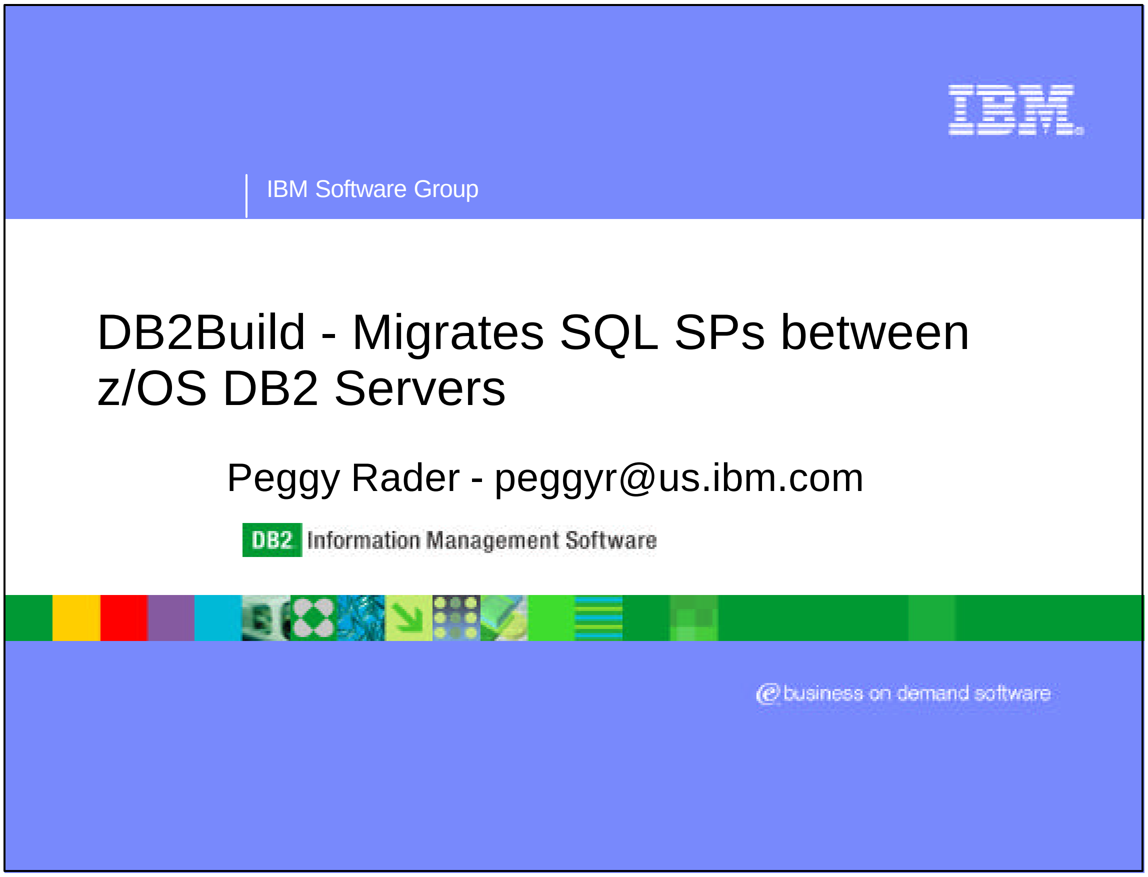


IBM Software Group

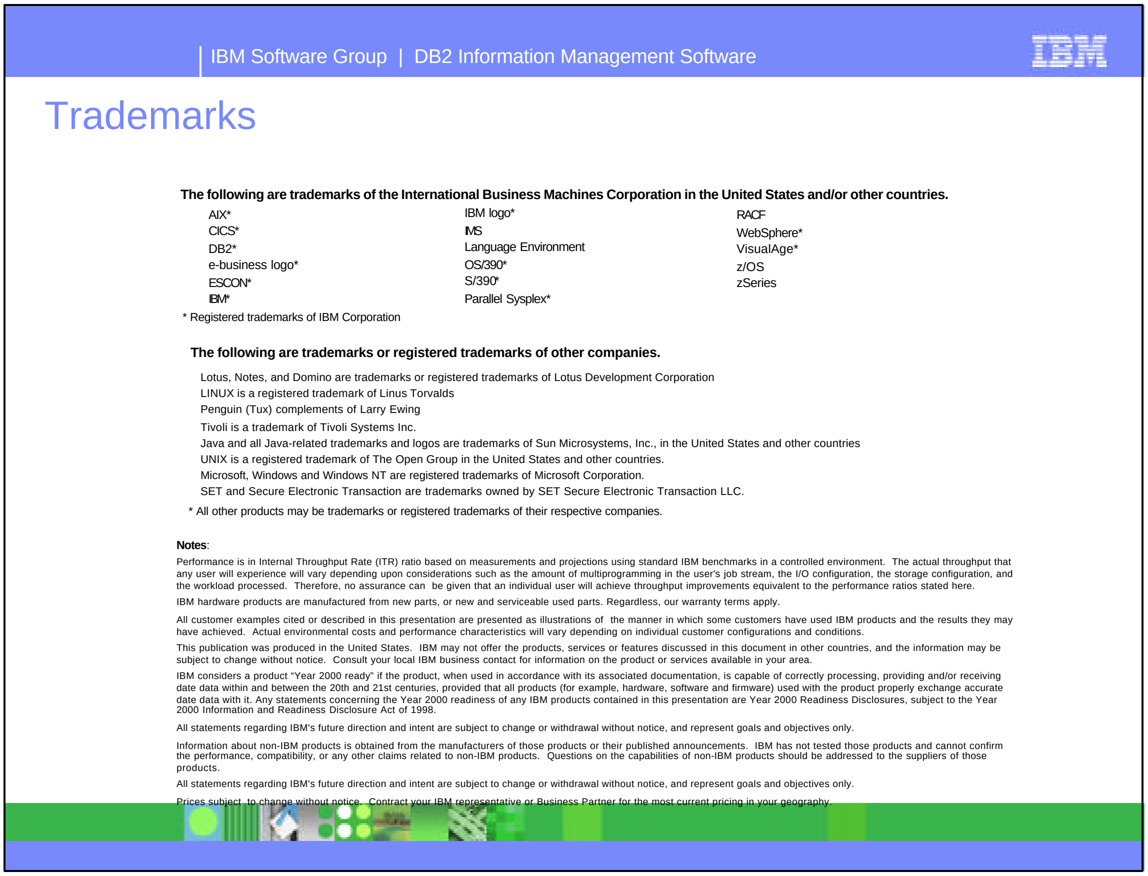
DB2Build - Migrates SQL SPs between z/OS DB2 Servers

Peggy Rader - peggyr@us.ibm.com

DB2 Information Management Software

business on demand software

- The DB2Build Batch Utility provides a command line interface to migrate SQL Stored Procedures created on one DB2 z/OS Server to another DB2 z/OS Server. This batch utility can be run from a DB2 UDB V8 client and supports DB2 V6, V7 and V8 on z/OS servers. This session will describe the setup steps including: DB2Build processing flow, Pre-reqs, Workstation and z/OS setup, DB2Build utility command options and Examples.



Trademarks

The following are trademarks of the International Business Machines Corporation in the United States and/or other countries.

AIX*
CICS*
DB2*
e-business logo*
ESCON*
IBM*
IBM logo*
IMS
Language Environment
OS/390*
S/390*
Parallel Sysplex*
RACF
WebSphere*
VisualAge*
z/OS
zSeries

* Registered trademarks of IBM Corporation

The following are trademarks or registered trademarks of other companies.

Lotus, Notes, and Domino are trademarks or registered trademarks of Lotus Development Corporation
LINUX is a registered trademark of Linus Torvalds
Penguin (Tux) complements of Larry Ewing
Tivoli is a trademark of Tivoli Systems Inc.
Java and all Java-related trademarks and logos are trademarks of Sun Microsystems, Inc., in the United States and other countries
UNIX is a registered trademark of The Open Group in the United States and other countries.
Microsoft, Windows and Windows NT are registered trademarks of Microsoft Corporation.
SET and Secure Electronic Transaction are trademarks owned by SET Secure Electronic Transaction LLC.

* All other products may be trademarks or registered trademarks of their respective companies.

Notes:

Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.

IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.

All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.

This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.

