

**F03 DB2 Development Add-Ins - Next Generation Visual Studio Tools for zOS, UNIX, and Windows**  
*Abdul Al-Azzawe, DB2 Development Center and SQL Debugger Architect, IBM*

The latest DB2 UDB release includes a set of fully integrated Development Add-Ins for Microsoft Visual Studio 6.0 (Visual Basic, Visual C++, and Visual InterDev) supporting the entire family of DB2 servers, allowing for rapid iterative development of server-side stored procedures and UDFs, as well as client side ADO code generation and integration with Visual Source Safe. In this session, you will learn how to use Visual Studio and the DB2 Development Add-Ins to develop SQL and Java stored procedures and UDFs, MQSeries table UDFs, OLE DB table UDFs, XML table UDFs, edit and debug your code, check in your DB2 code into Visual Source Safe, access the DB2 catalogues, and generate client side DB2 ADO Connections and Commands code and objects. You will also be shown to leverage this knowledge to easily build a component of the data layer of the Video Online sample solution using the Visual Basic Development Add-In.

# F3

## DB2 Development Add-Ins

Next Generation Visual Studio Tools for z/OS, UNIX, and Windows

Abdul Al-Azzawe

DB2 Development Tools and SQL Debugger Architect  
abdulh@us.ibm.com



**IBM Data Management Technical Conference**

**Anaheim, CA**

**Sept 9 - 13, 2002**

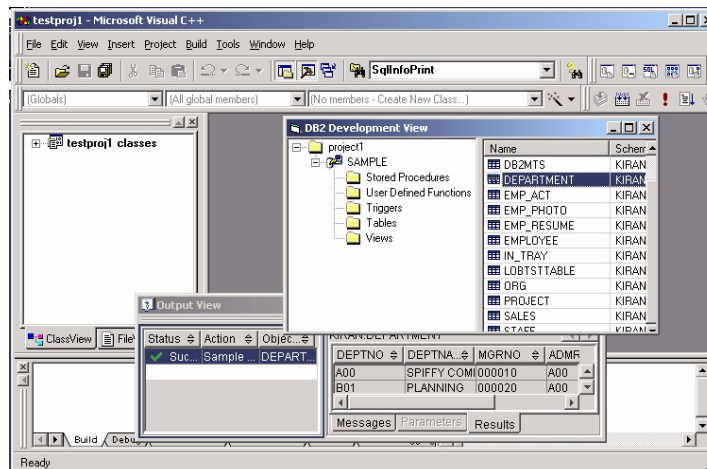
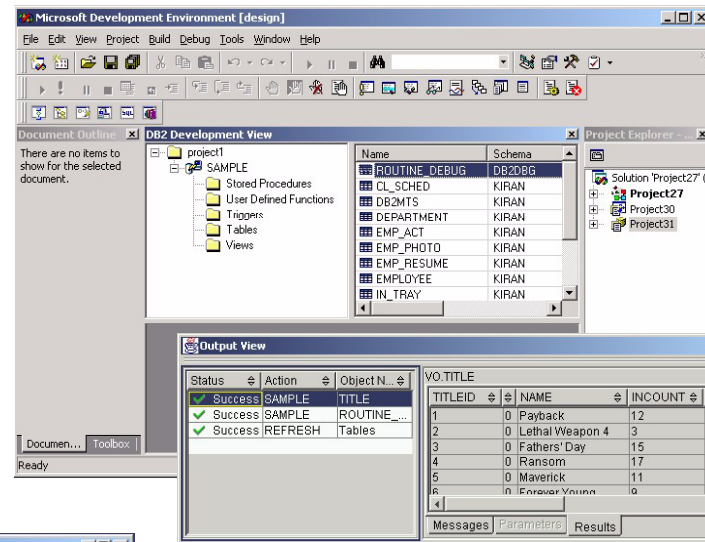
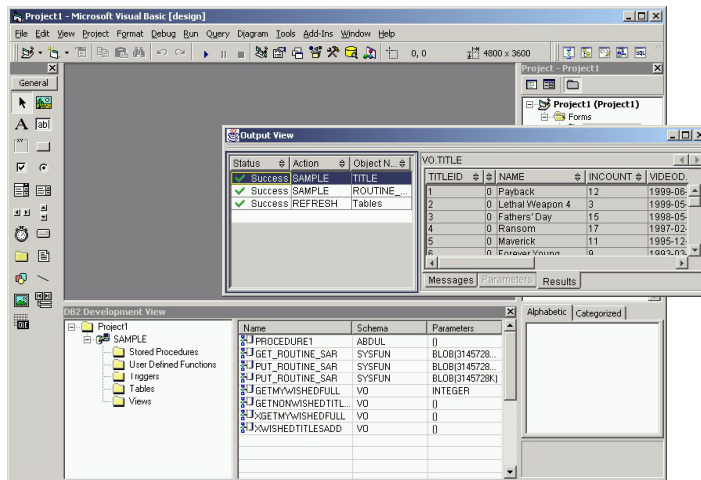


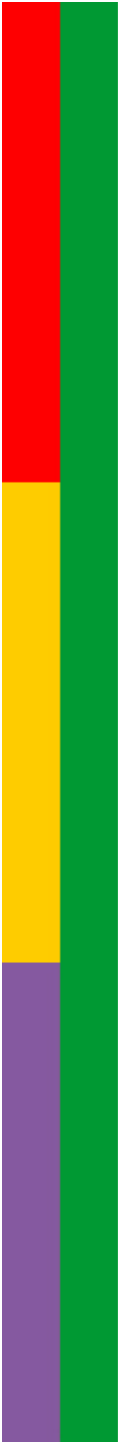
# Agenda

---

- Integration Overview
- Development View
- Working with Stored Procedures
- Working with User Defined Functions
- Working with Triggers, Tables, and Views
- Importing, Exporting, and Deploying
- Testing and Debugging
- Integrating with Visual Source Safe
- Summary

# DB2 Development Add-Ins





## Features – Integration

---

- Integrates tightly into Visual Basic, Visual C++, and Visual InterDev 6.0 IDEs using a DB2 Development View.
- Support for source code control management using Visual Source Safe 6.0.
- Client side ADO code generation for DB2 connections, procedures, and functions.
- Client side Data Environment object creation for DB2 connections, procedures, and functions.



## Features – Functionality

---

- Support for the entire family of DB2 servers (DB2 for z/OS™, iSeries™, UNIX®, Linux for 390, and Windows®).
- Enhanced z/OS support (SQL IDs and build options).
- Developing SQL and Java™ stored procedures.
- Developing SQL scalar and table user defined functions.
- Developing MQSeries®, OLE DB, and XML table functions.
- Developing structured data types for EJB methods and properties.
- Viewing database tables, views, triggers, stored procedures, and user-defined functions.



## Features – Project Management

---

- Import of stored procedures and functions from databases, other projects, and other source files.
- Deployment tool for deploying exported Development Center projects with binary deployment support for like DB2 servers.



## Features – Testing and Debugging

---

- Test any language stored procedure and user defined function.
- Saved test settings (parameter values with pre-execution and post-execution SQL scripts).
- Enhanced debugging of SQL stored procedures with variable value change support using an integrated SQL debugger.
- Simplified debugging of Java stored procedures using the IBM VisualAge® distributed debugger.



