Introducing Db2 Stored Procedures on z/OS

Objectives

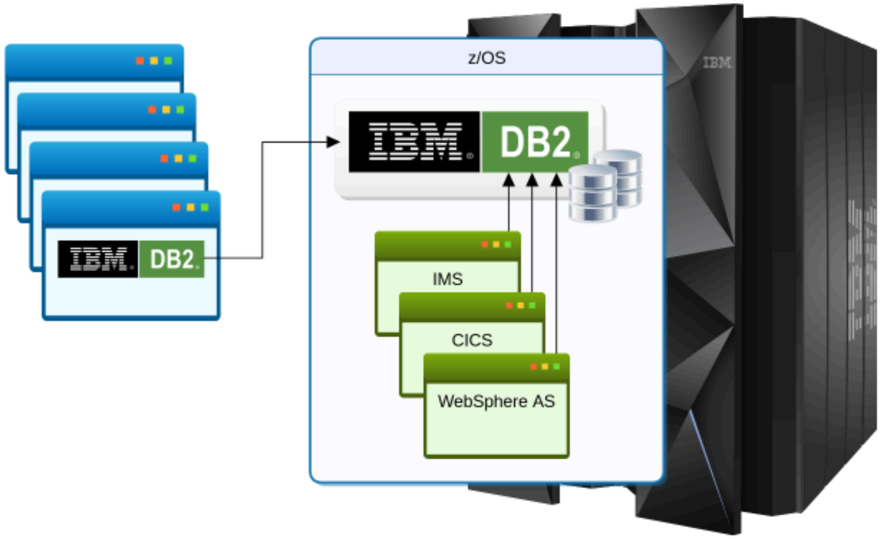
Introducing Db2 Stored Procedures on z/OS

In this module, you will be introduced to Db2 stored procedures on z/OS, which have many specific features, functions, and requirements.

Db2 database administrators (DBAs) and application programmers must understand stored procedures on z/OS so they can develop, implement, and maintain them. Managers and help desk staff also require this knowledge to manage problems and projects involving stored procedures on z/OS.

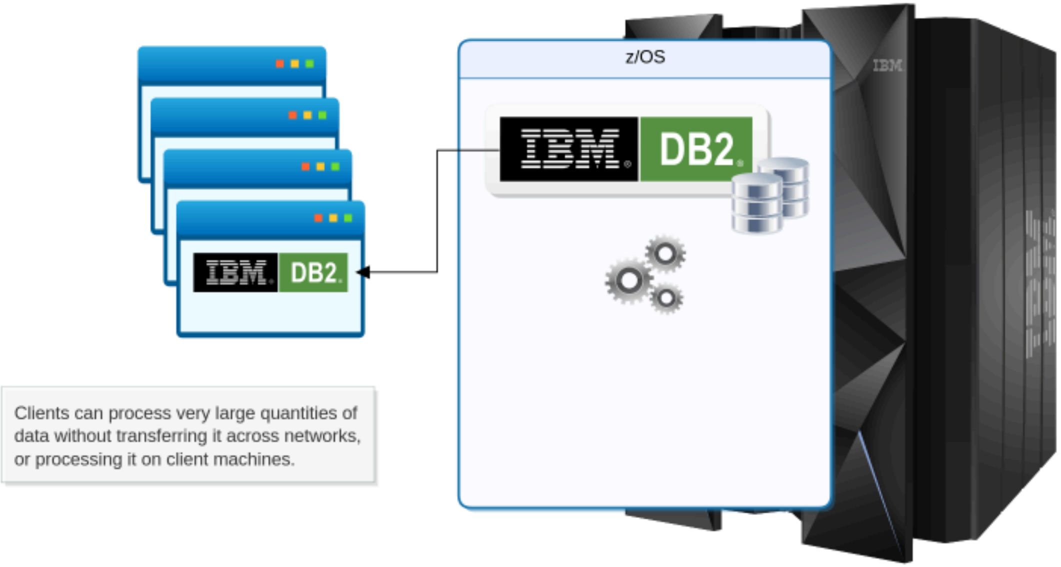
After completing this module, you will able to identify:

- The Benefits of Using Stored Procedures in a z/OS Environment
- How Stored Procedures are Created on z/OS
- Maintenance and Management Aspects Associated With Stored Procedures in a z/OS Environment



Db2 on the mainframe is often used as a central database hub that services many client applications, both on and off the mainframe.

In addition to exceptional availability, security, and performance, Db2 on z/OS has unique features, such as Db2 data-sharing groups. Stored procedures can, therefore, be especially valuable to Db2 on z/OS users and administrators.

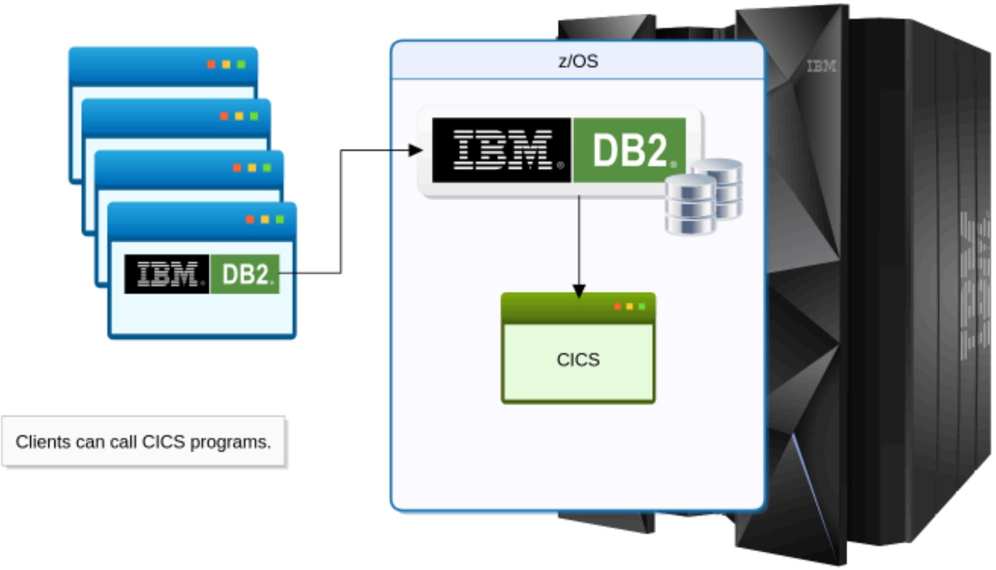


Clients can process very large quantities of data without transferring it across networks, or processing it on client machines.

Db2 stored procedures on z/OS provide the same benefits as other platforms, enhancing the manageability of Db2-based client-server computing.

Click **Play** to see some of the z/OS-specific advantages above.

Features of Stored Procedures on z/OS > Non-Db2 Resources



Clients can call CICS programs.

In addition to accessing Db2 databases, users of Db2 stored procedures can also access other z/OS data, applications, and services.

Click **Play** to see some of these above.

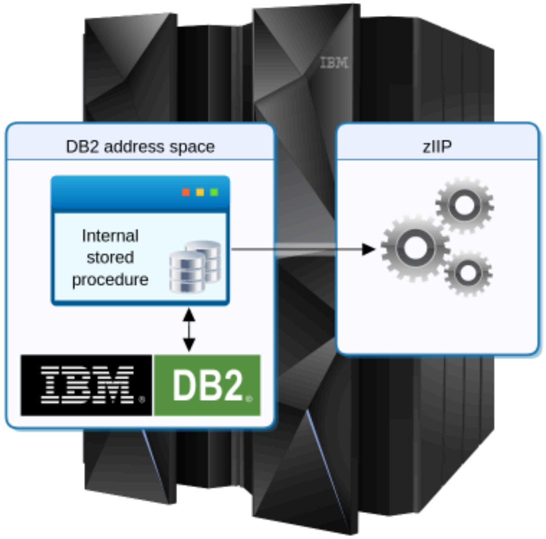
z/OS
Stored procedure
IBM DB2

Internal stored procedure

- No executable or source is stored outside of DB2
- Procedure is written in SQL procedural language (SQL PL)
- Runs in the DB2 DBM1 address space

As with Db2 on other platforms, Db2 stored procedures on z/OS can be internal or external.

Click **Play** to see an example of this concept.

