

DB2 Stored Procedures and UDFs: A Primer

Quest Software

Agenda

- Why Use DB2 Stored Procedures?
- Why Use UDFs?
- Setting up the Environment.
- Performance Tuning.
- References.

DB2 Stored Procedures Benefits

- Secure access to DB2 data.
- Access non-DB2 data if necessary – VSAM, IMS, etc.
- Code business logic once
- Maintenance is simpler
- Performance improvement?
 - Instead of multiple SQL statements being sent
- Lots of languages to choose from.

DB2 Stored Procedures

2 types of Stored Procedures
to choose from:

- SQL Procedures
- External Procedure

DB2 SQL Stored Procedures Benefits

- Programmer productivity – Familiar standard SQL.
- Only SQL statements and procedural control-flow statements.
- Portable.
- Compile and register into DB2 tables.
- Not interpreted, but compiled as C code.
- Member of a schema (dependencies).

DB2 SQL Stored Procedures

```
CREATE PROCEDURE UPDATE_SALARY
  (IN EMPLOYEE_NUMBER CHAR(10),
   IN RATE DECIMAL(6,2))
LANGUAGE SQL
WLM ENVIRONMENT SAMP1
COMMIT ON RETURN YES
IF RATE <= 0.50
THEN UPDATE EMP
  SET SALARY = SALARY * RATE
  WHERE EMPNO = EMPLOYEE_NUMBER;
ELSE UPDATE EMP
  SET SALARY = SALARY * 0.50
  WHERE EMPNO = EMPLOYEE_NUMBER;
END IF
```

DB2 External Stored Procedures

- As name Implies the code for external stored procedure is stored outside the database.
- Multiple threads can call and reuse the same stored procedure by defining as **reentrant**.

DB2 Stored Procedures Benefits

- To define the SP as reentrant...
 - Specify STAY RESIDENT YES on the CREATE PROCEDURE statement.
 - The default is STAY RESIDENT NO. Then, the SP is deleted from memory as soon as it has completed executing.
 - The SP will not need to be loaded into memory each time, so wait time is reduced.
 - Multiple threads will be able to share one copy of the SP in memory.

DB2 External Stored Procedure - CREATE

```
CREATE PROCEDURE
  SYSPROC.MYPROC(IN INT, OUT INT, OUT2
  DECIMAL(7,2))
  LANGUAGE COBOL
  EXTERNAL NAME MYMODULE
  PARAMETER STYLE GENERAL
  WLM ENVIRONMENT PARTSA
  STAY RESIDENT YES
  RESULT SETS 1;
```

Stored Procedures/UDFs

External Languages

- COBOL – LE/370 environment required.
- C – LE/370 environment required. V6 addition
- Assembler, PL/I
- C++, REXX and FORTRAN for stored procedures only – not for UDFs

DB2 Stored Procedures

SQL API's

CLI/JDBC

- SQL via API calls
- Dynamic SQL
- No precompiling
- Code is portable.
- Authorization by User

EXEC SQL/SQLJ

- SQL is embedded, e.g. C code
- Static & Dynamic SQL
- Translate SQL and create package.
- Authorization by Binder/Invoker