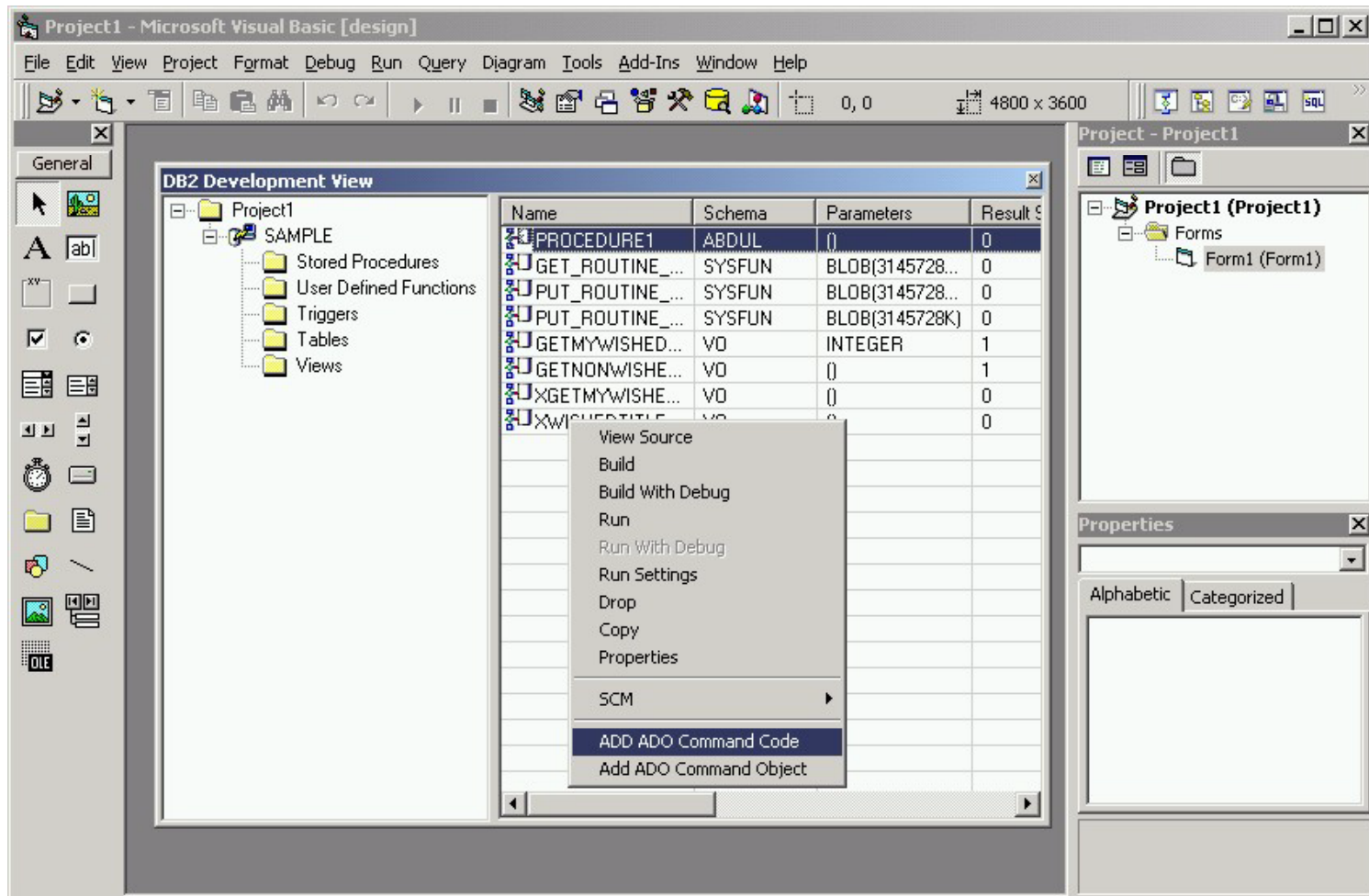


## Features – Performance & Scalability

---

- Online and offline database connections.
- Concurrent task execution using multi-threading.
- Connection pooling.
- Advance filtering prior to database catalogue data retrieval.
- Client caching and persistence of database catalogue data.

# DB2 Development View





# Development View

---

- Dockable native Visual Studio tool window.
- DB2 companion project having multiple connections.
  - **Stored procedures** – Create, show properties, view/edit source, test run, drop, and check-in / check-out.
  - **User defined functions** – Create, show properties, view/edit source, test run, drop, and check-in / check-out.
  - **Triggers** – Show properties (including the source).
  - **Tables** – Show properties and sample content.
  - **Views** – Show properties and sample content.
- Advance filtering prior to catalogue data retrieval.
- Client side catalogue data caching and persistence.



## *Demo* .:. Visual Basic Add-In

---

- Launch Development Add-In
- Work with DB2 Development View
- Add database connection
- Save project



# Working with Stored Procedures

---

- Develop SQL and Java (SQLJ / JDBC) stored procedures.
- View properties of any language stored procedure.
- Test any language stored procedure.
- Debug SQL and Java stored procedures.
- Wizard for creating new procedures.
- Customizable code generation using code fragments.
- Integrate with SQL Assist query builder.
- Java stored procedures with helper jars.
- Generate ADO Command wrapper.
- Generate Data Environment command object.



## *Demo* ... Stored Procedures

---

- Create new Java procedure using wizard.
- View procedure properties.
- Edit Java procedure.



# Working with User Defined Functions

---

- Develop SQL scalar and table user defined functions.
- View properties of any language user defined function.
- Test any language user defined functions.
- Wizard for creating new SQL functions.
- Customizable code generation using code fragments.
- Integrate with SQL Assist query builder.
- Wizards for creating specialized table functions
  - **MQ Series** message receiver / reader with optional view.
  - **OLE DB** data source with optional table and/or view.
  - **XML** document parser with optional XSL transformation.
- Generate ADO Command wrapper.
- Generate Data Environment command object.



## *Demo* ... User Defined Functions

---

- Create new SQL scalar function using wizard.
- Create new SQL MQ Series table function.
- Create and run OLE DB table function.
- Create and run XML parser table function.

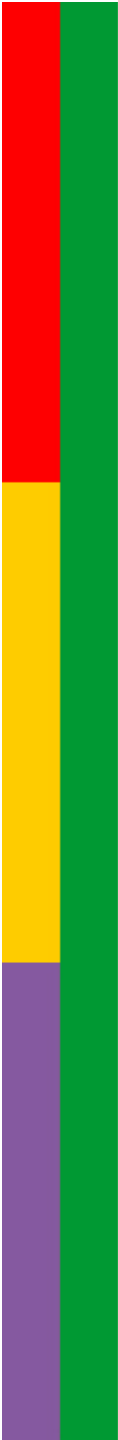




# Working with Triggers, Tables, and Views

---

- View source and properties of triggers.
- View properties and sample content of tables.
- View properties and sample content of views.



## *Demo* ... Tables

---

- Specify filter on tables.
- View properties of table.
- Sample content of table.



# Import, Export, and Deployment

---

- Import procedures and functions from text files, other projects, and other database connections.
- Deploy exported Development Center projects using deployment tool.
- Binary deployment to like servers (no compiler required).
- Modify build options prior to deployment.



## *Demo* ... Import, Export, and Deployment

---

- Import procedure from file using wizard.
- Import procedure from another database connection.
- Deploy exported project using deployment tool.
- Modify build options prior to deployment.