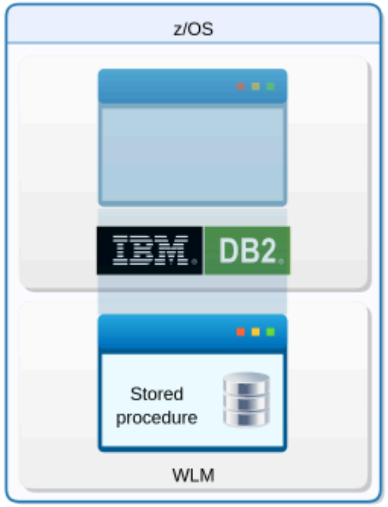


zIIP Eligible
- Reduced software costs
- Reduced use of general purpose processors

When a System z Integrated Information Processor (zIIP) is available, some of the internal stored procedure processing is automatically scheduled on zIIP, which reduces the load on general purpose processors.

zIIP workloads are not included in software licensing costs so performing processing on zIIPs can yield cost savings.

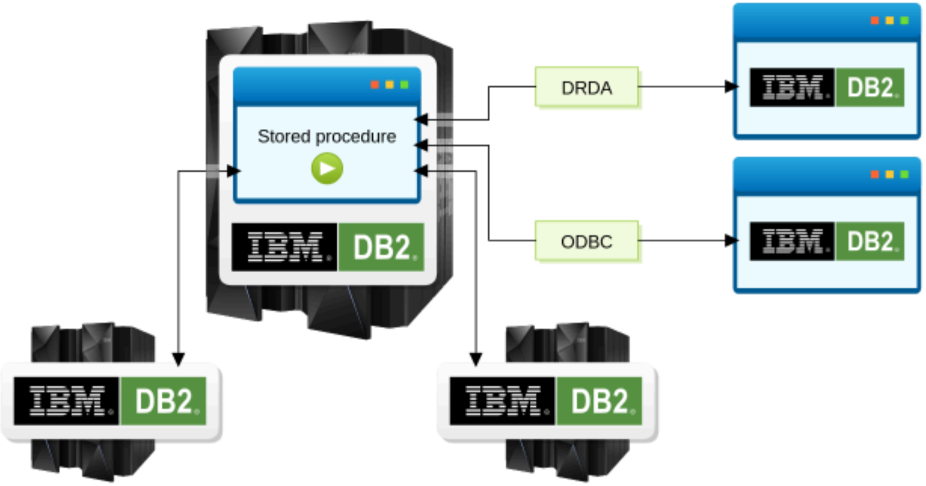
Features of Stored Procedures on z/OS > External Stored Procedures



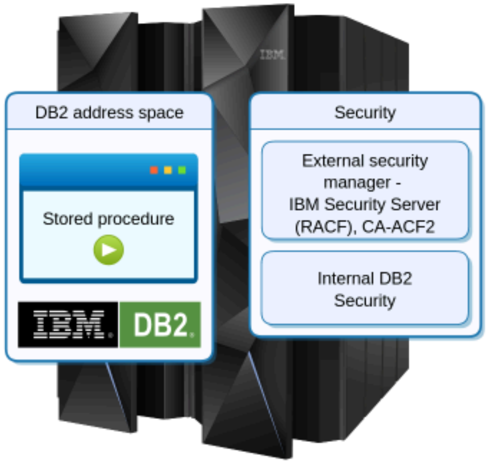
External stored procedure
- Generally slower than internal stored procedures.
- Runs in a separate WLM managed stored procedure address space.

External stored procedures run in separate address spaces that are controlled by z/OS Workload Manager (WLM). These can be written in several different languages.

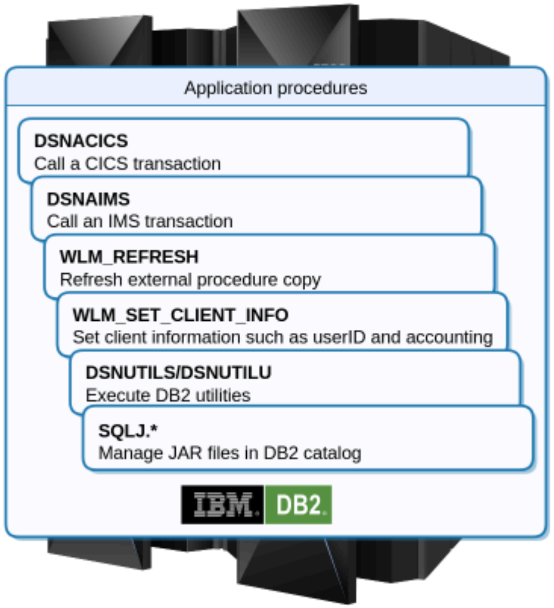
Click Play to see an example of this concept.



Like all Db2 resources, stored procedures on one Db2 system can be accessed by any Db2 subsystem in the same Db2 data-sharing group, or by any Db2 client using Open Database Connectivity (ODBC) or Distributed Relational Database Architecture (DRDA).



All stored procedures are secured in the same way as other Db2 resources, that is, by using internal Db2 security, or by using an external security manager (ESM) such as IBM Security Server (RACF) or CA-ACF2.



Application procedures

DSNACICS
Call a CICS transaction

DSNAIMS
Call an IMS transaction

WLM_REFRESH
Refresh external procedure copy

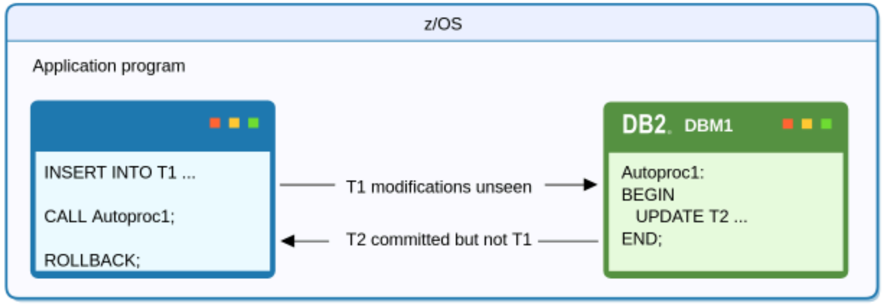
WLM_SET_CLIENT_INFO
Set client information such as userID and accounting

DSNUTILS/DSNUTILU
Execute DB2 utilities

SQLJ.*
Manage JAR files in DB2 catalog

Db2 on z/OS also provides standard stored procedures to perform the following tasks, most of which run as external stored procedures:

- Administer Db2
- Access z/OS data sets
- Control z/OS jobs
- Access CICS and IMS transactions
- Manage JAR files

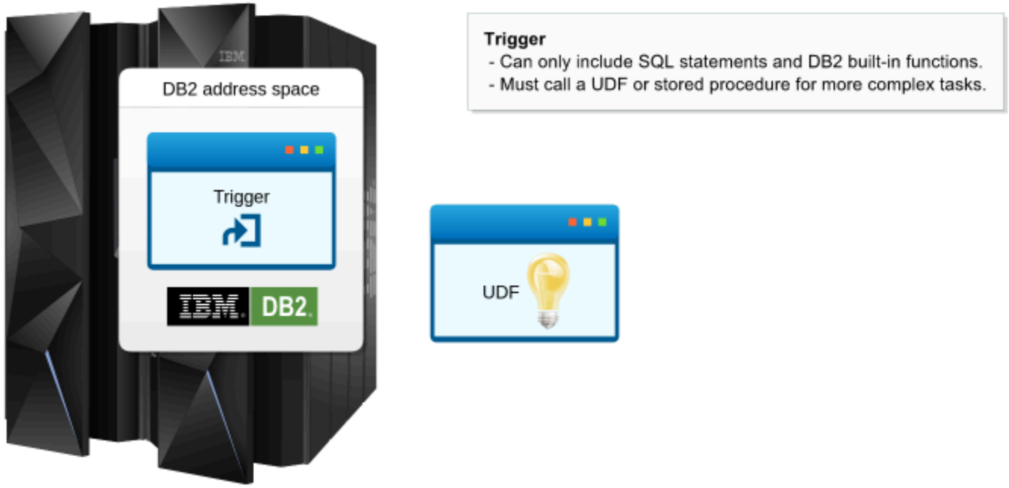


The autonomous stored procedure will have been created like other procedures shown in this module, using the CREATE PROCEDURE statement. The difference is that within the procedure, an AUTONOMOUS property is specified.

```
CREATE PROCEDURE AUTOPROC1
. . .
 LANGUAGE SQL
. . .
 AUTONOMOUS
. . .
END
```

For internal stored procedures, Db2 for z/OS V11 now allows you to create a procedure that performs its task independently of the calling application. This is called an autonomous stored procedure.