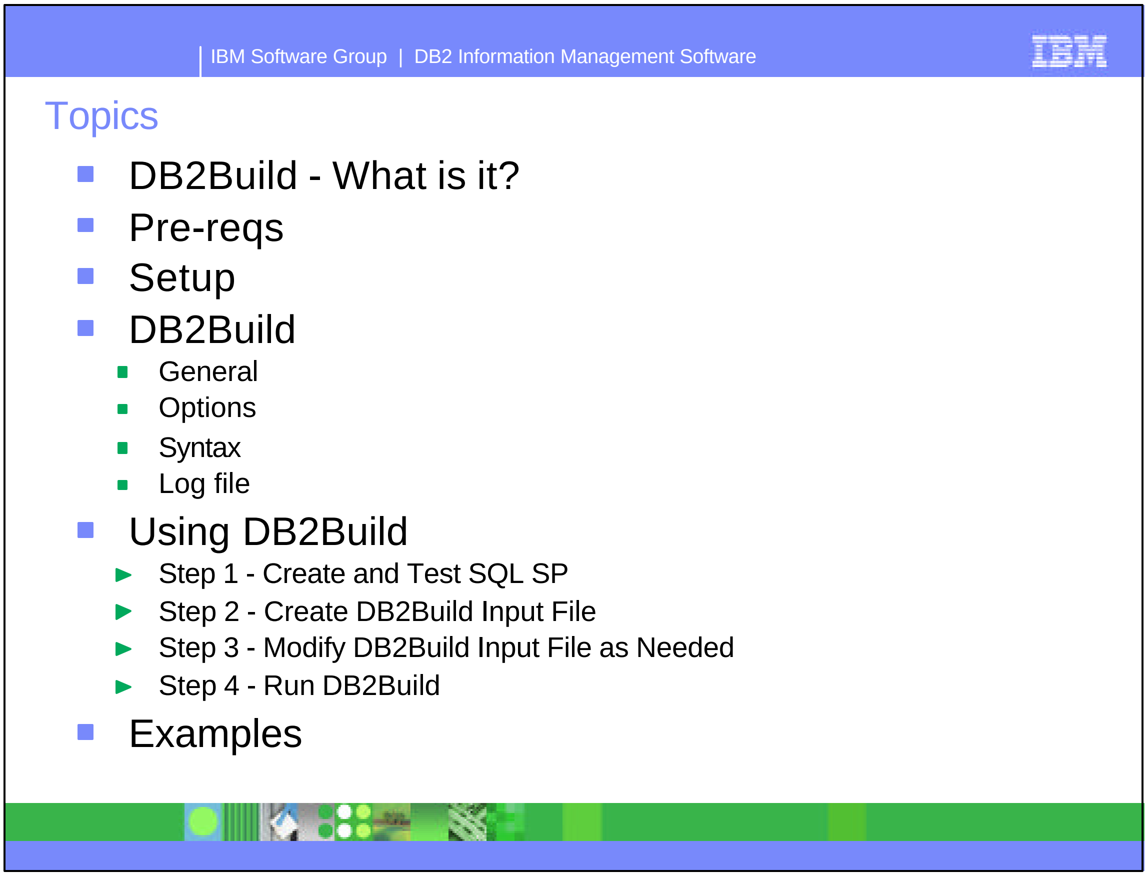
IBM considers a product "Year 2000 ready" if the product, when used in accordance with its associated documentation, is capable of correctly processing, providing and/or receiving date data within and between the 20th and 21st centuries, provided that all products (for example, hardware, software and firmware) used with the product properly exchange accurate date data with it. Any statements concerning the Year 2000 readiness of any IBM products contained in this presentation are Year 2000 Readiness Disclosures, subject to the Year 2000 Information and Readiness Disclosure Act of 1998.

All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

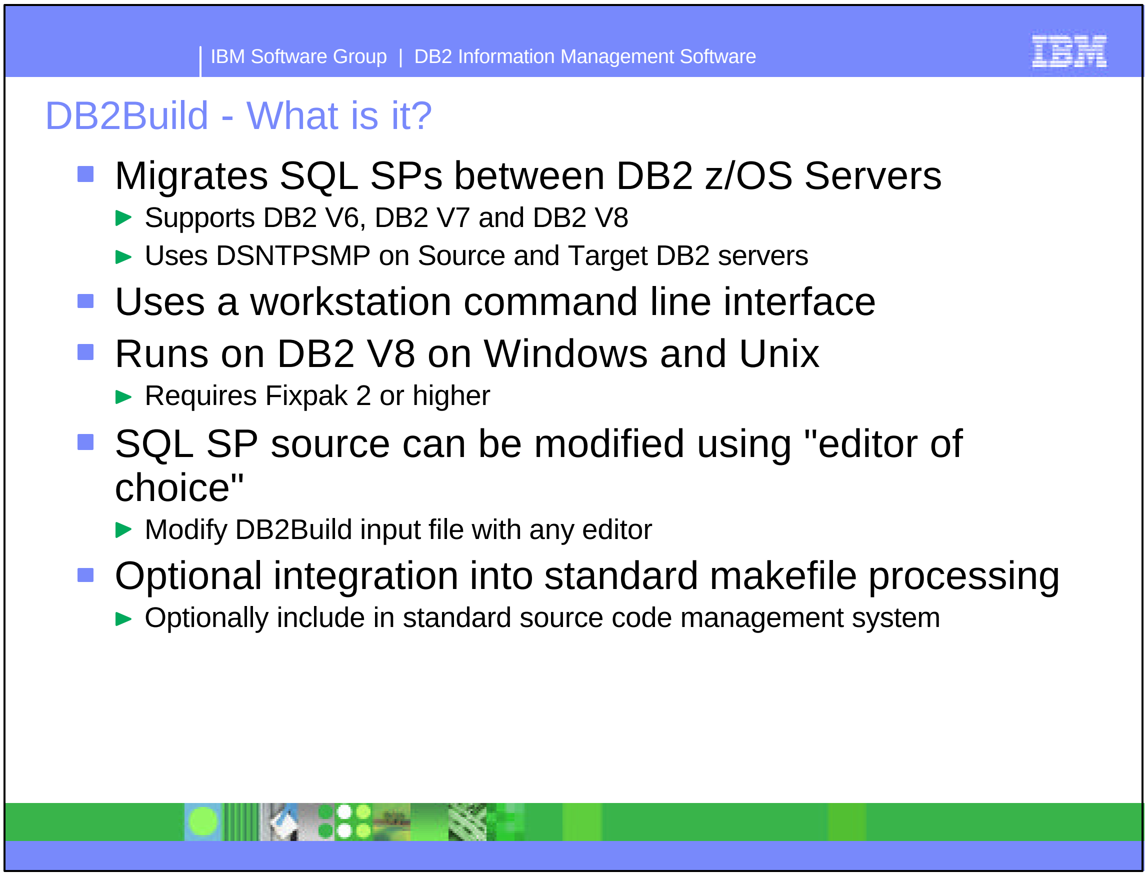
Prices subject to change without notice. Contract your IBM representative or Business Partner for the most current pricing in your geography.



Topics

- DB2Build - What is it?
- Pre-reqs
- Setup
- DB2Build
  - General
  - Options
  - Syntax
  - Log file
- Using DB2Build
  - Step 1 - Create and Test SQL SP
  - Step 2 - Create DB2Build Input File
  - Step 3 - Modify DB2Build Input File as Needed
  - Step 4 - Run DB2Build
- Examples