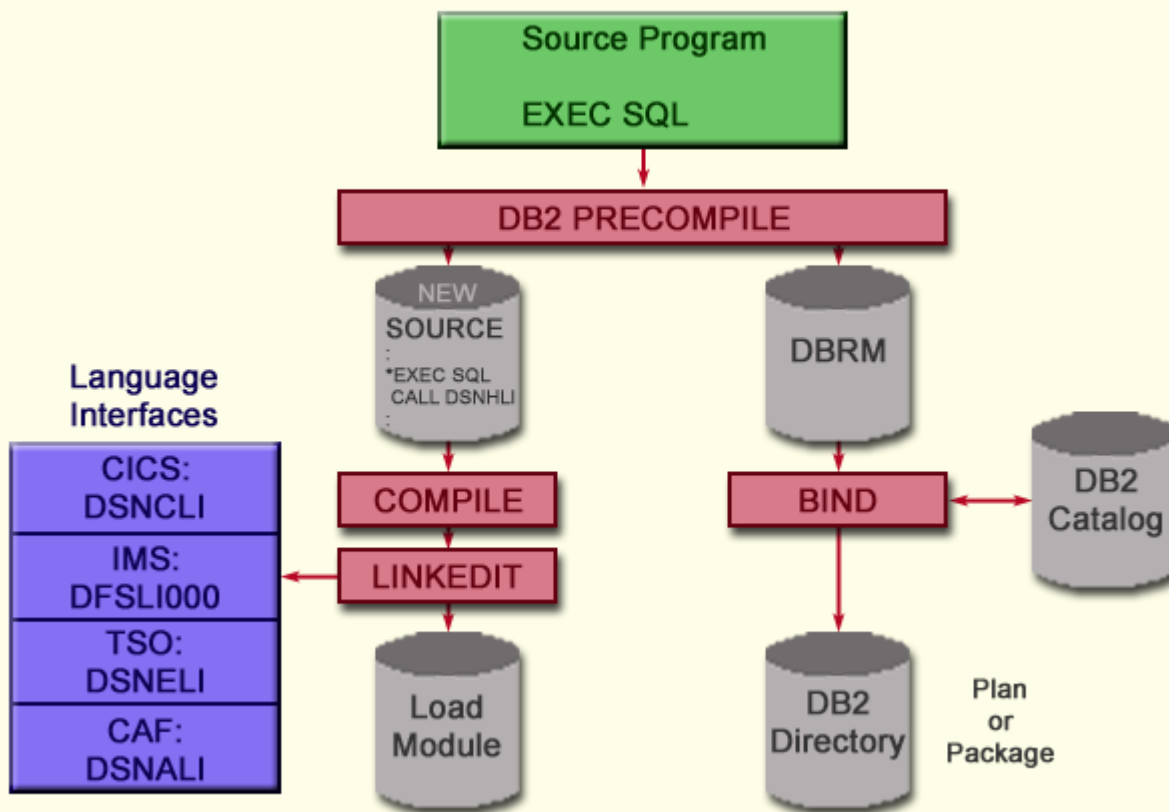
DB2 Stored Procedures Program Prep

Basic steps.

1. Precompile SQL source into byte-code that DB2 will recognize (this step is not needed for JDBC Java programs).
2. Bind extracted SQL into DB2 packages.
3. Compile/Link-edit.
4. Java, package the Java .class files for deployment platform.

DB2 Stored Procedures Program Prep

Program Preparation Cycle



DB2 User Defined Functions

What are they?

- Program code that can be called as a function in a SQL statement.
- Two types
 - **Sourced**: Based on another UDF or built-in function. Usually to enable a function for a user-defined data type.
 - **External**: Written in a supported language.
- Environment setup the same as for stored procedures. Run in WLM address spaces.

DB2 User Defined Functions Benefits

- A UDF in DB2 is like a host language subprogram, but can be invoked in a SQL statement (like a built-in function).
- Another way to share standard code for ease of use.
- Maintenance is done in one place.

DB2 User Defined Functions CREATE statement

- ***Sourced:***

```
CREATE FUNCTION AVE (HATSIZE) RETURNS  
HATSIZE  
SOURCE SYSIBM.AVG (INTEGER);
```

This UDF will enable the use of a user-defined data type HATSIZE with a built-in function, INTEGER.

DB2 User Defined Functions

CREATE statement

- ***External:***

```
CREATE FUNCTION CENTER (INTEGER, FLOAT)
  RETURNS FLOAT
  EXTERNAL NAME 'MIDDLE'
  LANGUAGE C
  DETERMINISTIC
  NO SQL
  FENCED
  PARAMETER STYLE DB2SQL
  NO EXTERNAL ACTION
  STAY RESIDENT YES;
```

DB2 User Defined Functions

External

- Table functions,
 - A special kind of UDF that returns a table.
 - Invoked in the FROM clause of a SELECT.
 - Can access non-DB2 data, including external data, in a SQL query. A file on the Web, a flat file on a system, etc. will be returned as a relational table from the function. Then can be INSERTed into a DB2 table, for example:

```
INSERT INTO global_temp_table  
SELECT <column names>  
FROM TABLE(DOCMATCH(d:\prod\data\filename)) ...
```

DB2 User Defined Functions

External

```
CREATE FUNCTION DOCMATCH  
(VARCHAR(255))  
  RETURNS TABLE (DOC_ID CHAR(16))  
  EXTERNAL NAME ABC  
  LANGUAGE C  
  PARAMETER STYLE DB2SQL  
  NO SQL  
  DETERMINISTIC  
  NO EXTERNAL ACTION  
  FENCED  
  SCRATCHPAD  
  FINAL CALL  
  DISALLOW PARALLEL  
  CARDINALITY 20;
```

DB2 Stored Procedures Coding

- Global Temporary Tables
 - Handle like a regular table, but it is temporary...**rows deleted at COMMIT.**
 - CREATE GLOBAL TEMPORARY TABLE temp_quest.
 - CREATE LIKE quest_tb.
 - Authority required same as for CREATE TABLE; or grant CREATETAB.
 - Not associated with a tablespace or database. Rows are written to work tablespaces on DSNDB07.
 - No logging or locks - efficient processing.