It means that the autonomous procedure can perform a task such as updating data to include a message in a report table, and then return control to the calling application. If the caller application then failed with its task, Db2 would only roll back that component's work and not the autonomous stored procedure activity.



Trigger
- Can only include SQL statements and DB2 built-in functions.
- Must call a UDF or stored procedure for more complex tasks.

In some ways, SQL stored procedures are similar to triggers and user-defined functions (UDFs), but, there are important differences.

Click **Play** to see these differences.

Stored procedures are similar to triggers and UDFs in some ways, but there are important differences.

External stored procedures can be written in C/C++, assembler, COBOL, PL/I, REXX, and SQL PL.

Stored procedures are secured in the same way as other DB2 resources.

Internal stored procedures run in the DB2 DBM1 address space, and can run on zIIP processors.

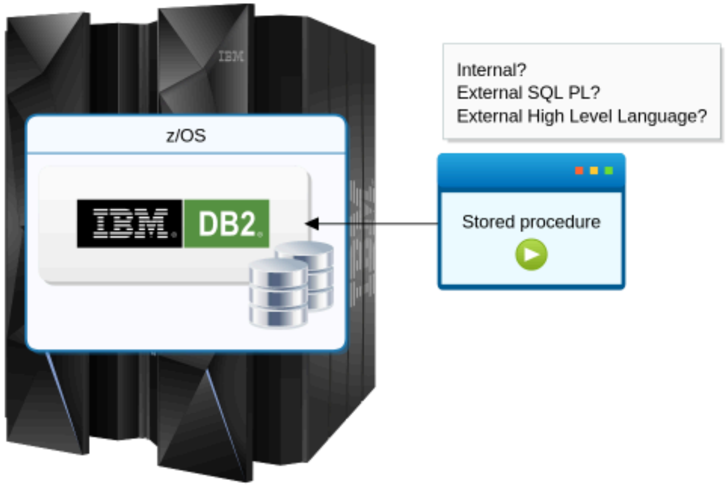
External stored procedures run in separate WLM-managed address spaces.

Stored procedures on z/OS enable programs to access DB2 resources in addition to non-DB2 resources such as IMS and CICS transactions, RRS services, WebSphere MQ Queues, and VSAM and sequential data sets.

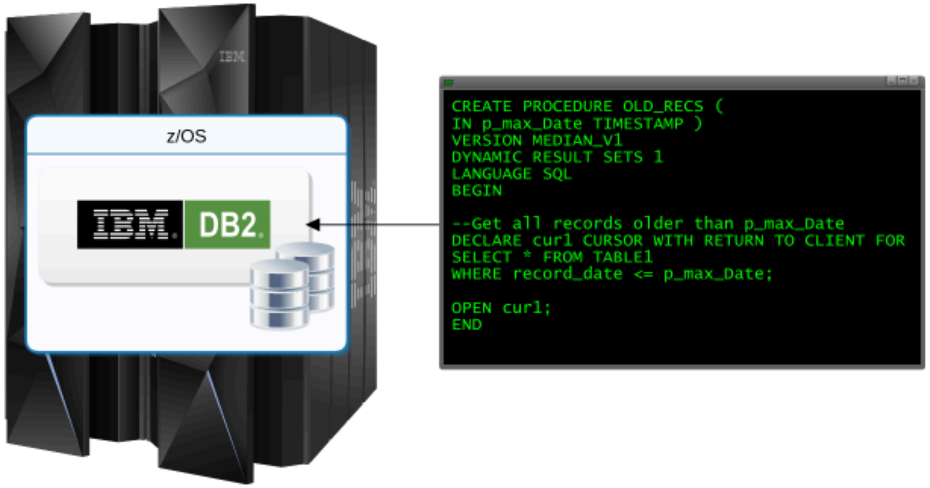
Stored procedures on one DB2 system can be accessed by any DB2 subsystem in the same DB2 data-sharing group, or by any DB2 client using ODBC or DRDA.

You have now been introduced to Db2 stored procedures on z/OS.

Next, you will discover how to create a stored procedure on z/OS.

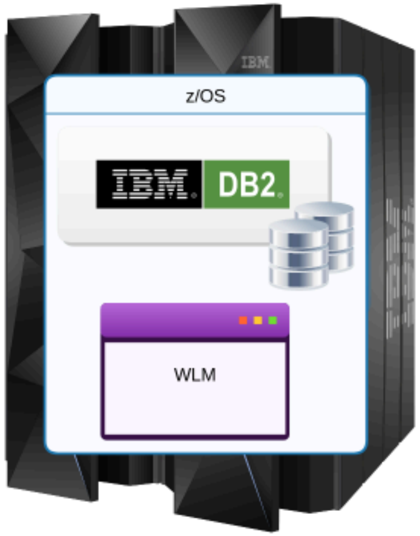


The process of creating a stored procedure on z/OS depends on the type of the stored procedure that is being created.



```
CREATE PROCEDURE OLD_RECS (
IN p_max_Date TIMESTAMP )
VERSION MEDIAN_V1
DYNAMIC RESULT SETS 1
LANGUAGE SQL
BEGIN

--Get all records older than p_max_Date
DECLARE cur1 CURSOR WITH RETURN TO CLIENT FOR
SELECT * FROM TABLE1
WHERE record_date <= p_max_Date;

OPEN cur1;
END
```

An internal stored procedure can be created with a simple SQL CREATE PROCEDURE statement. After creation, it is ready to run and nothing else needs to be done.



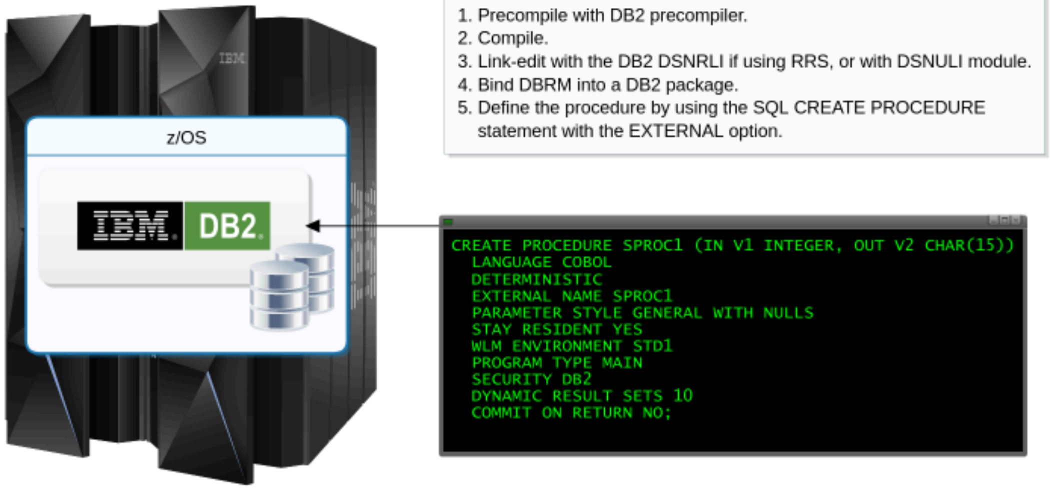
WLM
WLM must be customized to manage the address spaces where the external stored procedures will run.

REXX libraries, Java class libraries, and program libraries must be included in the JCL.

Some Db2 and z/OS customization must be performed before external stored procedures can be run.

Click **Play** to see further examples of this concept.

1. Precompile with DB2 precompiler.
2. Compile.
3. Link-edit with the DB2 DSNRLI if using RRS, or with DSNULI module.
4. Bind DBRM into a DB2 package.
5. Define the procedure by using the SQL CREATE PROCEDURE statement with the EXTERNAL option.