# Testing and Debugging

---

- Test run any language stored procedure and function.
  - Saved test parameter values.
  - Saved pre execution setup script.
  - Saved post execution cleanup script.
  - Results tracking in output view.
- Debug SQL procedures using integrated SQL debugger
  - SQL code, state, and exception tracking.
  - Break at source line or variable value change event.
  - Modify variable values during execution.
- Debug Java procedures using distributed Java debugger
  - Automated setup and launch of debugger



## *Demo* ... Testing and Debugging

---

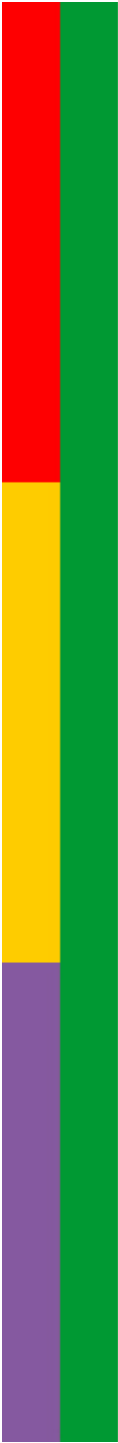
- Specify run settings (scripts and parameters).
- Use integrated SQL debugger.
- View results in output view.



# ADO Code and Data Environment Objects

---

- ADO code generation (VB and VBScript)
  - ADO Connection for DB2 database connections.
  - ADO Command for DB2 procedures and functions.
- Data Environment object creation
  - Data environment connection for DB2 database connections.
  - Data environment command for DB2 procedures and functions.
- Use IBM native OLE DB provider, IBMDADB2.
- Automatic mapping between DB2 SQL and ADO data types.



## *Demo* .:. DB2 Add-In for Visual Basic

---

- Launch DB2 Add-In.
- Add database connection.
- Generate ADO connection and command code.
- Create Data Environment objects.
- Check code into Visual Source Safe.





## Summary

---


- Tightly integrated into Visual Studio IDEs.
- Development of procedures and functions via easy-to-use wizards.
- Support for read-only access to stored procedures, UDFs, triggers, tables, and views.
- Online and offline database connection support with server catalogue data caching for offline mode.
- Integrated SQL debugger.
- Project management using Import, export, and deployment.
- Client-side VB and VBScript ADO code generation for DB2 connections, procedures, and functions.
- Supports Visual Source Safe.



## Additional Information

---

- DB2 Developer Domain (articles, news)
  - <http://www.ibm.com/software/data/developer>
- DB2 Video family of tutorials
  - <http://www.ibm.com/software/data/developer/samples/video>
- IBM Software Support
  - <http://www.ibm.com/software/support>



---

## User Notes (Backup Slides)

# Development Add-In for Visual Basic

The screenshot displays the Microsoft Visual Basic IDE interface. The main window is titled 'Project1 - Microsoft Visual Basic [design]'. The menu bar includes File, Edit, View, Project, Format, Debug, Run, Query, Diagram, Tools, Add-Ins, Window, and Help. The toolbar contains various icons for file operations and development. The 'General' tab is selected in the left-hand pane. The 'Output View' window is open, showing a table with the following data:

Status	Action	Object N...
✓ Success	SAMPLE	TITLE
✓ Success	SAMPLE	ROUTINE_...
✓ Success	REFRESH	Tables

Below the 'Output View' window, the 'DB2 Development View' is open, showing a tree view of the project structure. The tree view includes 'Project1' with sub-items: 'SAMPLE', 'Stored Procedures', 'User Defined Functions', 'Triggers', 'Tables', and 'Views'. The 'SAMPLE' folder is expanded, showing a list of database objects:

Name	Schema	Parameters
PROCEDURE1	ABDUL	0
GET_ROUTINE_SAR	SYSFUN	BLOB(3145728...
PUT_ROUTINE_SAR	SYSFUN	BLOB(3145728...
PUT_ROUTINE_SAR	SYSFUN	BLOB(3145728K)
GETMYWISHEDFULL	VO	INTEGER
GETNONWISHEDTITL...	VO	0
XGETMYWISHEDFULL	VO	0
XWISHEDTITLESADD	VO	0

# Development Add-In for Visual InterDev

The screenshot displays the Microsoft Development Environment (MDE) interface. The main window is titled "Microsoft Development Environment [design]". The menu bar includes File, Edit, View, Project, Build, Debug, Tools, Window, and Help. The toolbar contains various icons for file operations, development, and debugging. The Document Outline pane on the left shows a tree view for "project1" with folders for Stored Procedures, User Defined Functions, Triggers, Tables, and Views. The DB2 Development View pane in the center shows a table of database objects:

Name	Schema
ROUTINE_DEBUG	DB2DBG
CL_SCHED	KIRAN
DB2MTS	KIRAN
DEPARTMENT	KIRAN
EMP_ACT	KIRAN
EMP_PHOTO	KIRAN
EMP_RESUME	KIRAN
EMPLOYEE	KIRAN
IN_TRAY	KIRAN

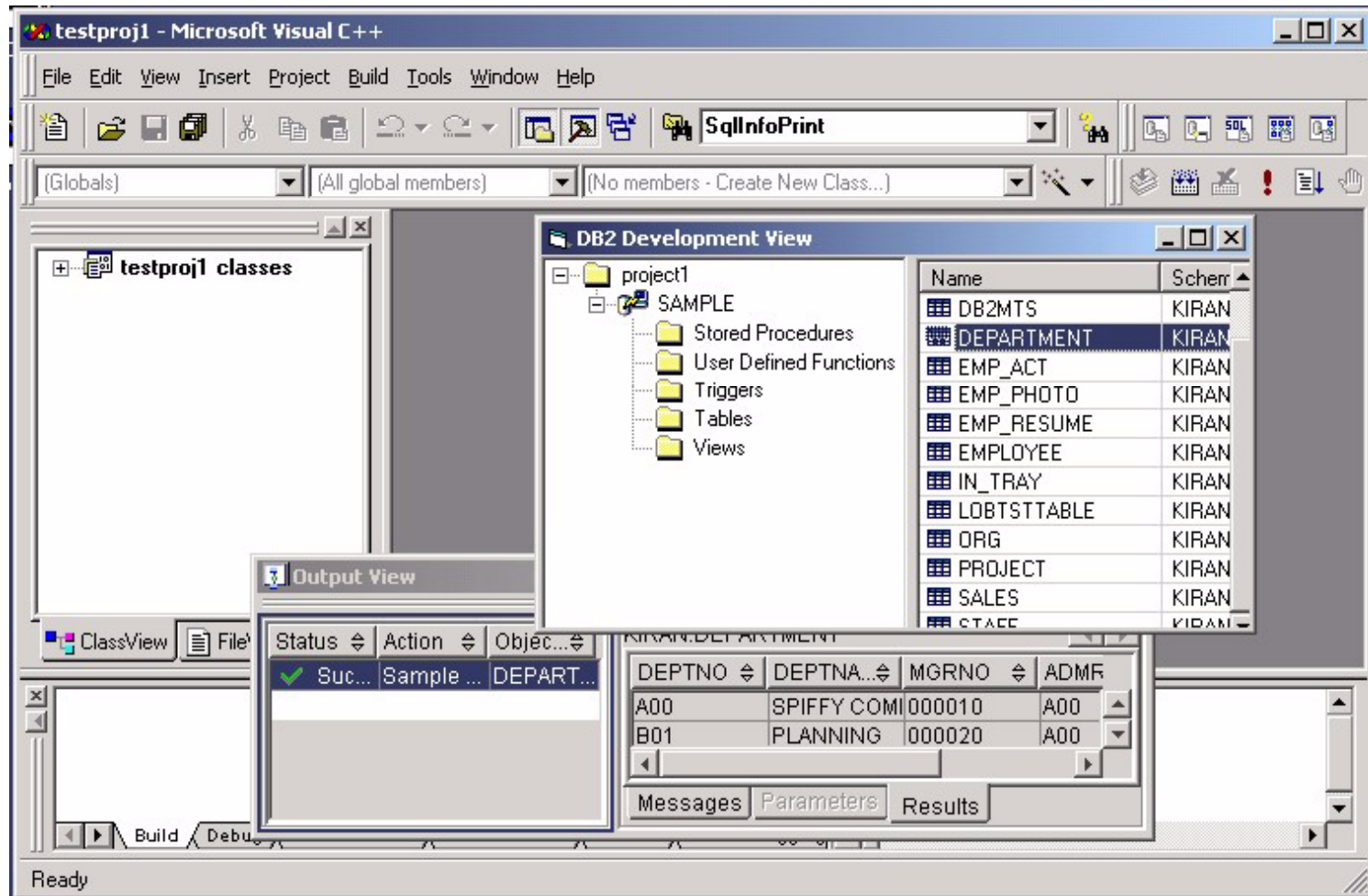
The Project Explorer pane on the right shows a Solution 'Project27' (3) with sub-projects Project27, Project30, and Project31. The Output View pane at the bottom shows a table of messages:

Status	Action	Object N...
Success	SAMPLE	TITLE
Success	SAMPLE	ROUTINE_...
Success	REFRESH	Tables

The Output View also displays a table of data:

TITLEID	NAME	INCOUNT	VIDEOD.
1	Payback	12	1999-06-
2	Lethal Weapon 4	3	1999-05-
3	Fathers' Day	15	1998-05-
4	Ransom	17	1997-02-
5	Maverick	11	1995-12-
6	Forever Young	9	1993-03-

# Development Add-In for Visual C++



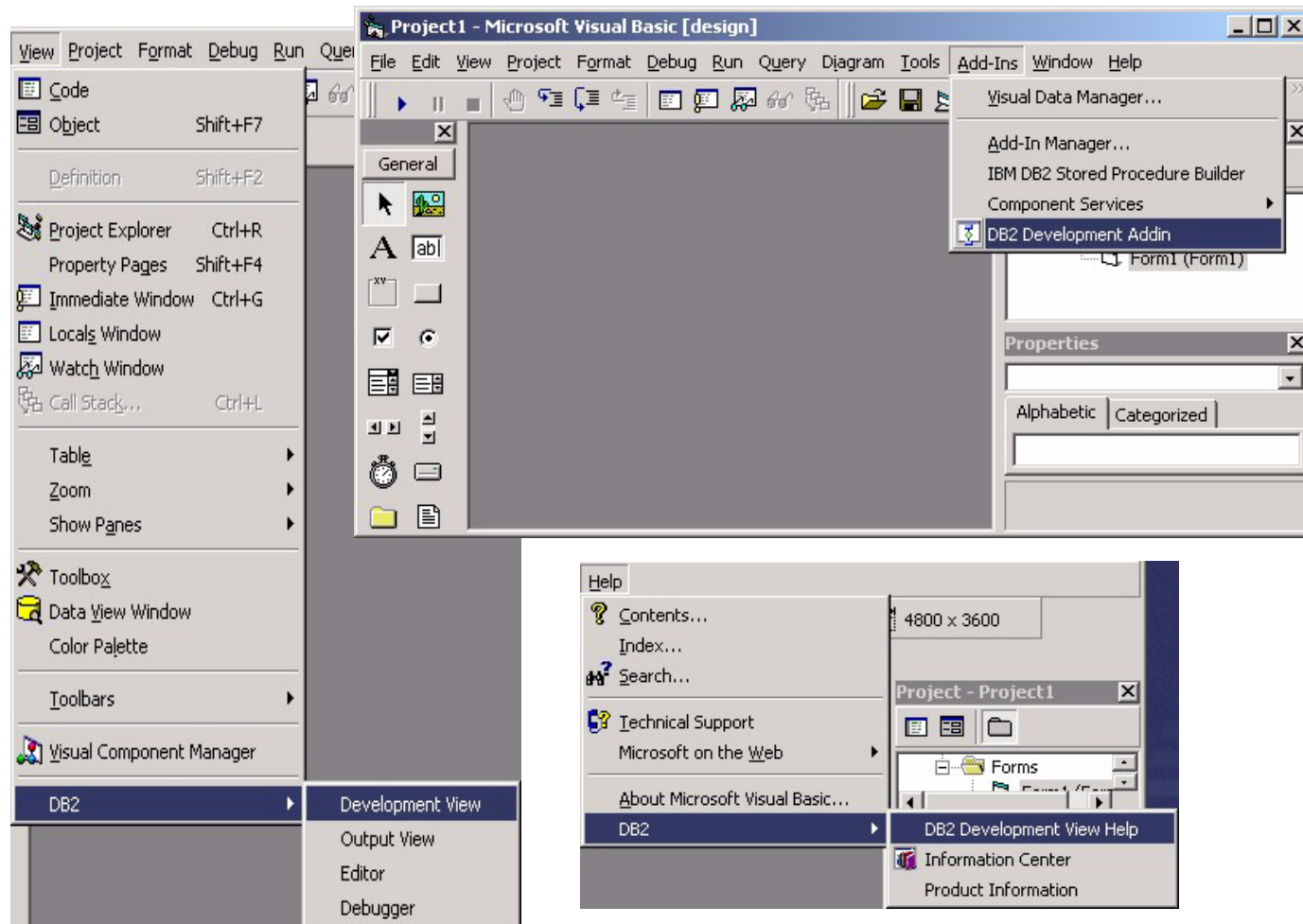


## Development Add-In Features

---

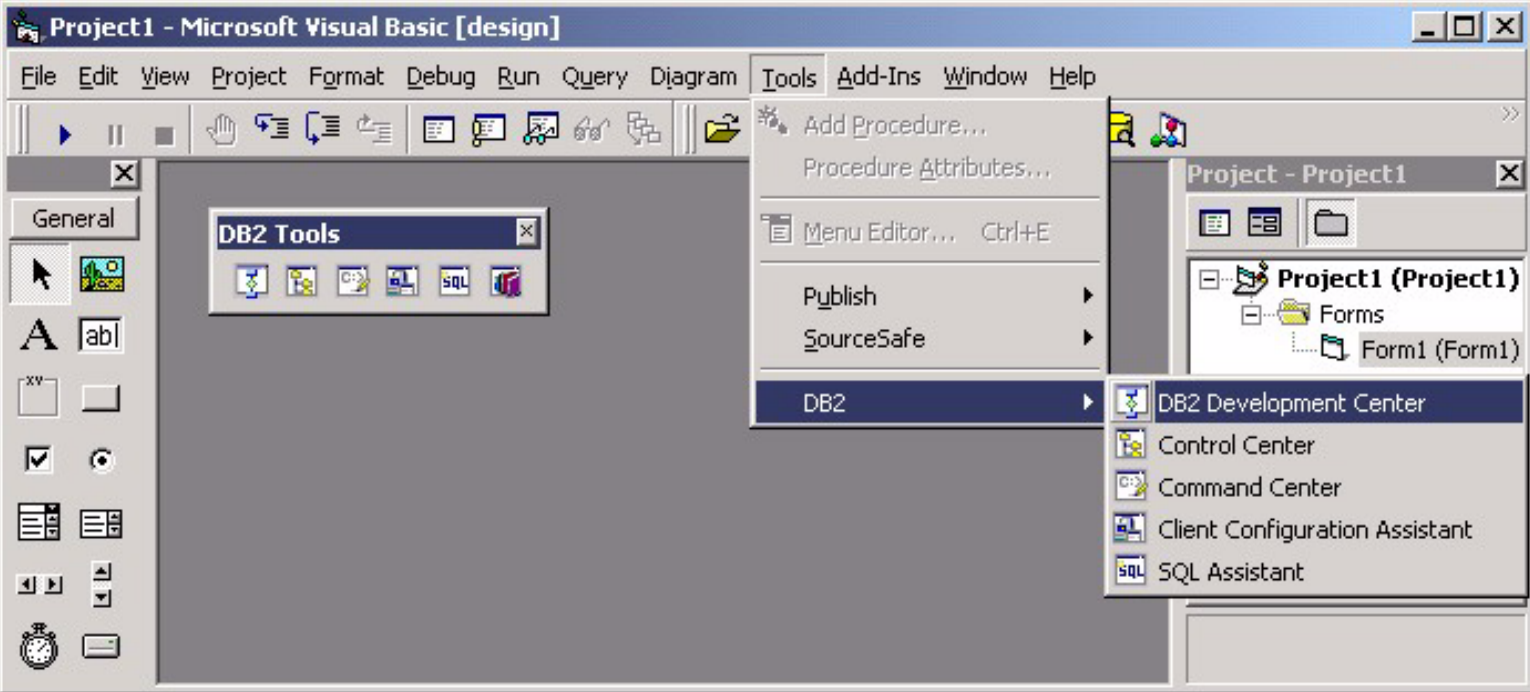
- Integrated into Visual Studio Menus / Toolbars
- Launch point for various DB2 Centers / Tools
- Dockable Development View
- Access to Development Center dialogs and wizards
- Development of SQL and Java stored procedures
- Development of SQL table and scalar UDFs
- Support MQ Series, OLE DB, and XML Table UDFs
- SQL Assistant Integration
- Microsoft Visual Source Safe integration
- Support for Active Data Objects (ADO)
  - VB and VBScript ADO code generation
  - Data Environment object creation