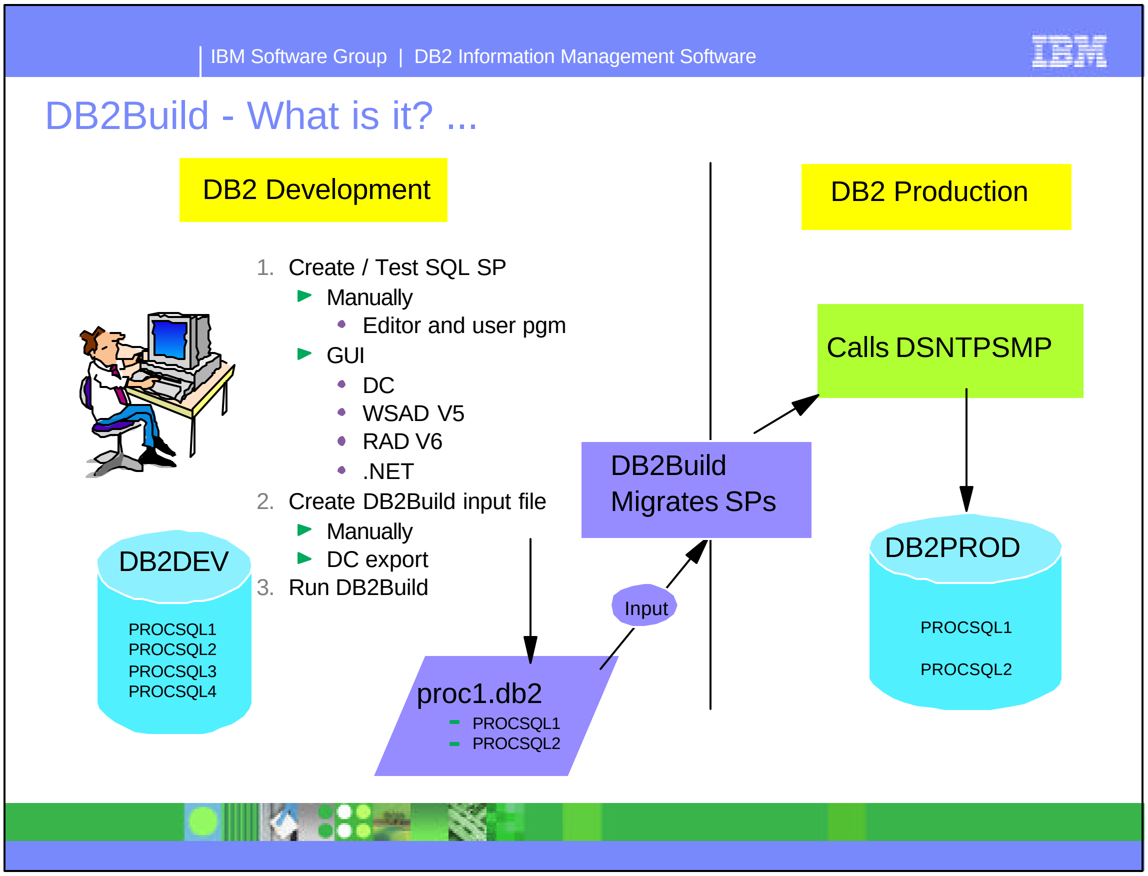
- The topics covered in this presentation include:
  - Describing what the DB2Build Batch Utility is
  - The Pre-reqs to use the DB2Build utility
  - Setup steps for both the workstation and the z/OS DB2 Servers
  - DB2Build processing including
    - General information
    - Describing the various DB2Build options and the syntax used to execute these options
    - The log file that is produced and the information that is written to the log
  - Using DB2Build requires the following steps to be completed
    - 1. First an SQL SP has to have previously been created and tested
    - 2. Create the DB2Build input file, either manually or through exporting an existing SQL SP using the DB2 V8 Development Center.
    - 3. Optionally, modify the DB2Build input file if needed.
    - 4. Run DB2Build
  - The presentation concludes with examples of the Drop, Create, Create force, Alter_source and Rebind actions.



DB2Build - What is it?

- Migrates SQL SPs between DB2 z/OS Servers
  - Supports DB2 V6, DB2 V7 and DB2 V8
  - Uses DSNTPSMP on Source and Target DB2 servers
- Uses a workstation command line interface
- Runs on DB2 V8 on Windows and Unix
  - Requires Fixpak 2 or higher
- SQL SP source can be modified using "editor of choice"
  - Modify DB2Build input file with any editor
- Optional integration into standard makefile processing
  - Optionally include in standard source code management system

- DB2Build is a batch utility that migrates SQL SPs between DB2 z/OS servers. It supports DB2 V6, V7 and V8 on z/OS.
- The utility uses the DSNTPSMP Rexx Stored procedure on both the Source and the Target DB2 servers.
- The utility commands are entered from a workstation command line interface.
- The utility runs on DB2 UDB V8 on both Windows and Unix where the code is included in Fixpak 2..
- The SQL SP source can be modified after it has been exported from the source DB2 on z/OS server using DB2 V8 Development Center (DC) before it is processed by the utility to the target DB2 on z/OS server.
- Optionally, the source can be integrated into your standard makefile processing and into any source code management system that supports the makefile.



IBM Software Group | DB2 Information Management Software

DB2Build - What is it? ...

DB2 Development

1. Create / Test SQL SP
   - Manually
     - Editor and user pgm
   - GUI
     - DC
     - WSAD V5
     - RAD V6
     - .NET
2. Create DB2Build input file
   - Manually
   - DC export
3. Run DB2Build

DB2DEV

PROCSQL1
PROCSQL2
PROCSQL3
PROCSQL4

proc1.db2
- PROCSQL1
- PROCSQL2

Input

DB2Build
Migrates SPs

DB2 Production

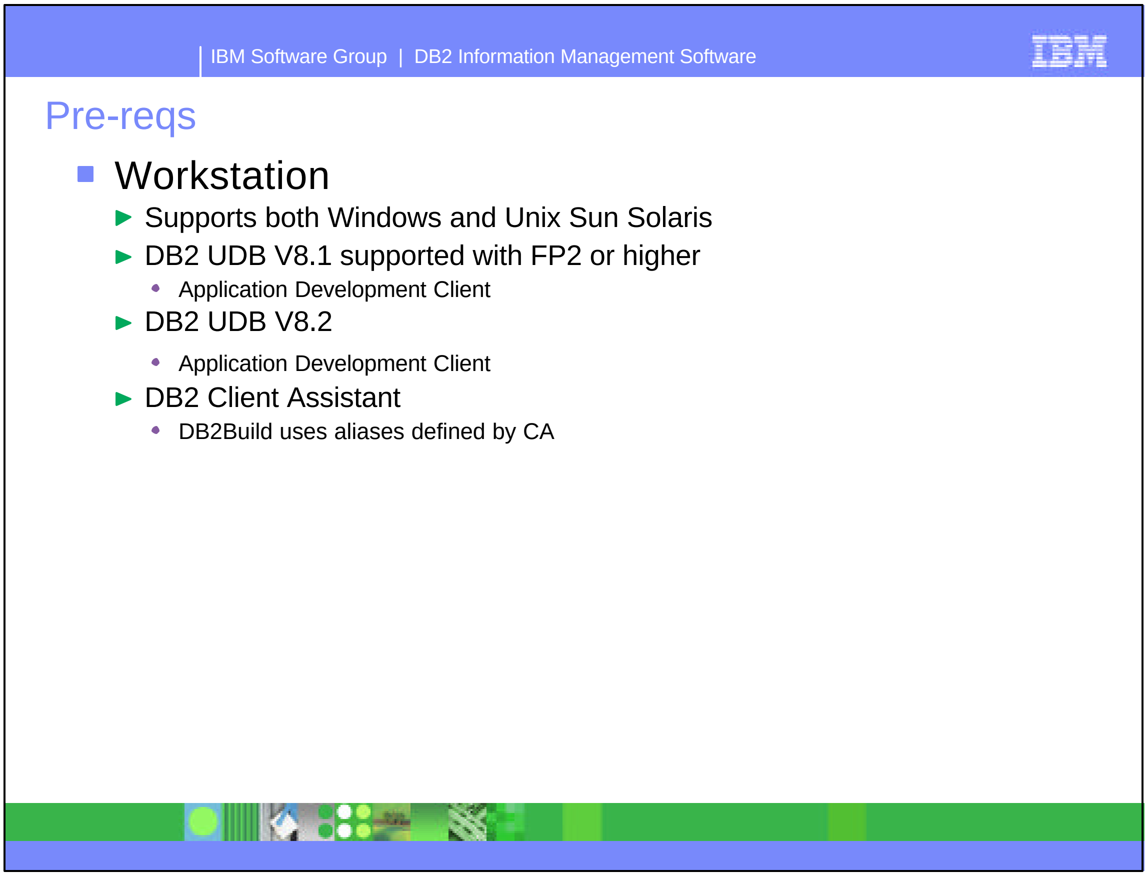
Calls DSNTPSMP

DB2PROD

PROCSQL1

PROCSQL2

- This slide provides a quick flow of how the DB2Build process works.
- 1 - First you create and test your SQL SPs. This can be done
  - Manually using an editor and user written program to test or
  - By using the DB2 UDB V8 Development Center, WSAD V5, RAD V6 or .NET. See slide title, "Step 1 - Create and Test SQL SP" for detailed steps
- 2 -Next you create the DB2Build input file, e.g. proc1.db2, with the selected SQL SPs. This can be done
  - Manually using an editor or
  - By using the export feature in DC to a file
- 3 - Run DB2Build with the input file.
  - This calls DSNTPSMP on the target DB2 server, following the steps described on later slide titled, "Step 1 - Create and Test SQL SP"