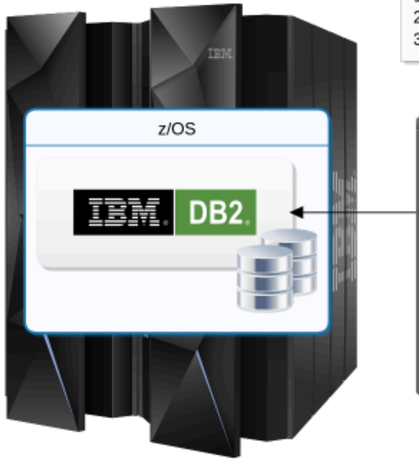


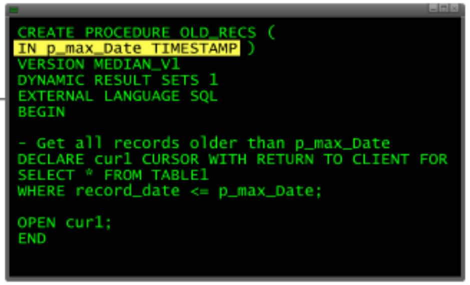
```
CREATE PROCEDURE SPROC1 (IN V1 INTEGER, OUT V2 CHAR(15))
  LANGUAGE COBOL
  DETERMINISTIC
  EXTERNAL NAME SPROC1
  PARAMETER STYLE GENERAL WITH NULLS
  STAY RESIDENT YES
  WLM ENVIRONMENT STD1
  PROGRAM TYPE MAIN
  SECURITY DB2
  DYNAMIC RESULT SETS 10
  COMMIT ON RETURN NO;
```

As with any program that accesses Db2, external stored procedures that are in HLLs, such as COBOL, C, or PL/I, must be compiled and bound.

Preparing re-entrant HLL stored procedures improves their performance. Sample stored procedures are included in the Db2 SDSNSAMP data set.

1. Define the procedure by using the SQL CREATE PROCEDURE statement.
2. Run the DB2 program DSNHPC with HOST(SQL) option.
3. Process the C program created by DSNHPC as for a C stored procedure.



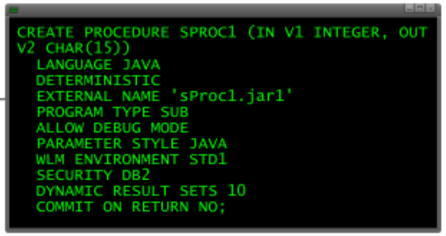


```
CREATE PROCEDURE OLD_RECS (
IN p_max_Date TIMESTAMP )
VERSION MEDIAN_V1
DYNAMIC RESULT SETS 1
EXTERNAL LANGUAGE SQL
BEGIN

- Get all records older than p_max_Date
DECLARE cur1 CURSOR WITH RETURN TO CLIENT FOR
SELECT * FROM TABLE1
WHERE record_date <= p_max_Date;

OPEN cur1;
END
```

To create an external SQL PL procedure, a C program must be created from the SQL PL source. The stored procedure is then created from this C program.

Sample external SQL PL stored procedures, and the jobs that are used to install them, are included in the Db2 SDSNSAMP data set. Db2 also supplies a REXX-stored procedure that can create an external stored procedure from SQL PL: DSNTPSMP. Other software development packages can also be used.

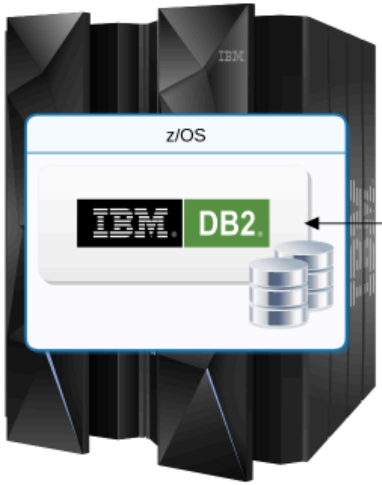
1. Prepare the Java program as for any Java program that accesses DB2.
2. If routine is in a JAR file, define the JAR file to DB2, or include in the CLASSPATH of WLM address space.
3. Define the procedure by using the SQL CREATE PROCEDURE statement.

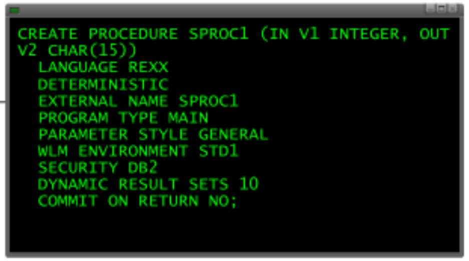


```
CREATE PROCEDURE SPROC1 (IN V1 INTEGER, OUT
V2 CHAR(15))
  LANGUAGE JAVA
  DETERMINISTIC
  EXTERNAL NAME 'sProc1.jar1'
  PROGRAM TYPE SUB
  ALLOW DEBUG MODE
  PARAMETER STYLE JAVA
  WLM ENVIRONMENT STD1
  SECURITY DB2
  DYNAMIC RESULT SETS 10
  COMMIT ON RETURN NO;
```

Java can be used to create external stored procedures. These programs can use JDBC methods, SQLJ clauses, or both. If using SQLJ, a preprocessing step and bind processing is required.

1. Prepare the REXX program as for any REXX program that accesses DB2.
2. Define the procedure by using the SQL CREATE PROCEDURE statement.



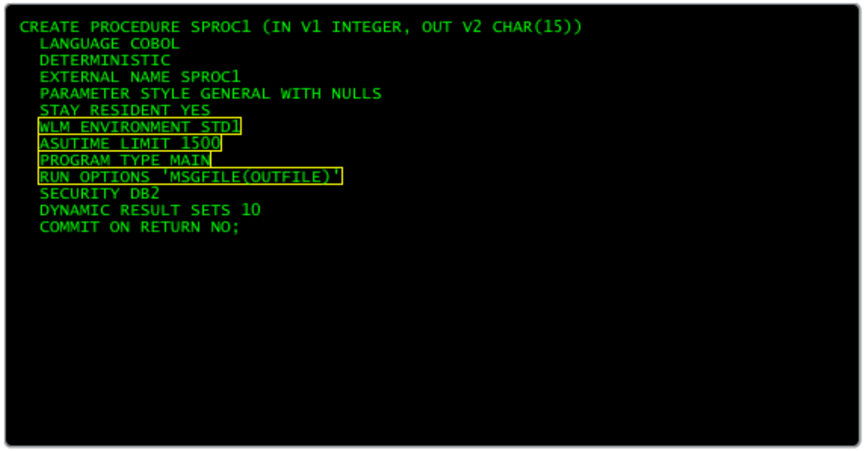


```
CREATE PROCEDURE SPROC1 (IN V1 INTEGER, OUT
V2 CHAR(15))
  LANGUAGE REXX
  DETERMINISTIC
  EXTERNAL NAME SPROC1
  PROGRAM TYPE MAIN
  PARAMETER STYLE GENERAL
  WLM ENVIRONMENT STD1
  SECURITY DB2
  DYNAMIC RESULT SETS 10
  COMMIT ON RETURN NO;
```

REXX can also be used to create external stored procedures. Db2 REXX Language Support must be installed.

REXX stored procedures do not require preparation like other REXX programs accessing Db2 because they are run by using one of the following predefined packages:

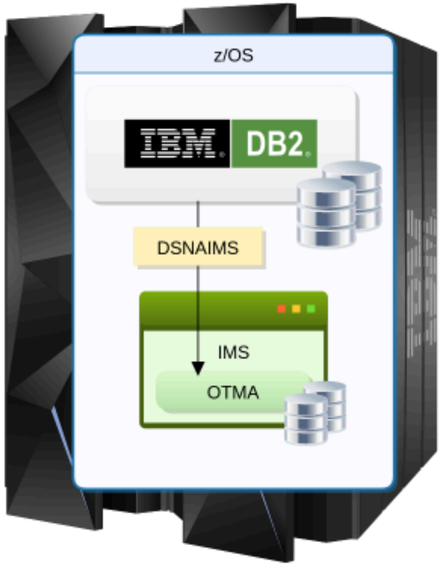
- DSNREXRR - Repeatable read (RR)
- DSNREXRS - Read stability (RS)
- DSNREXCS - Cursor stability (CS)
- DSNREXUR - Uncommitted read (UR)



```
CREATE PROCEDURE SPROC1 (IN V1 INTEGER, OUT V2 CHAR(15))
  LANGUAGE COBOL
  DETERMINISTIC
  EXTERNAL NAME SPROC1
  PARAMETER STYLE GENERAL WITH NULLS
  STAY RESIDENT YES
  WLM ENVIRONMENT STD1
  ASUTIME LIMIT 1500
  PROGRAM TYPE MAIN
  RUN OPTIONS 'MSGFILE(OUTFILE)'
  SECURITY DB2
  DYNAMIC RESULT SETS 10
  COMMIT ON RETURN NO;
```

The z/OS-specific CREATE PROCEDURE options that apply to individual stored procedures are shown above.

Mouse-over these options for more information.