# Integrated DB2 Menus





# Launch DB2 Centers and Tools





# Development View Features

---

- Tables & Views
  - Properties and sample content
- Triggers
  - Properties and view source
- Stored Procedures
  - Create, import, export, and deploy
  - Properties, view source, test run, and drop
  - ADO code-gen, and Data Environment object creation
  - Edit, build, and debug of SQL and Java stored procedures
- User Defined Functions
  - Create, import, export, and deploy
  - Properties, view source, test run, and drop
  - ADO code-gen, and Data Environment object creation
  - Edit, build, and debug of SQL and Java UDFs
- Create of MQ Series, OLE DB, and XML table UDFs



# Add-Ins and Development Center

---

- New database connection wizard
- Object enumeration filter dialogs
- Create stored procedure wizards
- Create user defined function wizards
- Import, export, and deployment wizards
- SQL and Java Editor
- SQL Debugger (call stack, breakpoints, and variables)
- Output View (sample content, test run, and build)
- Property dialogs for various objects

# Database Connections

- Support entire family of DB2 servers
- Online and offline development
- Support OS/390 default SQLID, Package Owner ID, and Build Owner ID

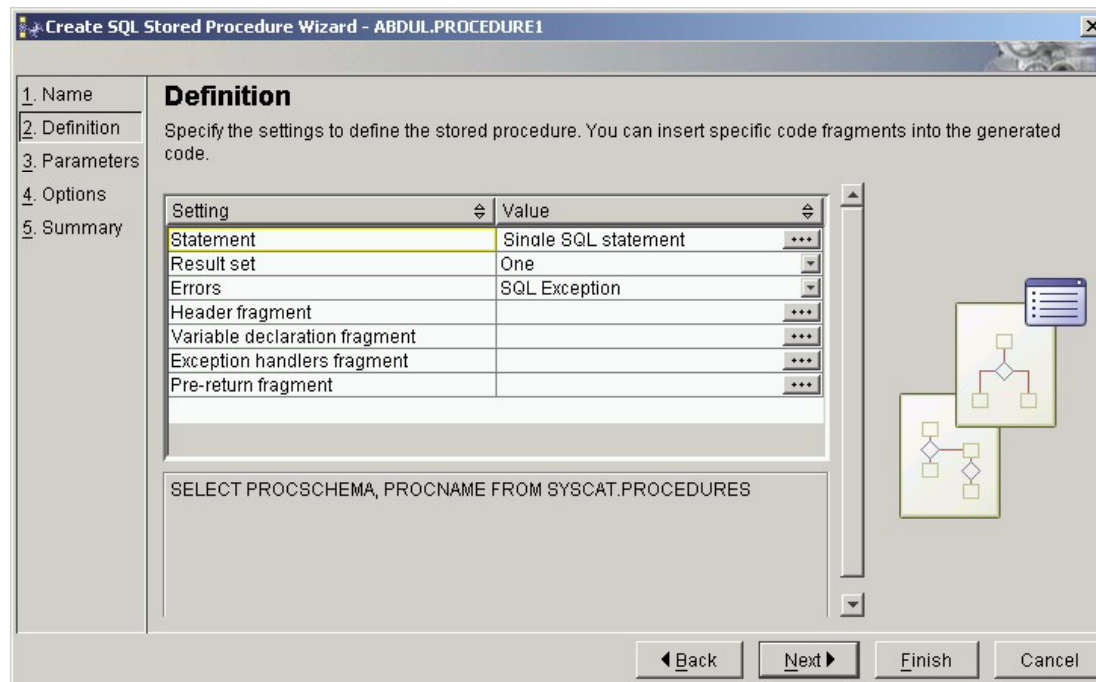
The screenshot shows the 'Add Database Connection Wizard - DCDB' dialog box. The 'Connection' step is selected in the left-hand pane. The main area contains the following fields and options:

- Driver:** IBM DB2 alias
- Database:** Alias: DCDB (with an 'Add...' button)
- Location(URL):** jdbc:db2:DCDB
- Driver class:** COM.ibm.db2.jdbc.app.DB2Driver
- User information:**
  - Use your current user ID and password
  - User ID:** abdul
  - Password:** masked with asterisks

At the bottom, there is a 'Test Connection' button and navigation buttons: 'Back', 'Next', 'Finish', and 'Cancel'.

# Stored Procedures

- Create wizard using customizable source templates
- Source code generation using multiple code fragments
- SQL and Java (with optional helper jars) Stored Procedures
- Run and view any language Stored Procedure



# User Defined Functions

- Create wizard using customizable source templates
- Source code generation using multiple code fragments
- SQL and Java (with optional helper jars) Scalar and Table UDFs
- Run and view any language UDF

1. Name  
2. Definition  
3. Return Data Type  
4. Parameters  
5. Options  
6. Summary

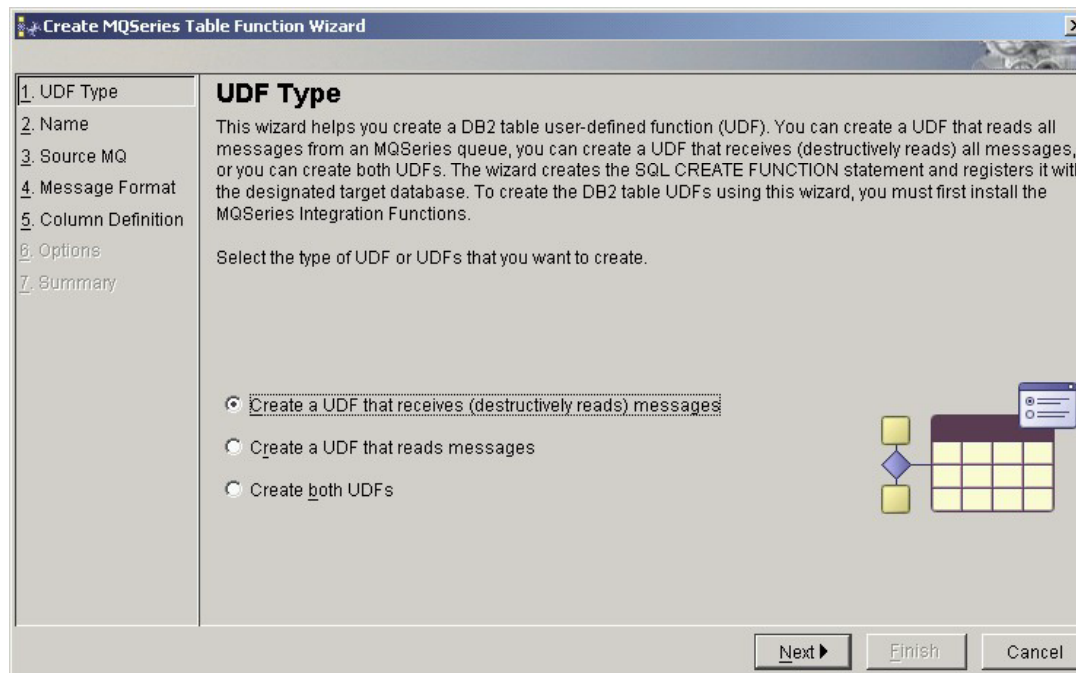
**Return Data Type**  
Specify the data type that you want the user-defined function to return.