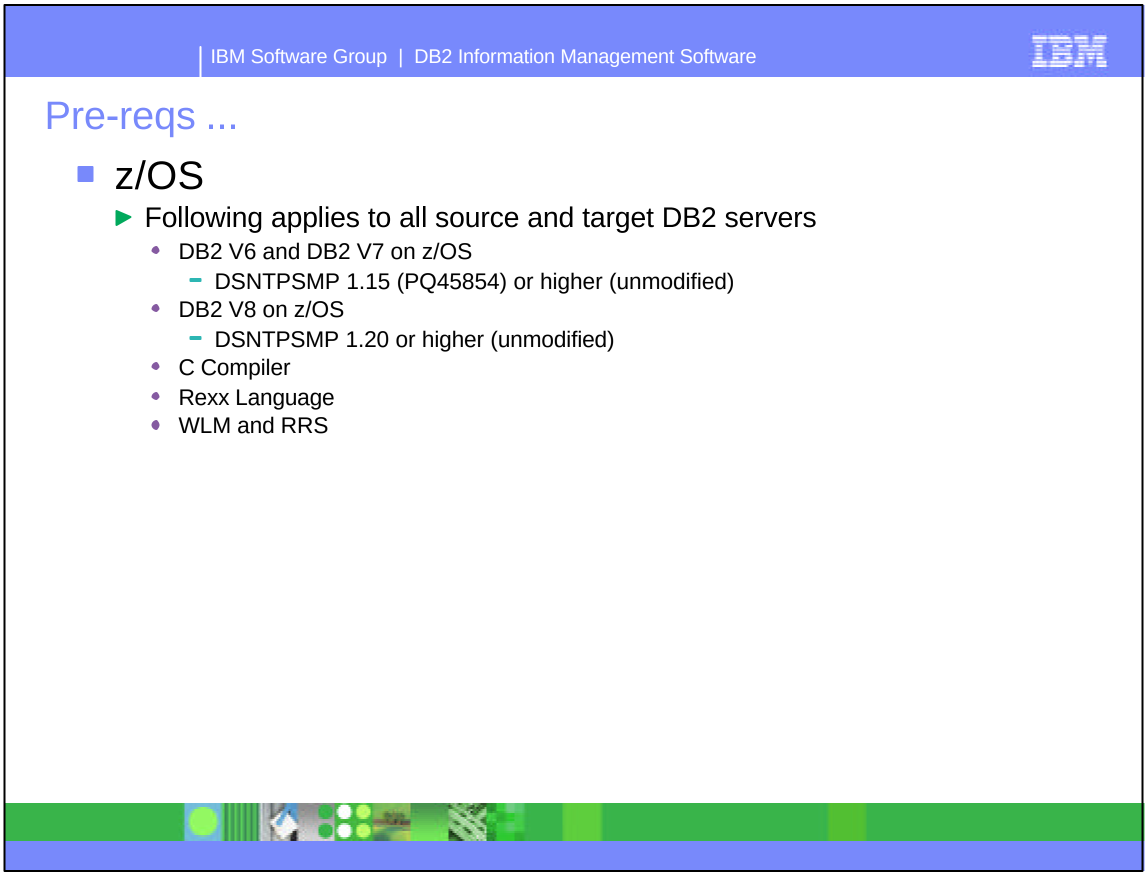


IBM Software Group | DB2 Information Management Software

Pre-reqs

- Workstation
  - Supports both Windows and Unix Sun Solaris
  - DB2 UDB V8.1 supported with FP2 or higher
    - Application Development Client
  - DB2 UDB V8.2
    - Application Development Client
  - DB2 Client Assistant
    - DB2Build uses aliases defined by CA

- This slide describes the pre-reqs needed to use DB2Build
  - Workstation
    - DB2Build operates on both a Windows and Unix client.
      - The client is supported as part of the Application Development Client of DB2 UDB V8.1with Fixpak2, in DB2 UDB V8.2



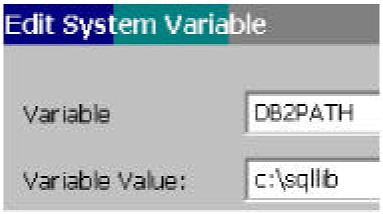
Pre-reqs ...

- z/OS
  - Following applies to all source and target DB2 servers
    - DB2 V6 and DB2 V7 on z/OS
      - DSNTPSMP 1.15 (PQ45854) or higher (unmodified)
    - DB2 V8 on z/OS
      - DSNTPSMP 1.20 or higher (unmodified)
    - C Compiler
    - Rexx Language
    - WLM and RRS

- z/OS
  - All source and target DB2 servers have the same requirements.
    - DSNTPSMP is required in an 'unmodified' state. That is, it must not have user mods included in it to be supported and operate with DB2Build.
    - DB2 V6 and V7 on z/OS servers require
      - DSNTPSMP at a 1.15 level or higher
    - DB2 V8 on z/OS servers require
      - DSNTPSMP at a 1.20 level or higher
    - C Compiler
    - Rexx Language
    - WLM and RRS

Setup - Workstation

- Verify DB2PATH environment system variable set
  - Point to ..\sqllib
- DB2Build files
  - ..sqllib\bin\db2build.bat
    - Batch file to run DB2Build for Windows
  - ..sqllib\bin\db2build
    - Batch file to run DB2Build for non-Windows version
  - ..sqllib\samples\db2build\proc1.db2
    - A sample SQL source file for creating two stored procedures using DB2Build



Edit System Variable

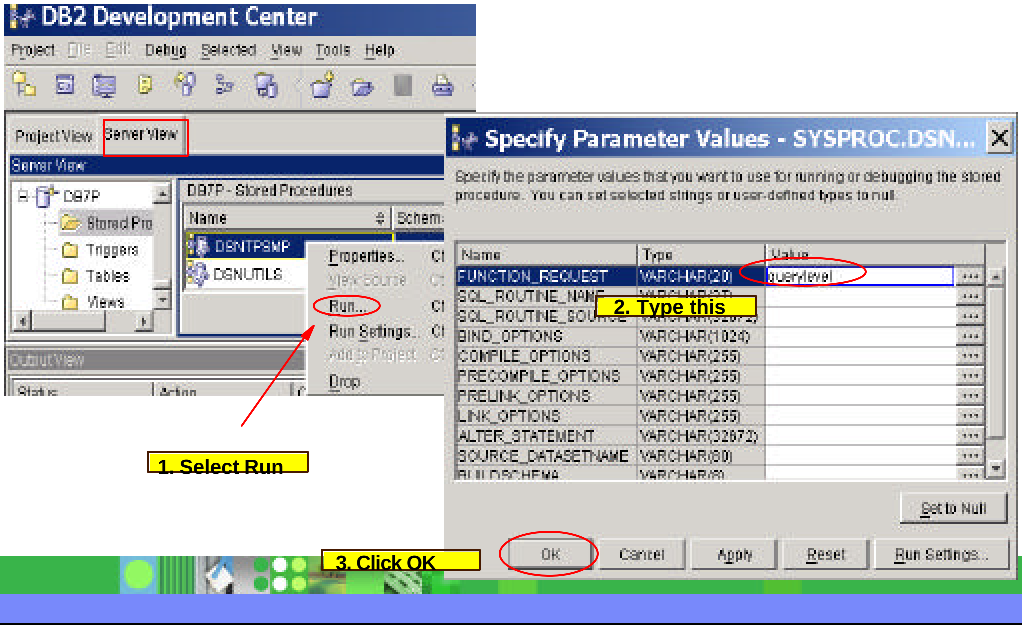
Variable: DB2PATH

Variable Value: c:\sqllib

- The workstation requires that the DB2PATH environment system variable be set up to point to ..\sqllib
- Some of the files used by DB2Build are described on this slide.

Setup - z/OS

- From DC Server View - Enter **querylevel** for FUNCTION_REQUEST click **OK**



1. Select Run

2. Type this

3. Click OK

Specify Parameter Values - SYSPROC.DSN...

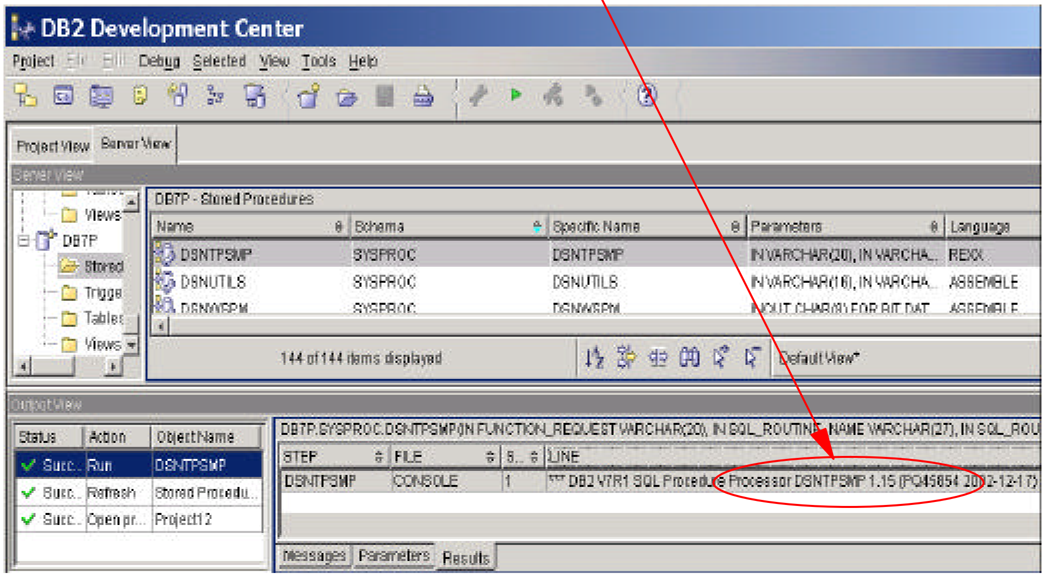
Specify the parameter values that you want to use for running or debugging the stored procedure. You can set selected strings or user-defined types to null.