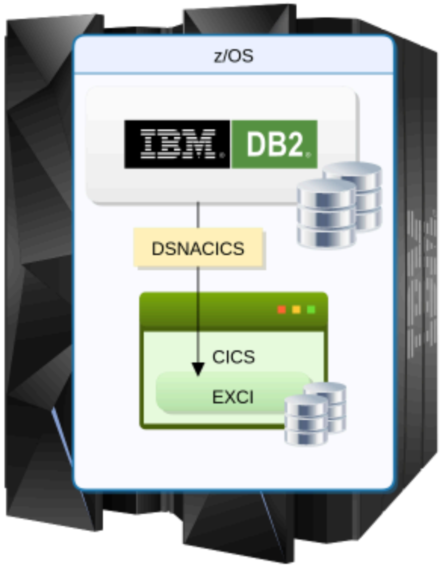


DSNAIMS
The DB2-supplied external stored procedure to call IMS transactions or access IMS databases is DSNAIMS.

DSNAIMS accesses IMS by using IMS Open Transaction Manager Access (OTMA), and requires DB2 RRSAF, IMS 7 or later, and the IMS OTMA command interface to be enabled. IMS must be on the same z/OS system as DB2.

Db2 stored procedures on z/OS can call IMS transactions and access IMS databases in several different ways.

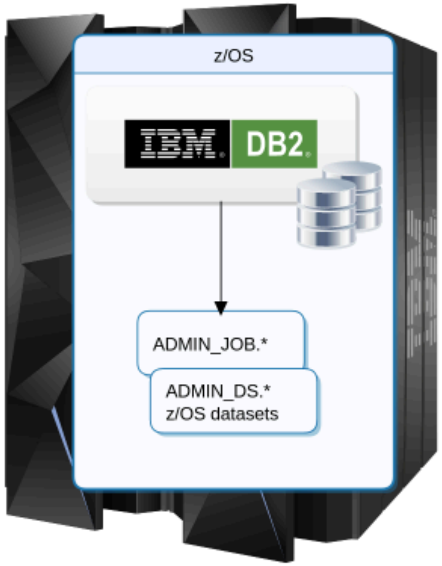
Click **Play** to see a demonstration of this concept.



DSNACICS
DSNACICS is the DB2-supplied external stored procedure to call CICS programs. DSNACICS uses the CICS EXCI interface, and requires DB2 RRSAF. CICS must be on the same z/OS system as DB2.

There are a number of different ways in which stored procedures can call CICS transactions and programs.

Click **Play** to see some examples.



Supplied stored procedures
DB2 supplies stored procedures to access z/OS data sets; and to submit and control z/OS batch jobs.

Db2-supplied stored procedures can be called to access other external z/OS resources, such as z/OS data sets, and also to submit and control z/OS batch jobs.

Click **Play** to see an example of this concept.

IMS transactions and data can be accessed by using the DSNAIMS stored procedure, IMS ODBA, APPC, or WebSphere MQ.

External stored procedures in HLLs, such as COBOL and C, must be prepared in the same way as any HLL routine that accesses DB2.

External SQL PL stored procedures are converted to C programs using DB2 supplied utilities or stored procedures.

CICS programs can be accessed by using the DSNACICS stored procedure, the CICS EXCI interface, APPC, or WebSphere MQ.

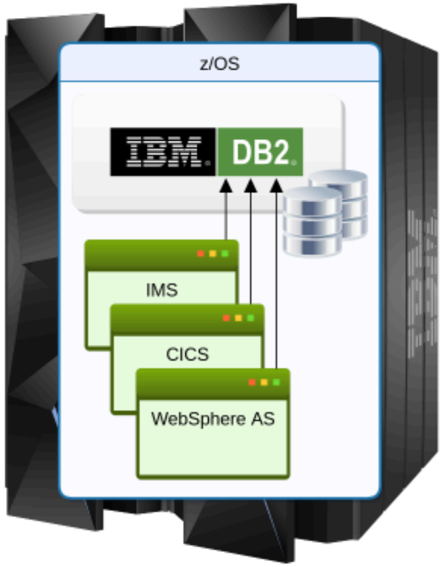
Stored procedures can use MQI calls to access WebSphere MQ.

DB2-supplied stored procedures can be used to access z/OS data sets.

DB2 and z/OS must be configured to enable external stored procedures to run. Internal stored procedures do not require extra configuration.

External stored procedures require WLM to be configured to run the stored procedure address spaces. IBM Language Environment and RRS may also need to be configured.

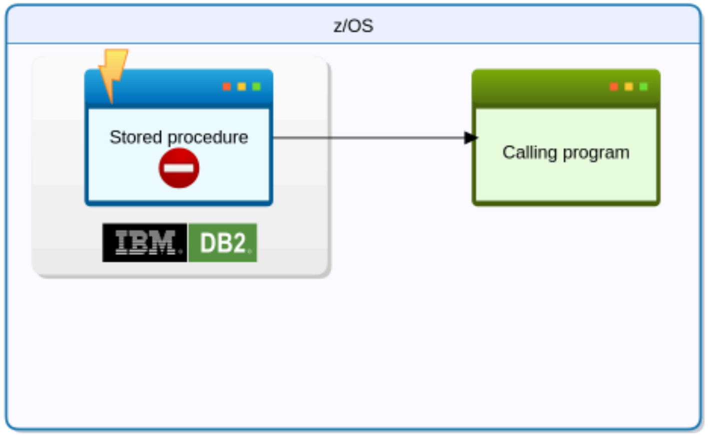
You have now discovered how to create a Db2 stored procedure on z/OS, and how to access z/OS-specific functions. Next, you will look at the administration of Db2 stored procedures.



The work required for Db2 stored procedures does not end after their creation. Administrators must subsequently:

- Detect, diagnose, and fix errors
- Maintain and update the stored procedures
- Control stored procedures

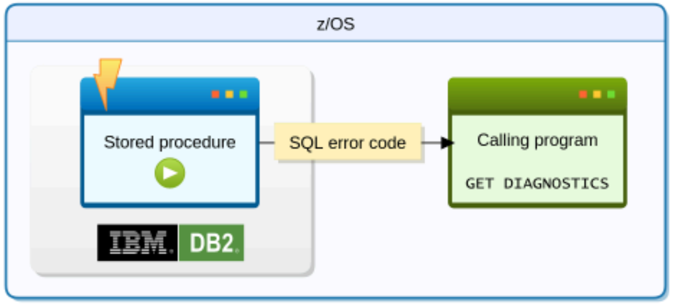
Db2 on z/OS provides several tools that enable administrators to perform these tasks.



Abend processing
3. The stored procedure is stopped if its STOP AFTER n FAILURES value is reached, or if the DB2 subsystem's MAX ABEND COUNT value is reached. The stored procedure must then be manually restarted before it can be called again.

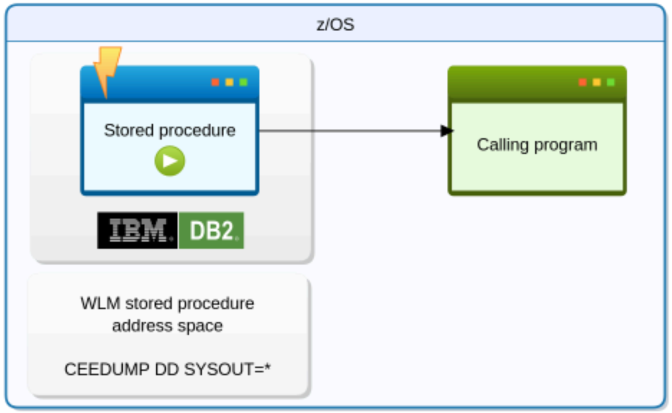
If a stored procedure fails, an SQL error code is returned to the caller.

Click **Play** to see the other actions that Db2 performs for stored procedures that abend.



Calling programs can use the SQL return code and the SQL GET DIAGNOSTICS statement to identify and handle errors. External stored procedures running under LE can also output error messages to the SYSOUT or MSGFILE DD data sets.

Development tools such as the Db2 Unified Debugger can be used to debug stored procedures.

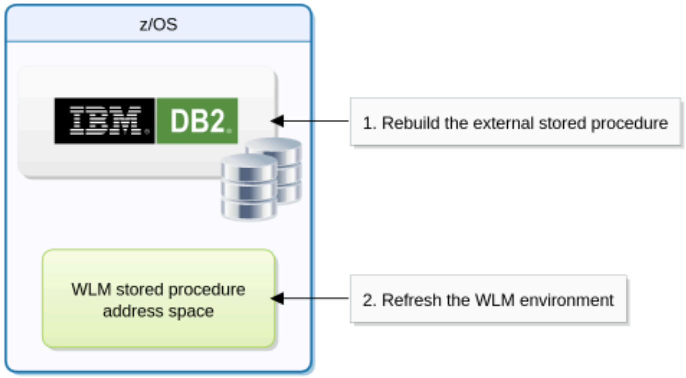


Abend processing
4. If running under LE, a dump is written to the CEEDUMP DD, if it exists.