Return type  
SQL type: INTEGER  
Length: 1  
Unit:   
Precision: 5  
Scale: 0  
 For bit data

Back Next Finish Cancel

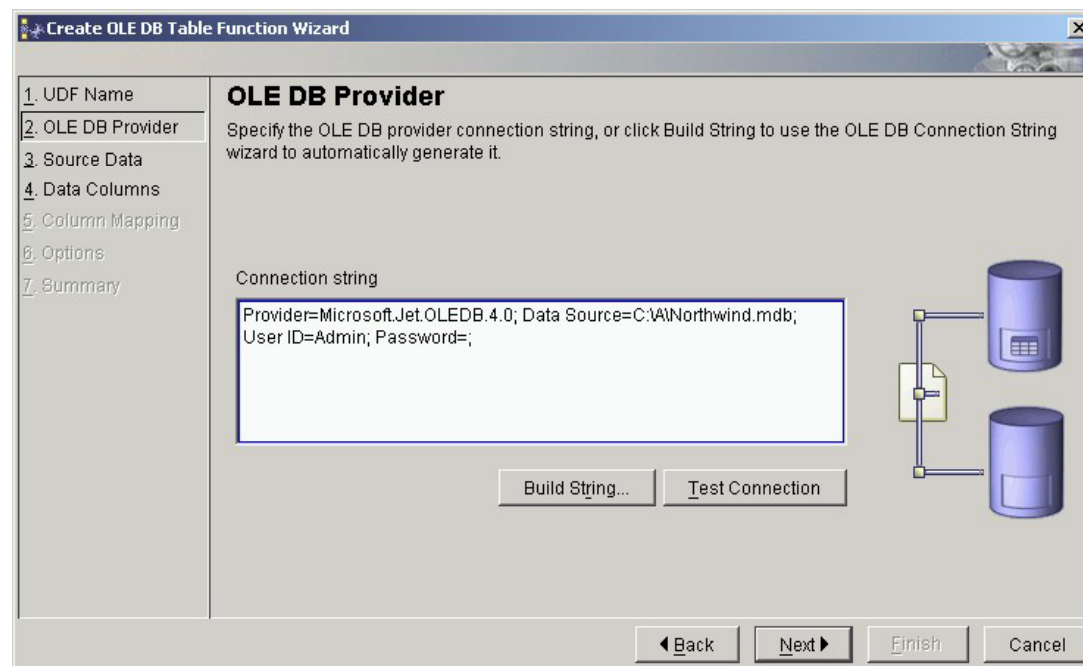
# MQ Series Table Function

- Create MQ Series Table UDF using wizard
  - Receive or Read MQ Series messages
  - Support delimited or positional data columns
  - Create optional table view



# OLE DB Table Function

- Create OLE DB Table UDF using wizard
  - Access to most OLE DB data sources using connection string wizard
  - Retrieve data from tables or using SQL query spanning multiple tables
  - Create optional table view
  - Optional import of data into new DB2 table -- Fast migration





# XML Parser Table Function

- Create XML Table UDF Wizard
  - Parse XML documents (extracting the SQL data from XML node values)
  - Supports XSL transformation prior to parsing

**Create XML Table Function Wizard**

1. Name  
2. XML Input  
3. XML Document Description  
4. Column Definition  
5. Options  
6. Summary

**Name**

This wizard helps you create a DB2 table user-defined function (UDF) that reads an XML document and returns data in relational table format. The wizard creates the SQL CREATE FUNCTION statement and registers it with the target database. The wizard also creates a set of helper UDFs that are referenced by the UDF that you are creating. The schema of the helper UDFs is XUDFS.

Specify the name of the table UDF, and optionally type a comment to describe the function.

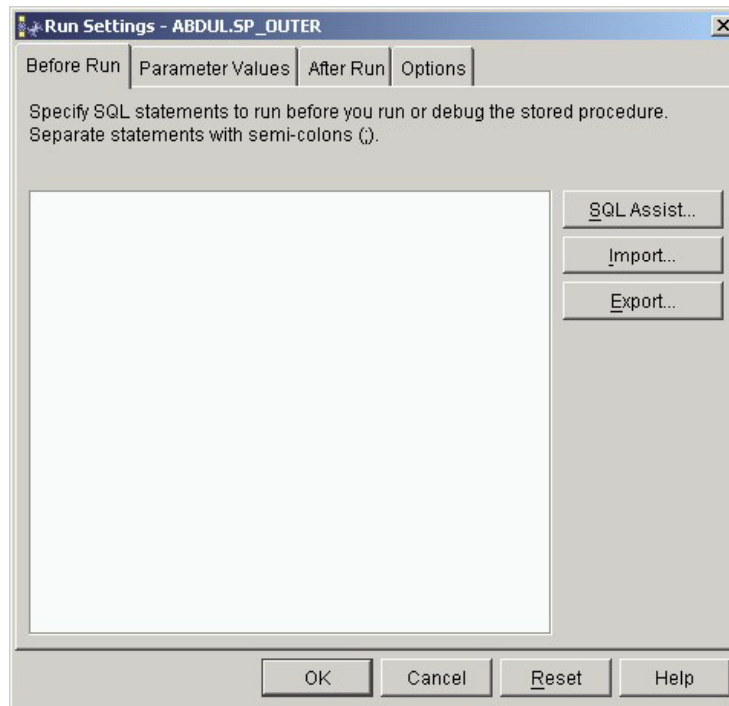
Name: XMLfunction  
Comment:   
 Replace existing

Next > Finish Cancel

# Run Settings

---

- Persist SP/UDF test settings
- Test run pre and post execution SQL scripts
- Parameter values, and run options



# Output View

- Multiple concurrent task tracking
- Multiple tasks per object
- Historical view of multiple task results
- Save / print task messages and results