Name	Type	Value
FUNCTION_REQUEST	VARCHAR(20)	querylevel
SQL_ROUTINE_NAME	[illegible]	
SQL_ROUTINE_SOURCE	[illegible]	
BIND_OPTIONS	VARCHAR(1024)	
COMPILE_OPTIONS	VARCHAR(255)	
PRECOMPILE_OPTIONS	VARCHAR(255)	
PRELINK_OPTIONS	VARCHAR(255)	
LINK_OPTIONS	VARCHAR(255)	
ALTER_STATEMENT	VARCHAR(32672)	
SOURCE_DATASETNAME	VARCHAR(80)	
BUILDSCHEMA	VARCHAR(8)	

- The z/OS setup requires DSNTPSMP at either 1.15 or higher for DB2 V6 and V7 and 1.20 or higher for DB2 V8. DSNTPSMP is the default name of this stored procedure. If this SP has been renamed for some reason, use the appropriate name for your environment in the following steps.
- You can verify the correct level is being used from DB2 Development Center.
- From Server View, select SYSPROC.DSNTPSMP, Right click and
- 1. select Run.
- 2. On the 'Specify Parameter Values' panel, enter querylevel in the FUNCTION_REQUEST Value field.
- 3. Click OK.
- DSNTPSMP has to be at a 1.15 level or above for this options to be displayed from DC. If it is not displayed, you will need to look directly at the //SYSEXEC DD pointing to where the DSNTPSMP Clist is located.
- If you are not currently operating at a 1.15 level you will not see the response to the run on the next slide. Looking directly at the comments at the beginning of the DSNTPSMP Rexx clist identifies the level being used. The Rexx Clist is located in the DD pointed to by //SYSEXEC in your WLM proc where DSNTPSMP is running. This dataset is normally located in, hlq.SDSNCLST.

Setup - z/OS ...

- DC returns DSNTPSMP level



DB7P - Stored Procedures

Name	Schema	Specific Name	Parameters	Language
DSNTPSMP	SYSPROC	DSNTPSMP	IN VARCHAR(20), IN VARCHA...	REXX
DSNUTILS	SYSPROC	DSNUTILS	IN VARCHAR(16), IN VARCHA...	ASSEMBLE
DSNWSPM	SYSPROC	DSNWSPM	INOUT CHAR(8) FOR BIT DAT...	ASSEMBLE

144 of 144 items displayed

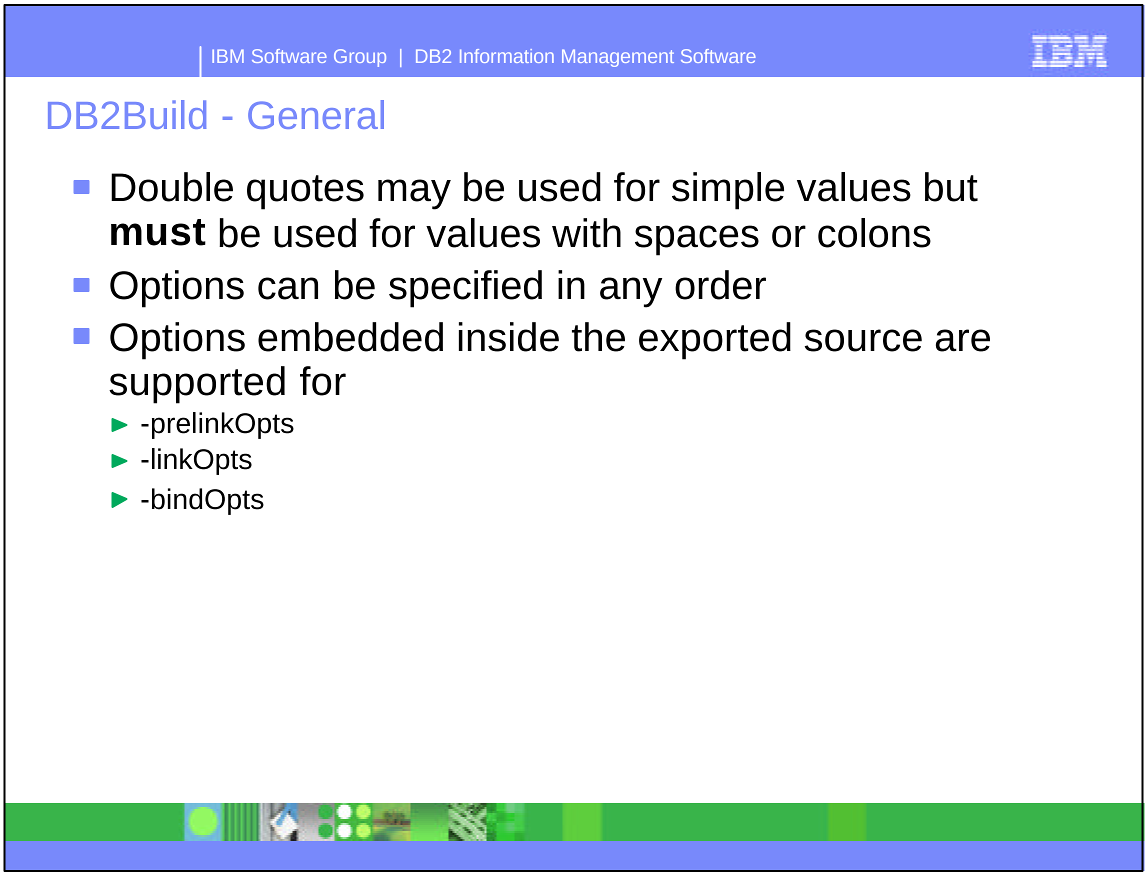
Default View*

Status	Action	Object Name
✓ Succ...	Run	DSNTPSMP
✓ Succ...	Refresh	Stored Procedu...
✓ Succ...	Open pr...	Project12

DB7P.SYSPROC.DSNTPSMP(IN FUNCTION_REQUEST VARCHAR(20), IN SQL_ROUTINE_NAME VARCHAR(27), IN SQL_ROU

STEP	FILE	S...	LINE
DSNTPSMP	CONSOLE	1	*** DB2 V7R1 SQL Procedure Processor DSNTPSMP 1.15 (PQ45854 2002-12-17)

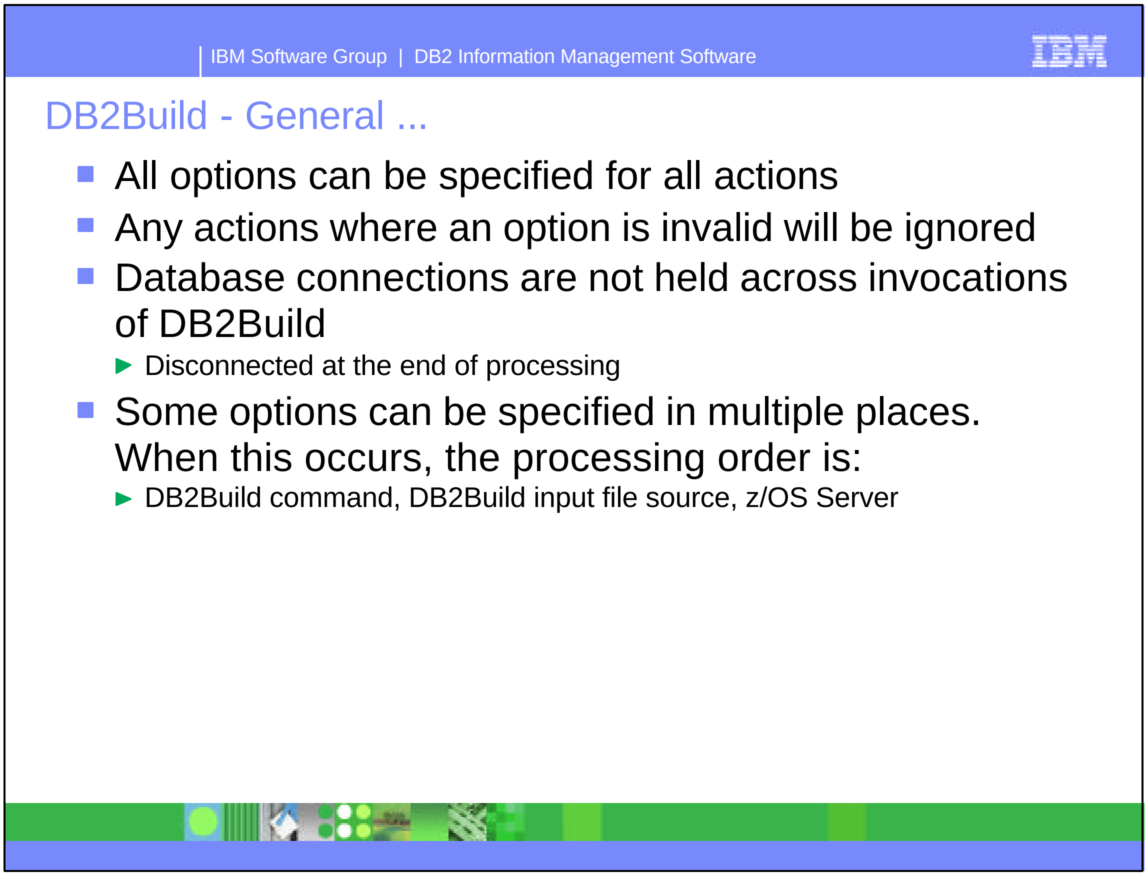
- This slide shows the output level of DSNTPSMP in the Development Center Results panel at the bottom right side of the panel.