Definitions for all stored procedures can be altered by using the SQL ALTER PROCEDURE statement. This statement can also be used to update the SQL body of an internal stored procedure.

The SQL DROP SQL statement removes the stored procedure definition and all associated Db2 packages from the Db2 catalog.

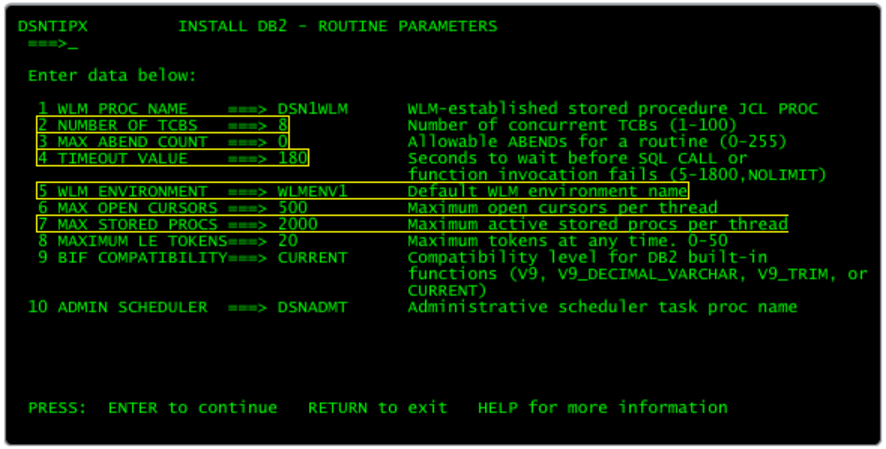
Click **Play** to see an example of this concept.


z/OS
IBM DB2
1. Rebuild the external stored procedure
WLM stored procedure address space
2. Refresh the WLM environment

To update an external stored procedure's code, the procedure must be rebuilt. For example, a COBOL program must be precompiled, compiled, and bound, and the DBRM must be rebound into a package.

Additionally, the WLM environment must be refreshed by either:

- Calling the Db2 WLM_REFRESH Stored Procedure. This is the preferred method.
- Issuing the z/OS command `V WLM,APPLENV=name,REFRESH`. This command stops existing stored procedure address spaces, and creates new ones.

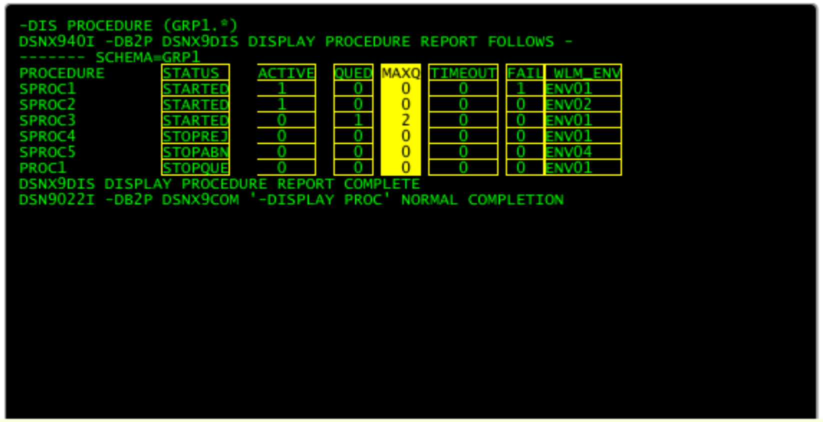


```
DSNTIPX         INSTALL DB2 - ROUTINE PARAMETERS
 ===>_

 Enter data below:

  1 WLM PROC NAME     ===> DSN1WLM      WLM-established stored procedure JCL PROC
  2 NUMBER OF TCBS    ===> 8            Number of concurrent TCBs (1-100)
  3 MAX ABEND COUNT   ===> 0            Allowable ABENDs for a routine (0-255)
  4 TIMEOUT VALUE     ===> 180          Seconds to wait before SQL CALL or
                                        function invocation fails (5-1800,NOLIMIT)
  5 WLM ENVIRONMENT   ===> WLMENV1      Default WLM environment name
  6 MAX OPEN CURSORS  ===> 500          Maximum open cursors per thread
  7 MAX STORED PROCS  ===> 2000         Maximum active stored procs per thread
  8 MAXIMUM LE TOKENS===> 20            Maximum tokens at any time. 0-50
  9 BIF COMPATIBILITY===> CURRENT       Compatibility level for DB2 built-in
                                        functions (V9, V9_DECIMAL_VARCHAR, V9_TRIM, or
                                        CURRENT)
 10 ADMIN SCHEDULER   ===> DSNADMT      Administrative scheduler task proc name




 PRESS:  ENTER to continue   RETURN to exit   HELP for more information
```

When defining a stored procedure by using the SQL CREATE PROCEDURE statement, parameters can be set to control the execution environment of the individual stored procedure.

Db2 also has several parameters that control how all stored procedures are executed. These parameters are set and changed by using the DSNTIPX panel in the Db2 installation CLIST.

Mouse-over the graphic for more information.



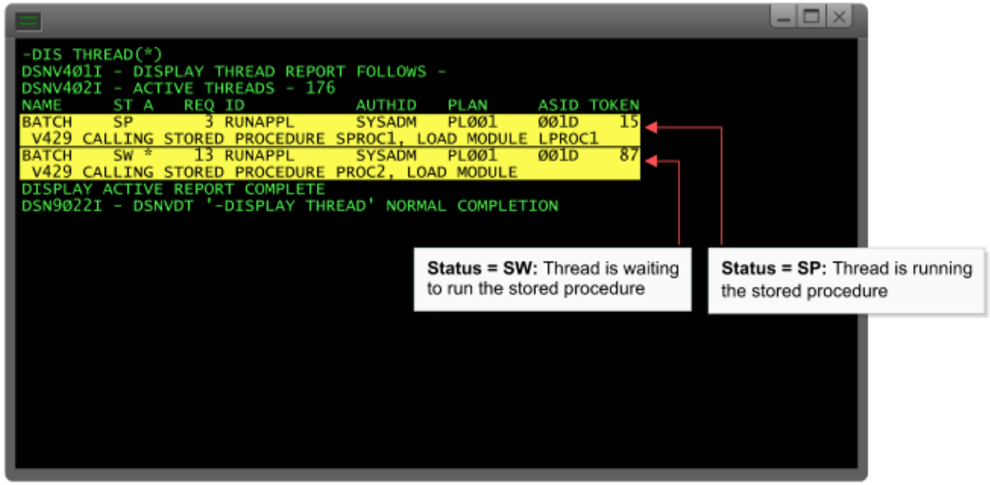
```
-DIS PROCEDURE (GRP1.*)
DSNX940I -DB2P DSNX9DIS DISPLAY PROCEDURE REPORT FOLLOWS -
------- SCHEMA=GRP1
PROCEDURE      STATUS    ACTIVE  QUED MAXQ TIMEOUT FAIL WLM_ENV
SPROC1         STARTED     1       0    0     0     1  ENV01
SPROC2         STARTED     1       0    0     0     0  ENV02
SPROC3         STARTED     0       1    2     0     0  ENV01
SPROC4         STOPREJ     0       0    0     0     0  ENV01
SPROC5         STOPABN     0       0    0     0     0  ENV04
PROC1          STOPQUE     0       0    0     0     0  ENV01
DSNX9DIS DISPLAY PROCEDURE REPORT COMPLETE
DSN9022I -DB2P DSNX9COM '-DISPLAY PROC' NORMAL COMPLETION
```

MaxQ: Highest number of threads that were simultaneously waiting to run the procedure since Db2 was started.

Information about individual stored procedures can be displayed by using the Db2 DISPLAY PROCEDURE command. In the above example, all stored procedures in the GRP1 schema are displayed.

Stored procedures can be stopped by using the Db2 STOP PROCEDURE command. This will stop any new requests, but will allow existing requests to finish. Stored procedures can be restarted by using the Db2 START PROCEDURE command.

Mouse-over the command output for more information.