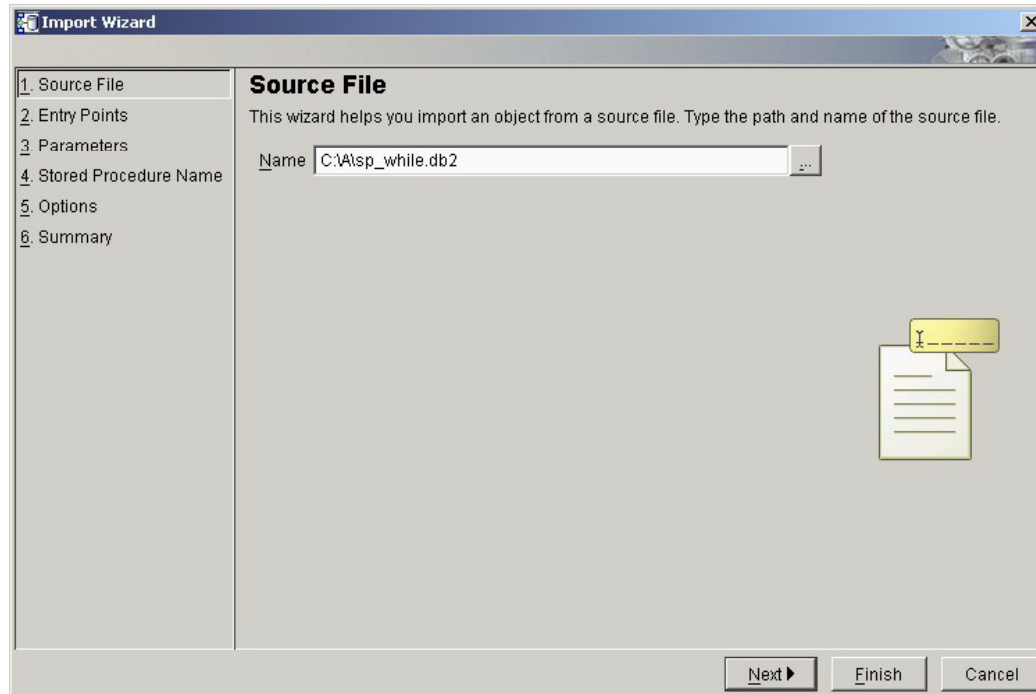
Status	Action	Object Name
✓ Success	Sample conte...	AUTHORS
✓ Success	Refresh	Table
✓ Success	Debug	SP_OUTER
✓ Success	Build for debug	SP_INNER
✓ Success	Build	SP_OUTER
✓ Success	Add	SP_OUTER
✓ Success	Add	SP_INNER
✓ Success	Refresh	Stored Procedures
✓ Success	Add	Database Conne...

ABDUL.AUTHORS				
AUTHORID	NAME	ADDRESS	CITY	STATE
james	James Johns	319 Dixon Ave.	Pleasant Hill	CA
edd	Eddy Marvin	452 Bowers Dr.	San Jose	CA
david	David Sawyer	221 Archie Blvd.	San Francisco	CA

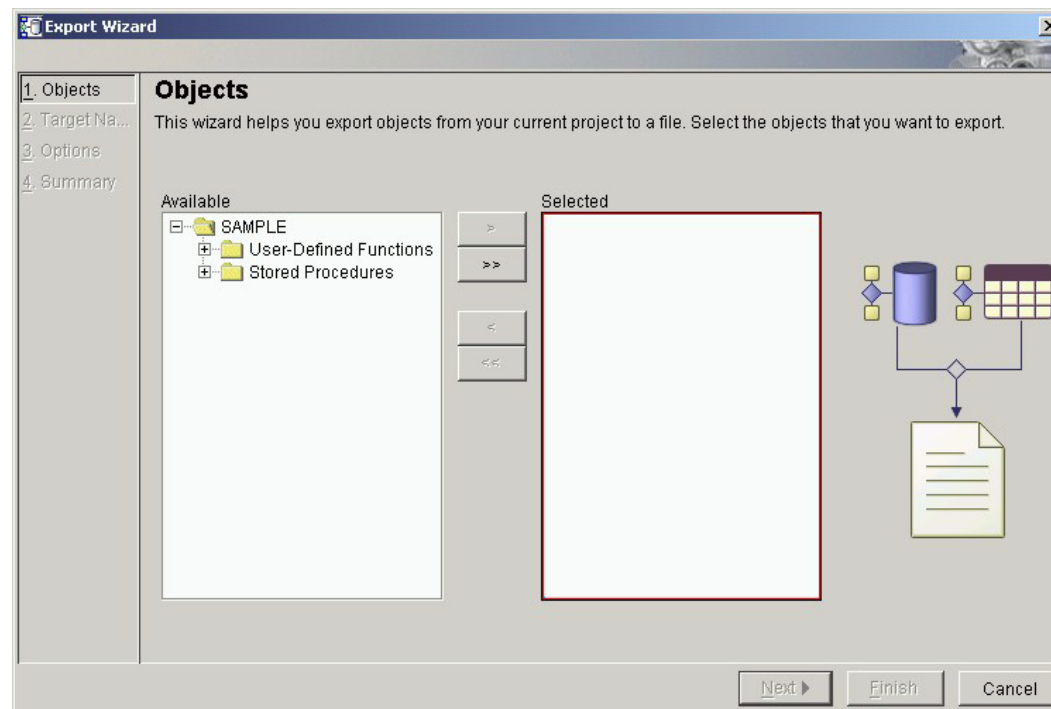
# Import Wizard

- Import Wizard for importing SPs/UDFs (from files, other projects, or DB2 databases) into current project



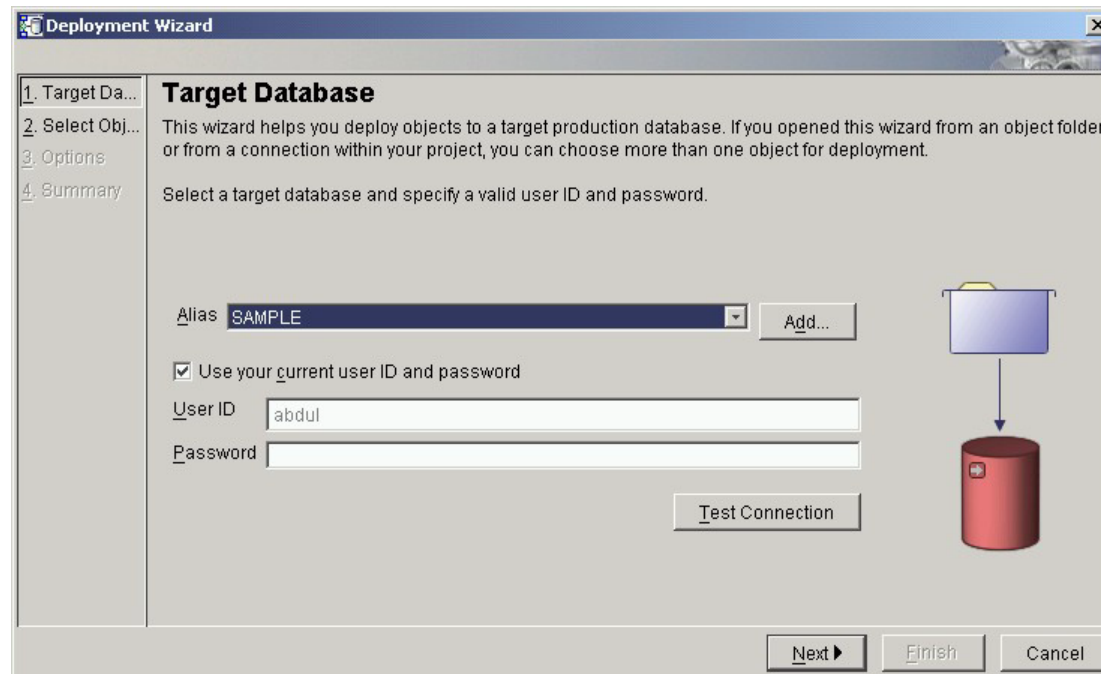
# Export Wizard

- Export Wizard for packaging current project to the local file system
  - Export as deployment project -- later deployment using tool/command
  - Export as deployment script (UNIX and Windows script files)
- Retrieves binaries from DB2 server for efficient redeployment