Stored Procedures/UDFs Environment Setup

- First step – Assemble an implementation SWAT team!
 - MVS and DB2 System Programmers
 - May need Unix skills for UNIX System Services on OS/390 (for JAVA).
 - DBA.
 - Developer.
- Then, get the IBM Redbooks (more on this toward the end)!

Stored Procedures/UDFs Environment Overview

Big Picture – sequence of events:

**Client -> Function -> DB2 -> WLM
->AE-> RRS -> DB2 -> Client**

Stored Procedures/UDFs Environment Setup

Tying it all together – sequence of events:

1. A client application connects to an OS/390 DB2 subsystem.
2. A CALL is issued for a stored procedure.
3. DB2 searches SYSROUTINES to locate the SP.
4. IF a WLM application environment is named in the WLM AE column, DB2 will ask WLM to initiate that WLM AE.
5. WLM takes over. The PROC associated with the AE is invoked with an MVS Start command for the PROC.

Stored Procedures/UDFs Environment Setup

Tying it all together – sequence of events:

6. The AE Address Space is created and begins running program DSNX9WLM, DB2's WLM application program.
7. The first task of DSNX9WLM is to establish a connection back to the DB2 subsystem using the RRS Attach Facility (RRSAF).
8. Once the communication path is established back to DB2, the WLM AE will fetch the SP program and run it.

Stored Procedures/UDFs Environment Setup

Tying it all together – sequence of events:

9. Results of the executed SP program are returned to DB2 via the RRSAF communication link.
10. The client receives the results from DB2.

Stored Procedures/UDFs Environment Setup

- WLM (Workload Manager) Address Spaces
 - Balance workloads dynamically – automatically routing requests to the least-busy address space available.
 - Starts up **new address spaces** if needed.
 - Inherit the **dispatching priority** of the DB2 thread that issued the CALL. So, high-priority work gets its SPs/UDFs executed ahead of lower-priority work.
 - Required for JAVA. Required for all in V8.

Stored Procedures/UDFs Environment Setup

- WLM Compatibility Mode
 - Before WLM, MVS controlled its dispatching priorities using two PARMLIB members...
 - installation performance specification (IPS)
 - installation control specification (ICS)
 - Manual effort to make changes as new applications came along.
 - Cumbersome, error-prone, and required IPL.
 - The procs for Application Environment address spaces must be started by operator

Stored Procedures/UDFs Environment Setup

- WLM Goal Mode
 - Goal Mode uses service polices and class periods to control dispatching.
 - Each service class is given a goal.
 - Response Goal: “TSO transactions should complete in .5 seconds”. WLM will try to achieve this goal by asking OS/390 to provide more resources.
 - In Goal Mode, WLM starts and stops address spaces as needed. No operator intervention

Stored Procedures/UDFs Environment Setup

- WLM Address Spaces
 - Application Environments are groups of address spaces started by the same PROC and used to help meet performance goals by dynamically changing the number of address spaces used for the workload.
 - Each AE definition represents one or more stored procedures & UDF's.

Stored Procedures/UDFs Environment Errors

How do I know when a WLM application environment is down?

D WLM,APPLENV=* will display the current status of each application environment.

RESPONSE=QSD1

IWM029I 12.57.37 WLM DISPLAY 364

APPLICATION ENVIRONMENT NAME	STATE	STATE DATA
DSD6CBMT	AVAILABLE	
DSD6CBST	AVAILABLE	
DSD6ENVW	AVAILABLE	
DSD6ENWT	AVAILABLE	
DSD6JBMT	STOPPED	

Stored Procedures/UDFs Security w/RACF

- Each AE definition must have a Server class policy defined by RACF.
- DB2 uses the RACF resource name "DB2.<subsystem_id>.<wlm_appl_env>" to control which userids have rights to create a routine in a WLM environment.