DB2Build - General

- Double quotes may be used for simple values but **must** be used for values with spaces or colons
- Options can be specified in any order
- Options embedded inside the exported source are supported for
  - -prelinkOpts
  - -linkOpts
  - -bindOpts

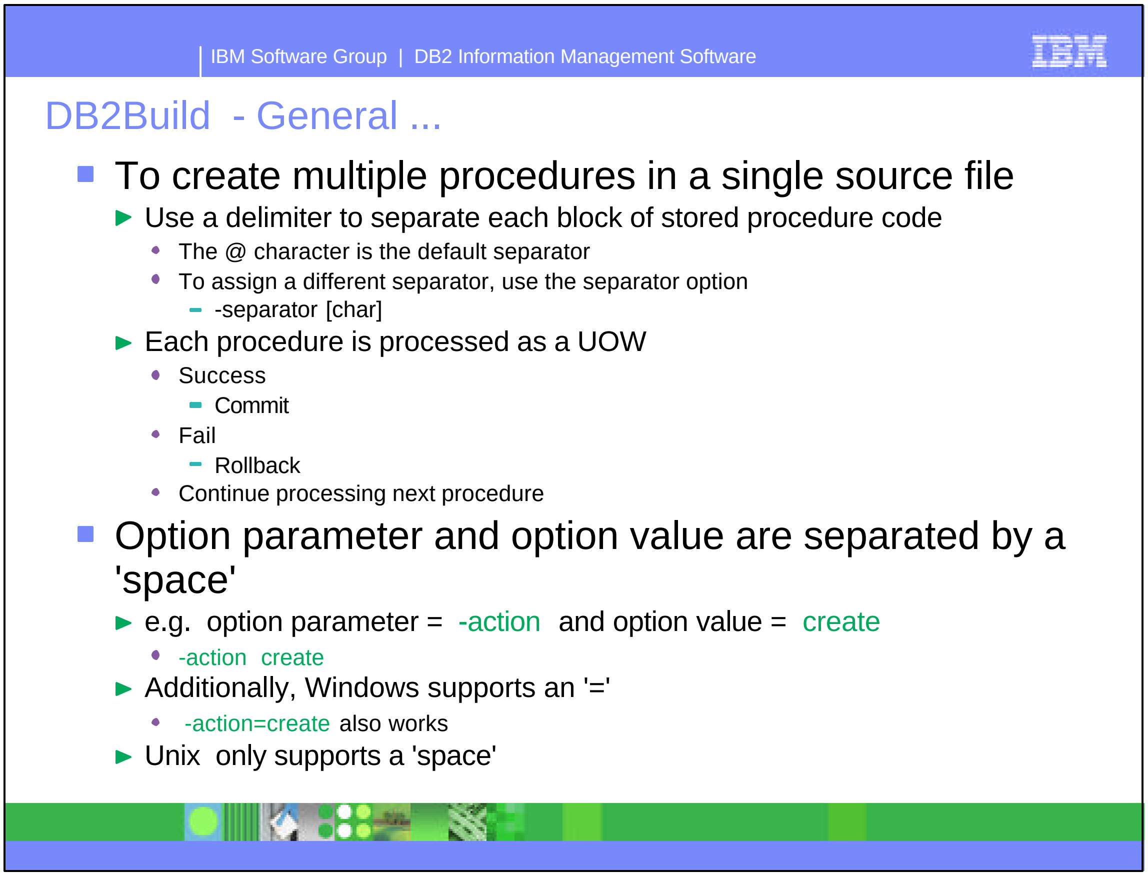
- This slide describes general information that applies to all the DB2Build options listed on the following slides.
- Double quotes " must be used for values that include spaces or colons, otherwise they are not required.
- Options can be specified in any order.
- The -prelinkOpts, -linkOpts and -bindOpts can be included in the exported source when multiple SQL SPs are included. That is PROCSQL1 and PROCSQL2 are both included in a single file. When multiple SQL SPs are included in a single file, and they all use the same -prelinkOpts, or -linkOpts, or -bindOpts, those options can be included on the main DB2BUILD -action command line.



DB2Build - General ...

- All options can be specified for all actions
- Any actions where an option is invalid will be ignored
- Database connections are not held across invocations of DB2Build
  - Disconnected at the end of processing
- Some options can be specified in multiple places. When this occurs, the processing order is:
  - DB2Build command, DB2Build input file source, z/OS Server

- All options can be specified for any action. Those that are invalid for the action, will be ignored. A message will be written to the logfile, indicating any option that was ignored because it was invalid.
- A database connection is made when DB2Build starts processing. When the processing is completed, the database is disconnected.

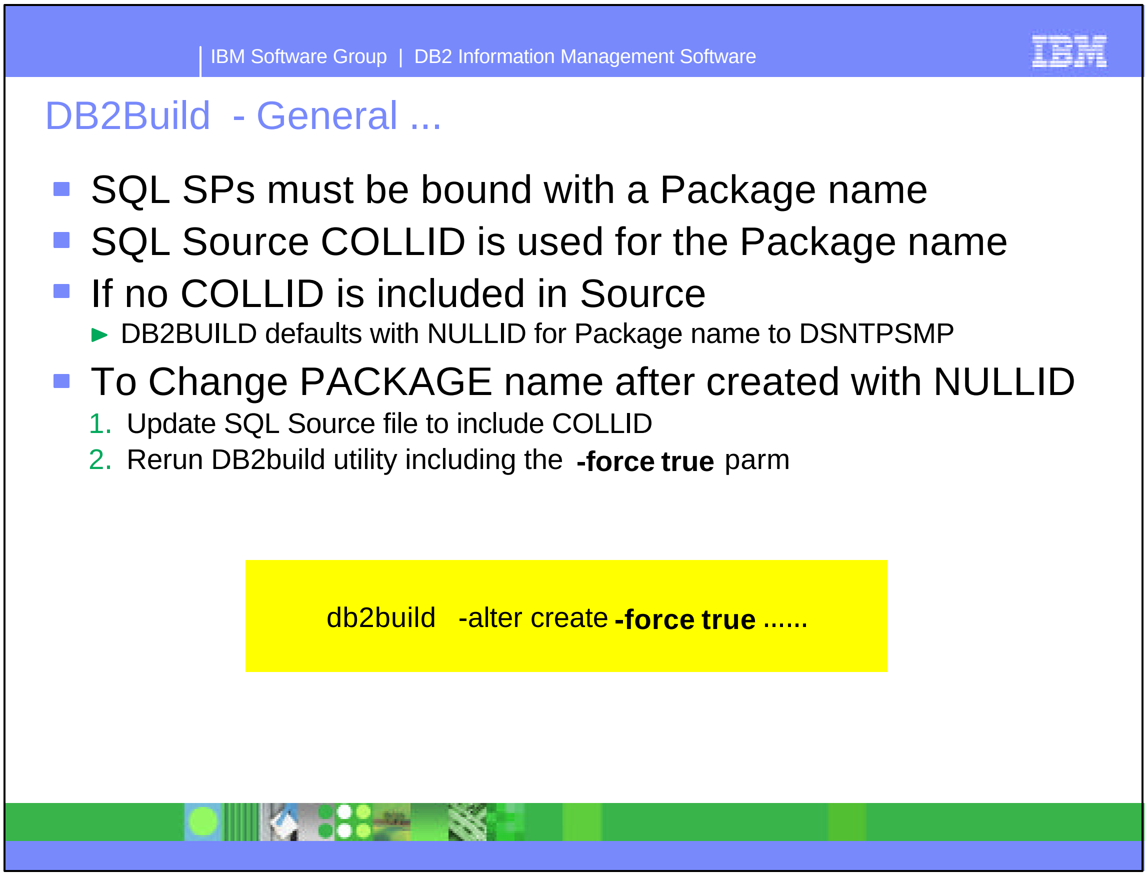


IBM Software Group | DB2 Information Management Software

DB2Build - General ...