# Deployment Wizard

- Deploy current project or sub-project to production systems
- Efficient deployment -- avoiding rebuild for like servers
- Modify deployment rebuild options based on target server





# Deployment Tool and Command

---

- Deploy exported project or sub-project to production systems
- Graphical tool for interactive deployment
- Command line for batch and/or scheduled deployment
- Efficient deployment avoiding rebuild for like servers
- Modify deployment rebuild options based on target server

# Integrated SQL Debugger

The screenshot shows the Integrated SQL Debugger interface for a procedure named 'sp\_while'. The main window displays the following SQL code:

```
Row 15      Column 24      Insert      Browse
DECLARE CONTINUE HANDLER FOR not found SET at_end = 1;

set sp_allnames = '';

OPEN c1;
FETCH c1 INTO sp_schema, sp_name;
WHILE (at_end = 0) DO
    set sp_allnames = sp_allnames || ' : ' || sp_schema || '.' || sp_name;
    FETCH c1 INTO sp_schema, sp_name;
end while;
CLOSE c1;
set sp_names = sp_allnames;
```

The 'set sp\_allnames = '';' line is highlighted in yellow, and a breakpoint is set at line 24. The bottom panel shows the following sections:

Breakpoints		Call Stack		Variables			
%	Name	Name	Line	%	Name	Type	Value
☑	Line: 15			☑	SQLSTATE	CHAR(5)	00000
☑	Line: 24			☑	SQLCODE	INT	0
					SP_NAMES	VARCHAR(4000)	:ABDUL .PROCEDURE1:ABDUL .SP_INI
					SP_SCHEMAS	VARCHAR(32)	ABD%
					SP_ALLNAMES	VARCHAR(4000)	
					SP_NAME	VARCHAR(256)	



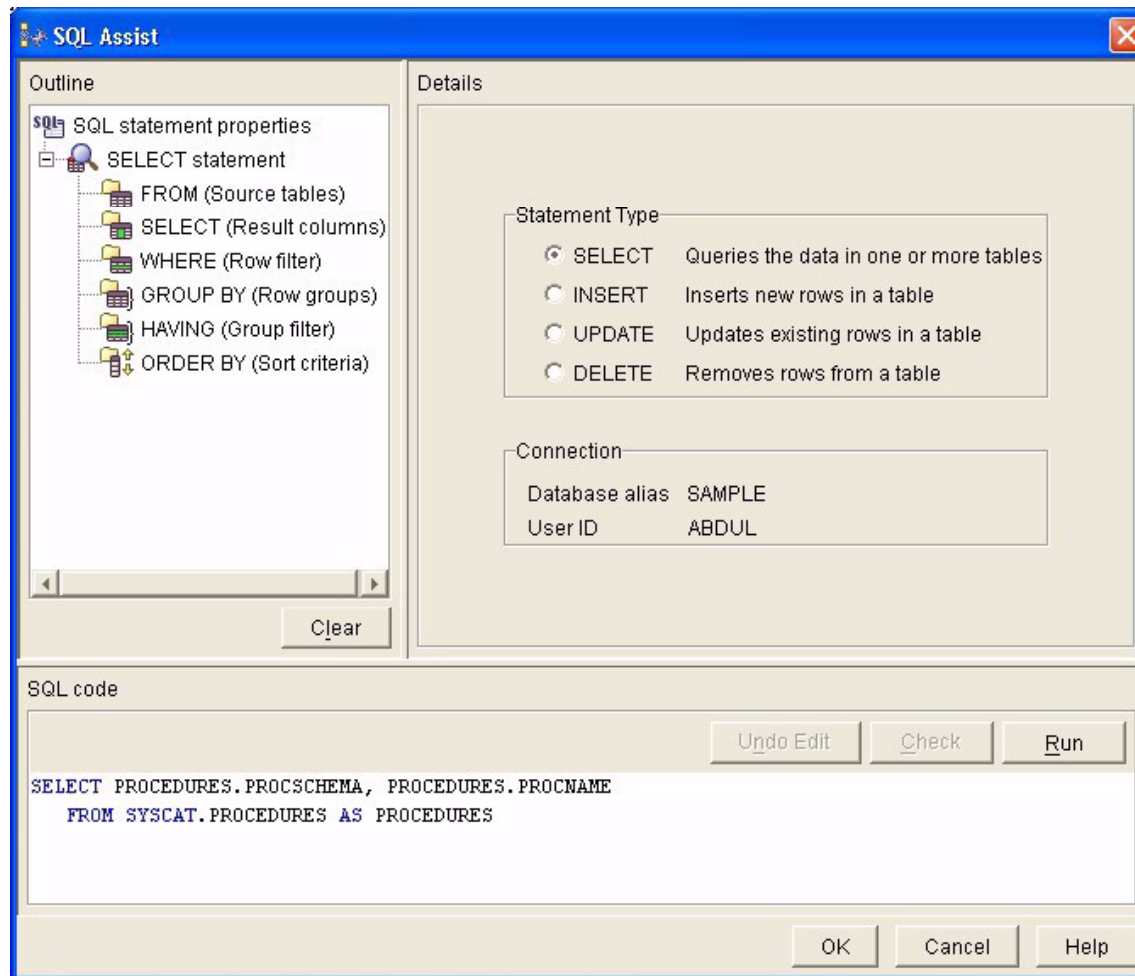


# SQL Debugger Features

---

- Fenced and unfenced stored procedure debugging
- Nested stored procedures support
- SQL State, SQL Code, and SQL Exception notification
- View and modify variable values
- Break on variable value change event
- Break on source code line reached event
- Break at Nth occurrence of a breakpoint event
- Enabling and disabling breakpoints
- Run, pause, step in, out, over, return, and run to line

# SQL Assist Query Builder





# SQL Assist Features

---

- GUI based query builder
- Pluggable component
- New features highlights:
  - Fullselect support (UNION, INTERSECT, ...etc.)
  - Redesigned condition builder
  - Improved performance
  - Accessibility
  - Round trip query editing
  - Extensible design (improved SQL coverage)



# ADO Connection Code

---

- VB and VBScript Code Generation

```
' Create an ADO Connection Object.  
Public Function CreateConnection() As ADODB.Connection  
    ' Create new connection object  
    Dim newConnection As ADODB.Connection  
    Set newConnection = New ADODB.Connection  
    With newConnection  
        .ConnectionString = "Provider=IBMDADB2; " & _  
            "Data Source=SAMPLE"  
    End With  
    ' Return new connection object  
    CreateConnection = newConnection  
End Function
```



# ADO Command Code

---

- VB and VBScript Code Generation for Functions and Procedures

```
' Create an ADO Command Object.
Public Function CreateCommand(myCon As ADODB.Connection) As ADODB.Command
    ' Create new command object
    Dim newCommand As ADODB.Command
    Set newCommand = New ADODB.Command
    With newCommand
        .CommandType = adCmdStoredProc
        .ActiveConnection = myCon
        .CommandText = "TEST.PROCEDURE1"
        .Parameters.Append newCommand.CreateParameter("empNumber", _
            adInteger, _
            adParamInput)
        .Parameters.Append newCommand.CreateParameter("empFullName", _
            adVarChar, _
            adParamOutput, _
            20)
    End With
    ' Return new command object
    CreateCommand = newCommand
End Function
```



## DB2 Add-Ins for Visual Studio .Net

---

- Visual Studio Integrator Program (VSIP)
- DB2 project supporting tables, views, procedures, functions, triggers, and other DB2 scripts.
- Wizards for creating DB2 objects.
- Wizards for integrating with MQSeries, XML, and WebServices.
- Use Visual Studio.Net views (editor, debugger, properties, output).
- DB2 Server Explorer.
- ADO.Net code generation.