURSOR C1;
END;
```

---

### Section II - A Visual Basic program for invoking the stored procedure

The following section contains VisualBasic program code samples that show how to invoke a DB2 for i Stored Procedure that returns its output via an SQL result set. Sample code showing how to access the returned result set is also

included.

The VB program consists of a form with two input boxes and a command button to call the procedure.

Text2 is the control that the user enters the customer number into.

Text1 is the control that the customer name is returned to.

Command1 is the button to pass the parameter and call the procedure - and handle the results.

=====

VB Declarations:

```
Dim env1 As rdoEnvironment
```

```
Dim conn1 As rdoConnection
```

```
Dim strsql As String
```

```
Dim ps As rdoPreparedStatement
```

```
Dim rs As rdoResultset
```

Here is the Form\_Load Event Code:

'=====

```
rdoEngine.rdoDefaultCursorDriver = rdUseOdbc
```

```
Set env1 = rdoEngine.rdoCreateEnvironment(" ", " ", " ")
```

```
strsql = "DSN=MLRDEV;UID=SHUBA;PWD=Password;"
```

```
Set conn1 = env1.OpenConnection(" ", rdDriverNoPrompt, False,  
strsql)
```

```
conn1.QueryTimeout = 3600
```

'=====

The user enters a customer number into Text2, and the following code is run when the command button is clicked:

'=====

```
strsql = "call SPLIB.GETNAMESET(?)"
```

```
Set ps = conn1.CreatePreparedStatement("NAMETEST", strsql)
```

```
ps(0).Direction = rdParamInput
```

Chat Now

```
ps(0).Value = CInt(Text2.Text)
```

```
Set rs = ps.OpenResultset(rdOpenStatic)
```

```
If rs.RowCount = 0 Then
```

```
    Text1.Text = "Not Found"
```

```
Else
```

```
    Text1.Text = rs!CNAME
```

```
End If
```

```
Text2.SelStart = 0
```

```
Text2.SelLength = Len(Text2.Text)
```

```
ps.Close
```

```
'=====
```

---