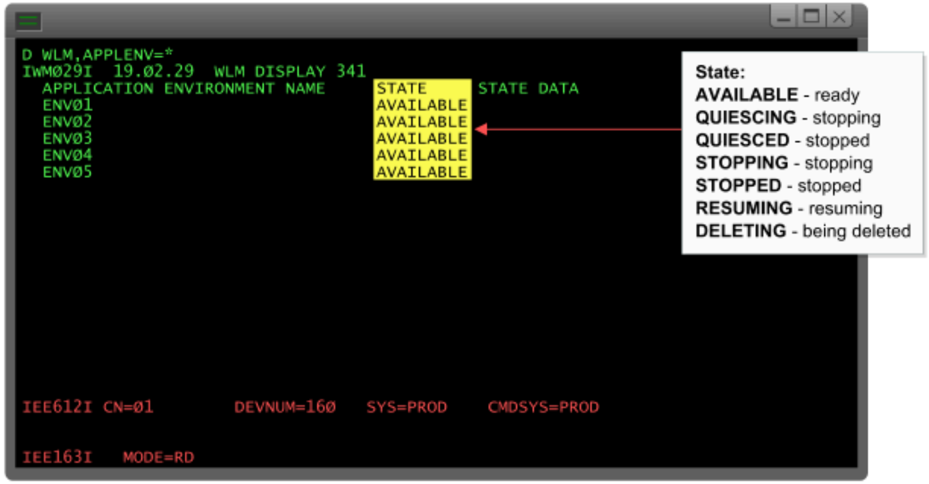


```
-DIS THREAD(*)
DSNV4Ø1I - DISPLAY THREAD REPORT FOLLOWS -
DSNV4Ø2I - ACTIVE THREADS - 176
NAME     ST A   REQ ID          AUTHID   PLAN     ASID TOKEN
BATCH    SP       3 RUNAPPL     SYSADM   PLØØ1    ØØ1D    15
 V429 CALLING STORED PROCEDURE SPROC1, LOAD MODULE LPROC1
BATCH    SW *    13 RUNAPPL     SYSADM   PLØØ1    ØØ1D    87
 V429 CALLING STORED PROCEDURE PROC2, LOAD MODULE
DISPLAY ACTIVE REPORT COMPLETE
DSN9Ø22I - DSNVDT '-DISPLAY THREAD' NORMAL COMPLETION
```

Status = SW: Thread is waiting to run the stored procedure

Status = SP: Thread is running the stored procedure

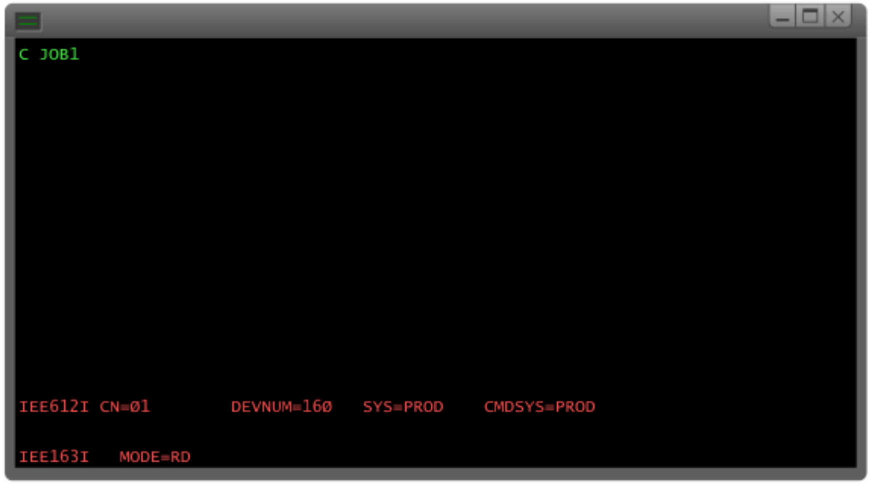
The Db2 DISPLAY THREAD command shows information about threads that are running a stored procedure, or waiting for a stored procedure.



```
D WLM,APPLENV=*
IWM029I  19.02.29  WLM DISPLAY 341
  APPLICATION ENVIRONMENT NAME          STATE      STATE DATA
  ENV01                                 AVAILABLE
  ENV02                                 AVAILABLE
  ENV03                                 AVAILABLE
  ENV04                                 AVAILABLE
  ENV05                                 AVAILABLE




IEE612I CN=01        DEVNUM=160   SYS=PROD    CMDSYS=PROD

IEE163I   MODE=RD
```

State:
AVAILABLE - ready
QUIESCING - stopping
QUIESCED - stopped
STOPPING - stopping
STOPPED - stopped
RESUMING - resuming
DELETING - being deleted

The z/OS console command shown above can be used to display the status of an application environment in which a stored procedure runs.



```
C JOB1





IEE612I CN=01        DEVNUM=160    SYS=PROD     CMDSYS=PROD

IEE163I    MODE=RD
```

Method 2
Cancel the calling job or transaction, if possible.

Occasionally, a stored procedure must be cancelled manually if, for example, it is looping and it was created with ASUTIME = NOLIMIT, which is the default ASUTIME value.

Click **Play** to see the three different methods of cancelling a stored procedure.

How to Maintain and Control Stored Procedures > Summary

To alter an external stored procedure, the stored procedure module must be rebuilt and the WLM environment refreshed.

Individual stored procedure execution parameters are specified by using the CREATE PROCEDURE and ALTER PROCEDURE SQL statements.

The body of an internal stored procedure can be updated by using the SQL ALTER PROCEDURE REPLACE statement.

Stored procedure execution parameters can be changed by using the SQL ALTER PROCEDURE statement.

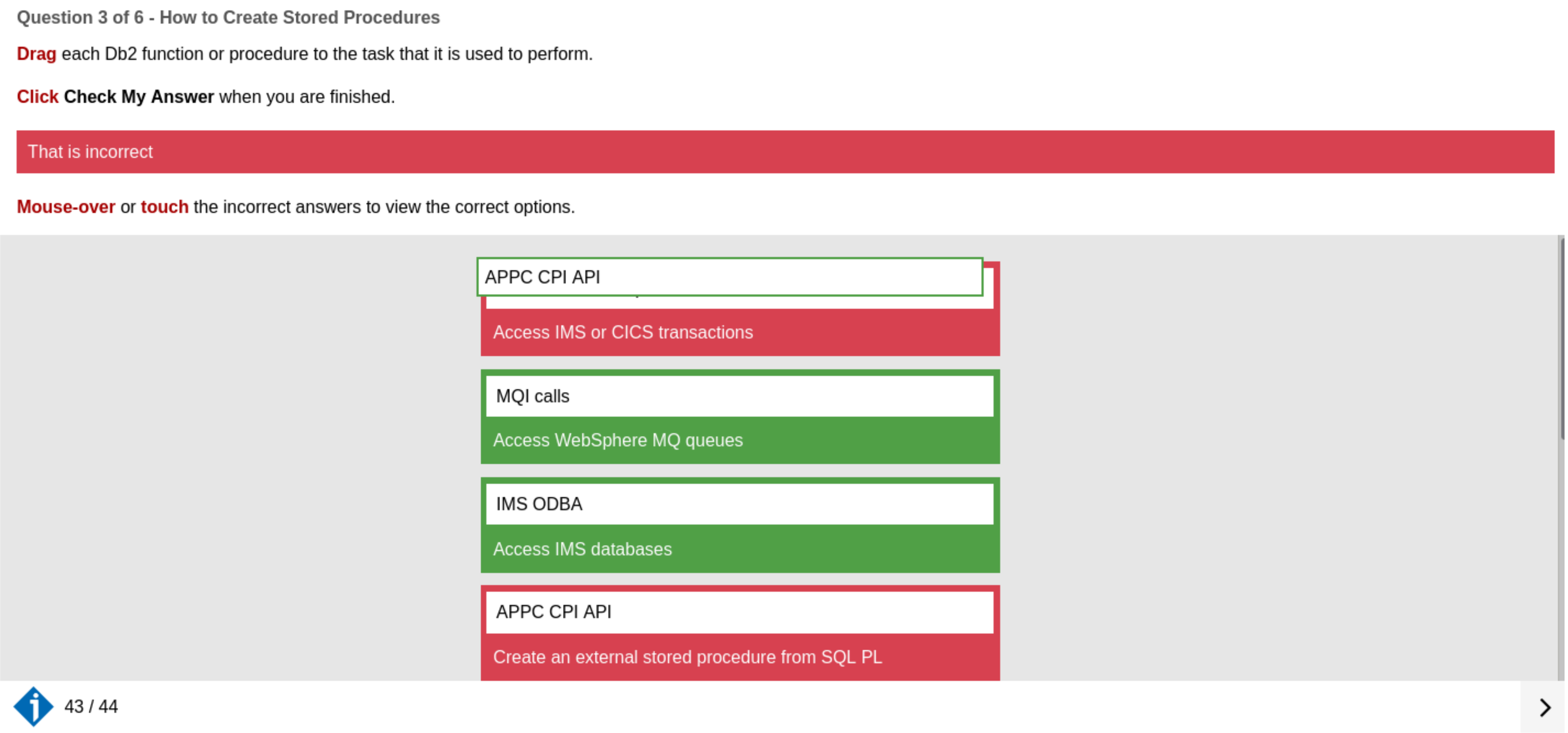
Parameters for all stored procedures are specified in the DSNTIPX panel of the DB2 install CLIST.