Stored Procedures/UDFs Environment Setup

- Resource Recovery Services (RRS)
 - If you have a SYSPLEX, RRS enables dynamic shifting of resources.
 - Also, enables communication between DB2 and applications via RRSAF (attach facility).

Stored Procedures/UDFs Environment Errors

If RRS is not running, or not configured:

The communication error will look like:

SSID:DSNQ - Error during IDENTIFY,

Return Code: 8 Reason Code: F30091

...and the error in the application

environment will look like this:

```
DSNX967I DSNX9WLM ATTEMPT TO PERFORM  
WLM FUNCTION IWMMCOMM FAILED WITH WLM
```

```
RC = 00000008 RSN = 1104085C
```

```
SSN = D999 PROC=...
```

Stored Procedures/UDFs

EXECUTE Authority in DB2

- Authorizations to CALL a stored procedure or invoke a User Defined Function:
 - EXECUTE privilege on the routine (sp or udf).
 - Who must have the privilege for an SP depends on whether the SP name in the CALL is a variable or a literal.
 - If SP name is a literal, the owner of the client app package must have EXECUTE authority on the SP.
 - DYNAMICRULES bind option, if used, dictates who needs EXECUTE authority.

Stored Procedures/UDFs

DISPLAY command

```
-DISPLAY PROCEDURE(ROSTER.*,  
  PROD.*)
```

Lists stored procedures that have been CALLED since DB2 was last started. Information includes:

- Schema
- WLM_ENV name
- Status of SP (Started, Stopped/why)
- Number of active threads using the load module, queued threads waiting for SP, and timed out CALLS.

Stored Procedures/UDFs

DISPLAY command

-DISPLAY FUNCTION SPECIFIC (*.*)

Lists functions that have been CALLED since DB2 was last started. Information includes:

- **Schema**
- **WLM_ENV name**
- **Status of function (started, stopped/why)**
- **Number of active threads currently running the function, queued threads waiting for the function to be scheduled, and timed-out waits.**

Stored Procedures/UDFs

START command

-START/STOP FUNCTION SPECIFIC (*.*)

Activates a UDF that is stopped.

-START/STOP PROCEDURE (*.*)

Activates the definition of a stored procedure that is stopped or refreshes one that is cached.

DB2 Stored Procedures Performance Tuning

DISPLAY THREAD (*) to see current status of threads calling stored procedures:

ST = SP means that the thread is executing in a stored procedure.

ST = SW means that the thread is waiting for a stored proc to be scheduled.

DB2 Stored Procedures Performance Tuning

If DISPLAY THREAD (*) output's status indicates too much waiting, investigate:

- WLM service policies.
- WLM application environment status and TCB's defined.
- Application problems (locking, badly tuned SQL, long-running tasks).
- System tuning possibilities (buffer pools, etc.)

DB2 Stored Procedures Performance Tuning

- If DB2 runs out of TCBs for your SP, and delays the start of the SP after a request, you won't see any SQLCODEs.
- You can investigate further with traces...

**-START TRACE(PERFM) DEST(??) CLASS(??)
IFCID(233,242,243) TDATA(CPU,DIST)**

233: generated when a SP is called or completes.

242: generated when a SP has to wait for a TCB (max TCBs hit).

243: generated when a waiting SP can now run (max TCBs hit).

Performance Tuning

- Tune the USS environment.
 - See www-1.ibm.com/servers/eserver/zseries/zos/unix/

- Tune TCP/IP.

- Tune your SP and client code.
 - Cache objects, etc. Lots of recommendations:
 - www-1.ibm.com/servers/eserver/zseries/software

References

- **IBM Redbooks:**

- ✓ SG24-5945, *DB2 Java Stored Procedures: Learning by Example.*
- ✓ SG24-5619, *Java Programming Guide for OS/390.*
- ✓ SG24-4693, *Getting Started with DB2 Stored Procedures: Give Them a Call through the Network.*
- ✓ SG24-5485-01, *Cross Platform DB2 Stored Procedures*
- ✓ REDP0183 *Setup of the DB2 Application Development Environment for Stored Procedures*

Conclusion

- Successful implementation of DB2 Stored Procedures and UDFs takes:
 - A team of the right people.
 - LOTS of planning.
 - Important environmental decisions.
 - Application requirements knowledge.
 - Use of reference material available.
 - Performance monitoring and tuning of system.

THANK YOU
FOR LISTENING