- To create multiple procedures in a single source file
  - Use a delimiter to separate each block of stored procedure code
    - The @ character is the default separator
    - To assign a different separator, use the separator option
      - -separator [char]
  - Each procedure is processed as a UOW
    - Success
      - Commit
    - Fail
      - Rollback
    - Continue processing next procedure
- Option parameter and option value are separated by a 'space'
  - e.g. option parameter = -action and option value = create
    - -action create
  - Additionally, Windows supports an '='
    - -action=create also works
  - Unix only supports a 'space'

- When multiple procedures are included in one or more files passed to the DB2Build utility, each procedure is processed as an individual unit of work.
- The default separator character is the @ character. To change to a different character include the -separator option on the DB2Build command line.
  - e.g. -separator %
  - Separator can exist or 'not' exist at the end of a file.
- When multiple procedures exist in a single file, or multiple files are included in one DB2Build -action execution, each procedure is processed as a single UOW. It either succeeds, and the operation is committed, or fails, and the operation is rolled back. Processing continues with the next procedure until all are completed.
- The option parameter is prefixed by a -
- The option parameter and the option value are separated by a SPACE for both Windows and Unix platforms. Additionally, Windows supports an = sign as a separator.

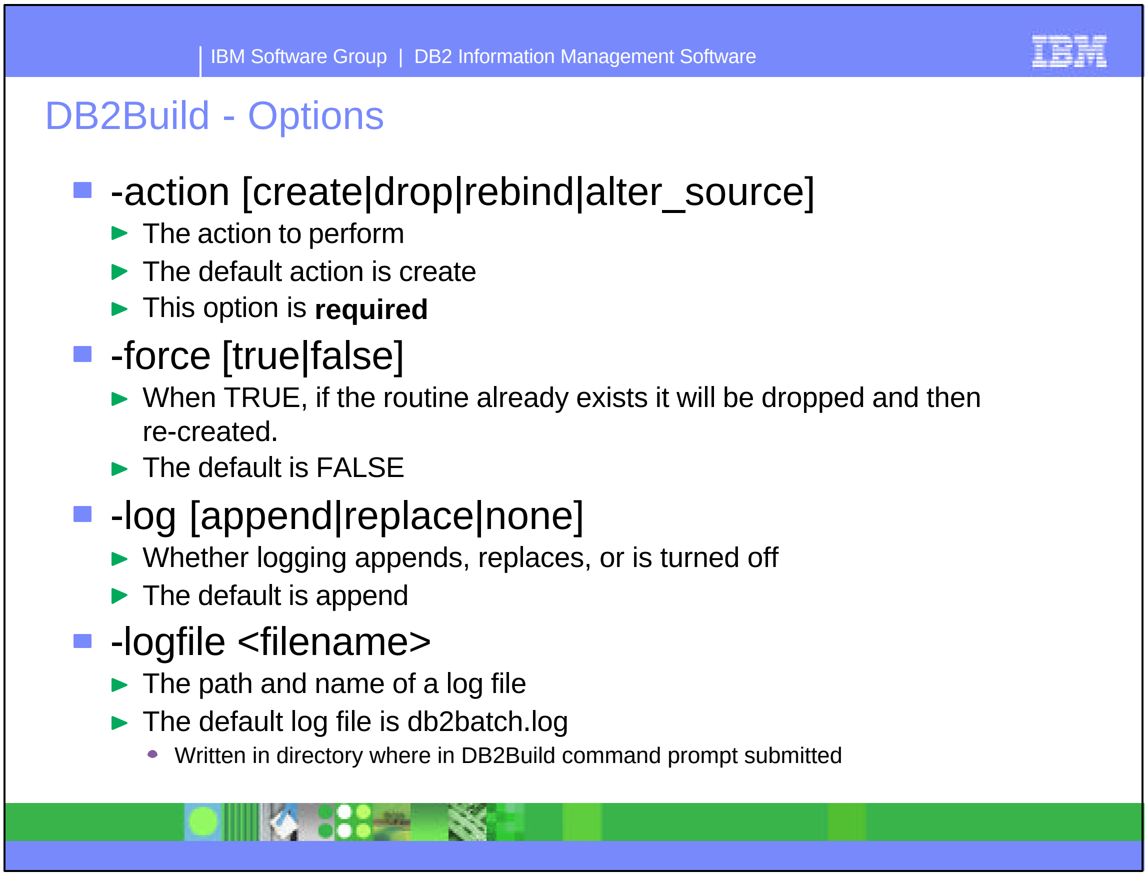


DB2Build - General ...

- SQL SPs must be bound with a Package name
- SQL Source COLLID is used for the Package name
- If no COLLID is included in Source
  - DB2BUILD defaults with NULLID for Package name to DSNTPSMP
- To Change PACKAGE name after created with NULLID
  1. Update SQL Source file to include COLLID
  2. Rerun DB2build utility including the **-force true** parm

```
db2build   -alter create -force true ......
```

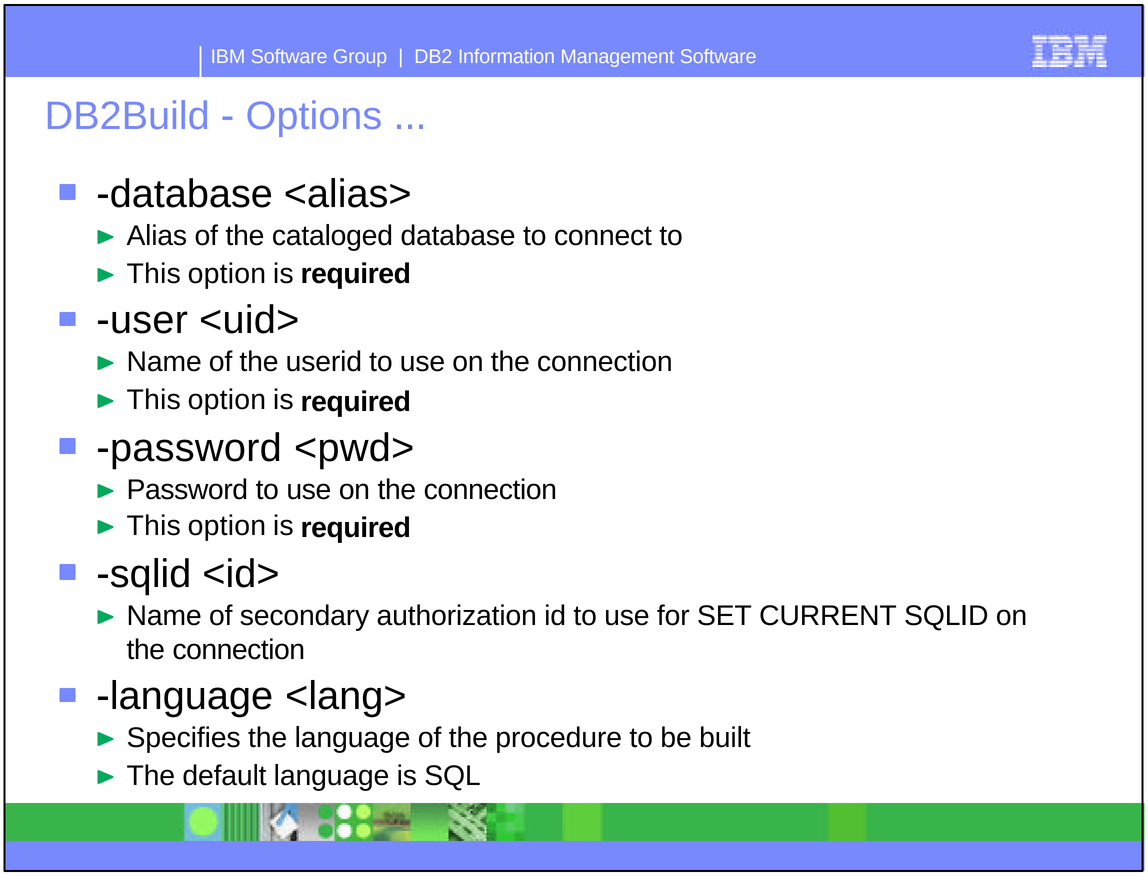
- SQL SPs require a package name when the SP is bound which is performed by DSNTPSMP. The COLLID in the Procedure definition is used for the name. If a COLLID is not present in the Procedure definition, a default of NULLID is passed to DSNTPSMP by the DB2build utility.
- To change the package name on the target system from NULLID to another value
  - 1. Update the source file with the NULLID for a package name
  - 2. Add a COLLID value to the Procedure source
  - 3. Rerun DB2build with -action create -force true