Stored Procedures are removed using the SQL DROP statement.

DB2 stored procedures return an SQL error code to the calling routine. The SQL GET DIAGNOSTICS statement is used to obtain more information. You should check the success or failure of every stored procedure call.

The DB2 commands DISPLAY PROCEDURE and DISPLAY THREAD can be used to display stored procedure status and statistics. The D WLM z/OS command displays the status of an application environment in which stored procedures run.

You have now discovered how to maintain and control stored procedures.



Question 3 of 6 - How to Create Stored Procedures

Drag each Db2 function or procedure to the task that it is used to perform.

Click **Check My Answer** when you are finished.

That is incorrect

Mouse-over or **touch** the incorrect answers to view the correct options.

APPC CPI API	Access IMS or CICS transactions
MQI calls	Access WebSphere MQ queues
IMS ODBA	Access IMS databases
APPC CPI API	Create an external stored procedure from SQL PL

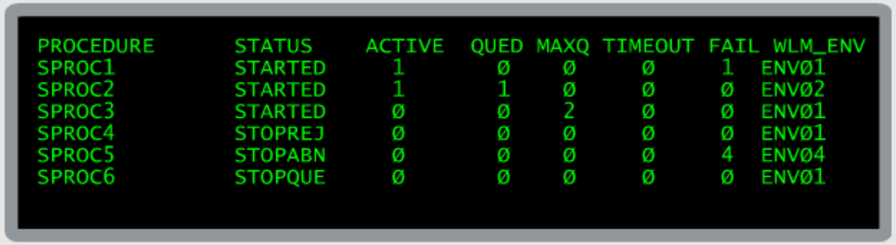
Question 5 of 6 - How to Maintain and Control Stored Procedures

Which four statements describing this output from a Db2 command are true?

Select the correct options.

Click Check My Answer when you have finished.

That is correct!



```
PROCEDURE       STATUS    ACTIVE  QUED MAXQ TIMEOUT FAIL WLM_ENV
SPROC1          STARTED     1       0    0     0      1  ENV01
SPROC2          STARTED     1       1    0     0      0  ENV02
SPROC3          STARTED     0       0    2     0      0  ENV01
SPROC4          STOPREJ     0       0    0     0      0  ENV01
SPROC5          STOPABN     0       0    0     0      4  ENV04
SPROC6          STOPQUE     0       0    0     0      0  ENV01
```

☑ The output is from the Db2 DIS PROCEDURE command.

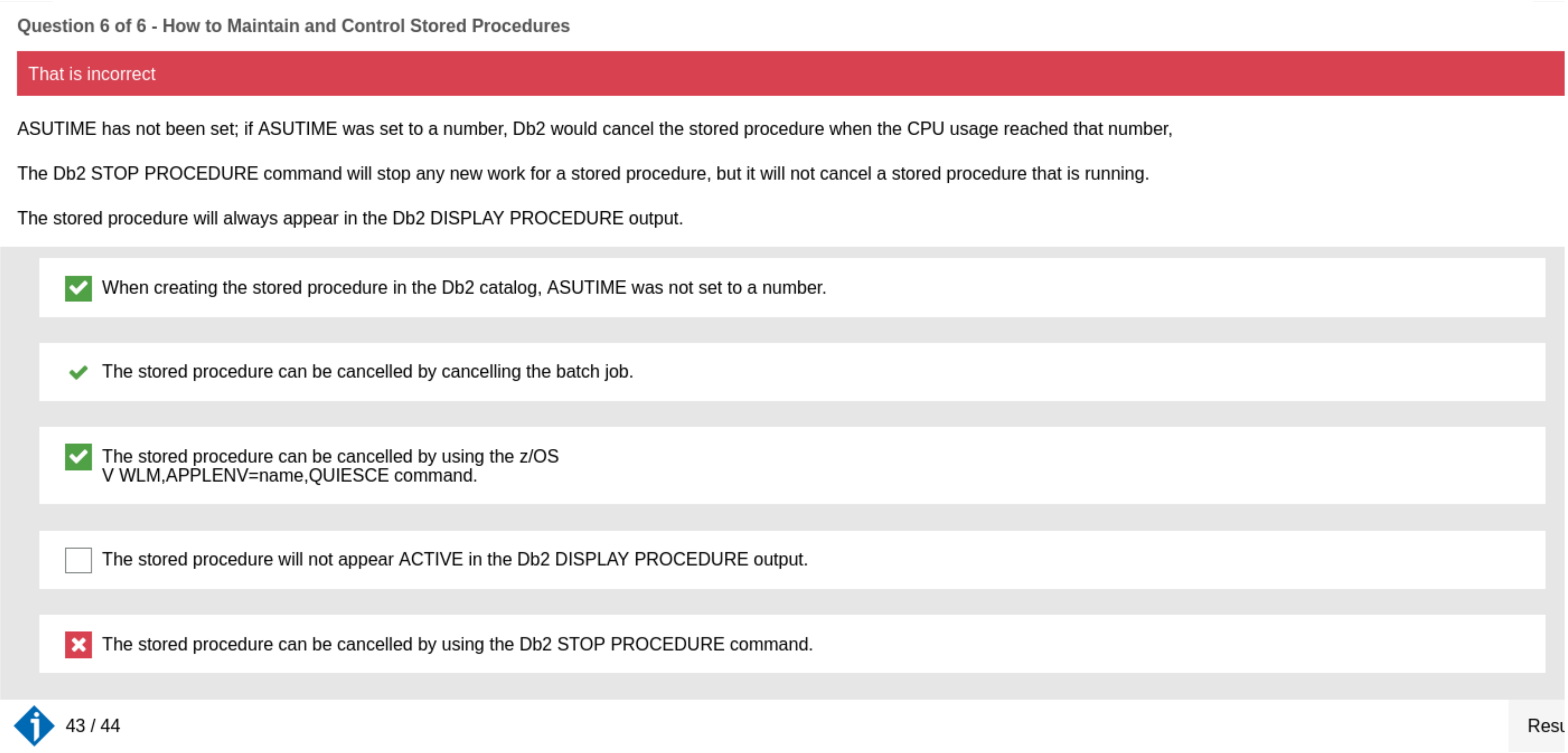
□ A call to SPROC4 or SPROC5 will be rejected.

□ The output is from the z/OS D WLM,APPLENV=* command.

☑ SPROC4 and SPROC6 have been stopped by using the Db2 STOP PROCEDURE command.

☑ SPROC4, SPROC5, and SPROC6 must be activated with the Db2 START PROCEDURE command.

☑ SPROC5 has exceeded the STOP AFTER n FAILURES value or the Db2 MAX ABEND COUNT.



Question 6 of 6 - How to Maintain and Control Stored Procedures

That is incorrect

ASUTIME has not been set; if ASUTIME was set to a number, Db2 would cancel the stored procedure when the CPU usage reached that number,

The Db2 STOP PROCEDURE command will stop any new work for a stored procedure, but it will not cancel a stored procedure that is running.

The stored procedure will always appear in the Db2 DISPLAY PROCEDURE output.

☑ When creating the stored procedure in the Db2 catalog, ASUTIME was not set to a number.

✓ The stored procedure can be cancelled by cancelling the batch job.

☑ The stored procedure can be cancelled by using the z/OS
V WLM,APPLENV=name,QUIESCE command.

□ The stored procedure will not appear ACTIVE in the Db2 DISPLAY PROCEDURE output.

☒ The stored procedure can be cancelled by using the Db2 STOP PROCEDURE command.