DB2Build - Options

- -action [create|drop|rebind|alter_source]
  - The action to perform
  - The default action is create
  - This option is **required**
- -force [true|false]
  - When TRUE, if the routine already exists it will be dropped and then re-created.
  - The default is FALSE
- -log [append|replace|none]
  - Whether logging appends, replaces, or is turned off
  - The default is append
- -logfile <filename>
  - The path and name of a log file
  - The default log file is db2batch.log
    - Written in directory where in DB2Build command prompt submitted

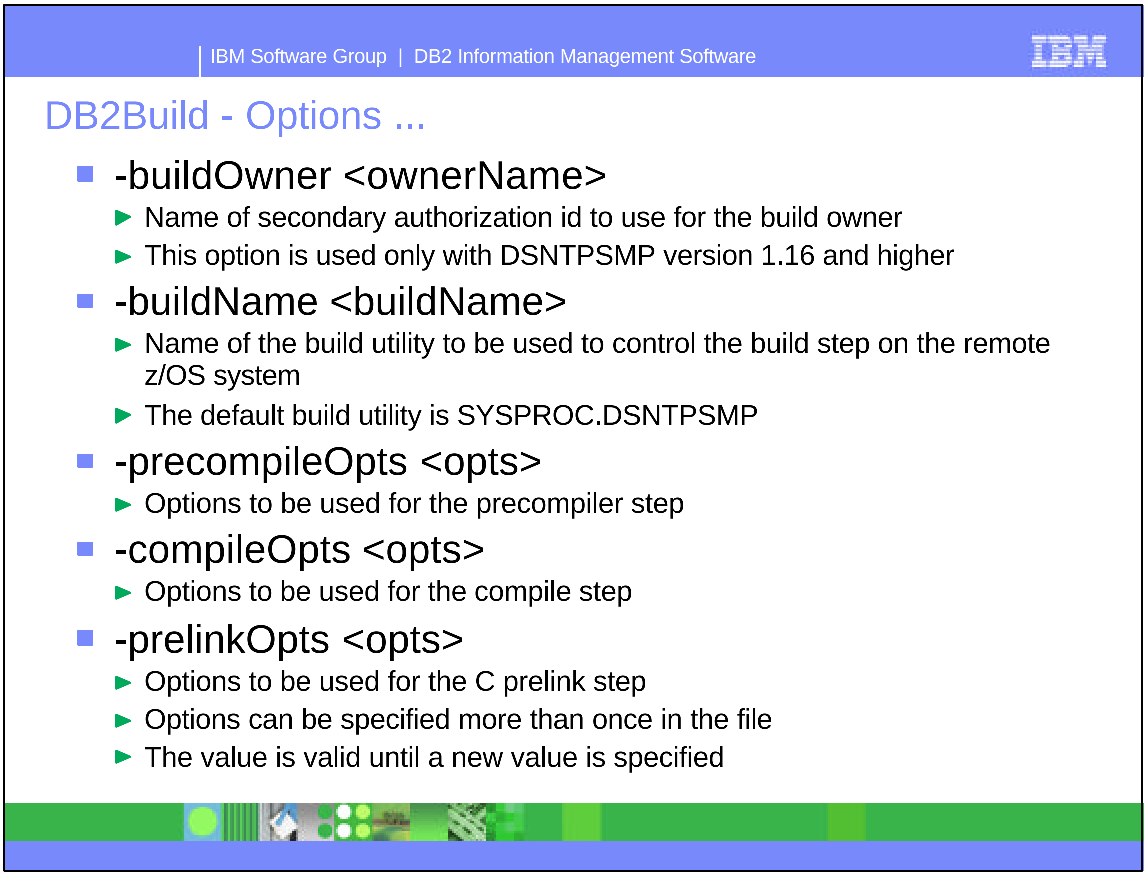
- The next few slides describe the DB2Build options that are supported. These are self explanatory.
- These are all displayed by entering **DB2Build** from a command prompt.
- Required options are noted.
- Defaults are also noted.



DB2Build - Options ...

- -database <alias>
  - Alias of the cataloged database to connect to
  - This option is **required**
- -user <uid>
  - Name of the userid to use on the connection
  - This option is **required**
- -password <pwd>
  - Password to use on the connection
  - This option is **required**
- -sqlid <id>
  - Name of secondary authorization id to use for SET CURRENT SQLID on the connection
- -language <lang>
  - Specifies the language of the procedure to be built
  - The default language is SQL

- This slide describes more of the DB2Build options that are supported. These are self explanatory.

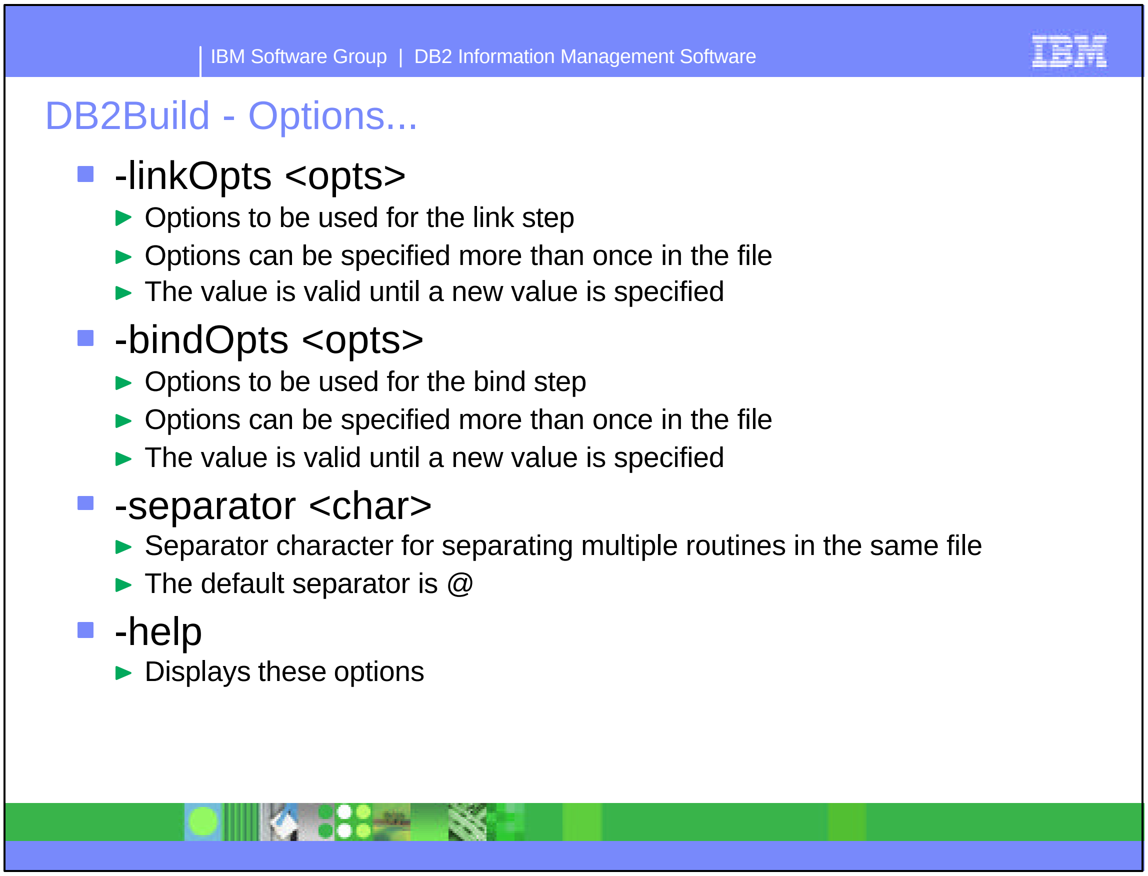


IBM Software Group | DB2 Information Management Software

DB2Build - Options ...

- -buildOwner <ownerName>
  - Name of secondary authorization id to use for the build owner
  - This option is used only with DSNTPSMP version 1.16 and higher
- -buildName <buildName>
  - Name of the build utility to be used to control the build step on the remote z/OS system
  - The default build utility is SYSPROC.DSNTPSMP
- -precompileOpts <opts>
  - Options to be used for the precompiler step
- -compileOpts <opts>
  - Options to be used for the compile step
- -prelinkOpts <opts>
  - Options to be used for the C prelink step
  - Options can be specified more than once in the file
  - The value is valid until a new value is specified

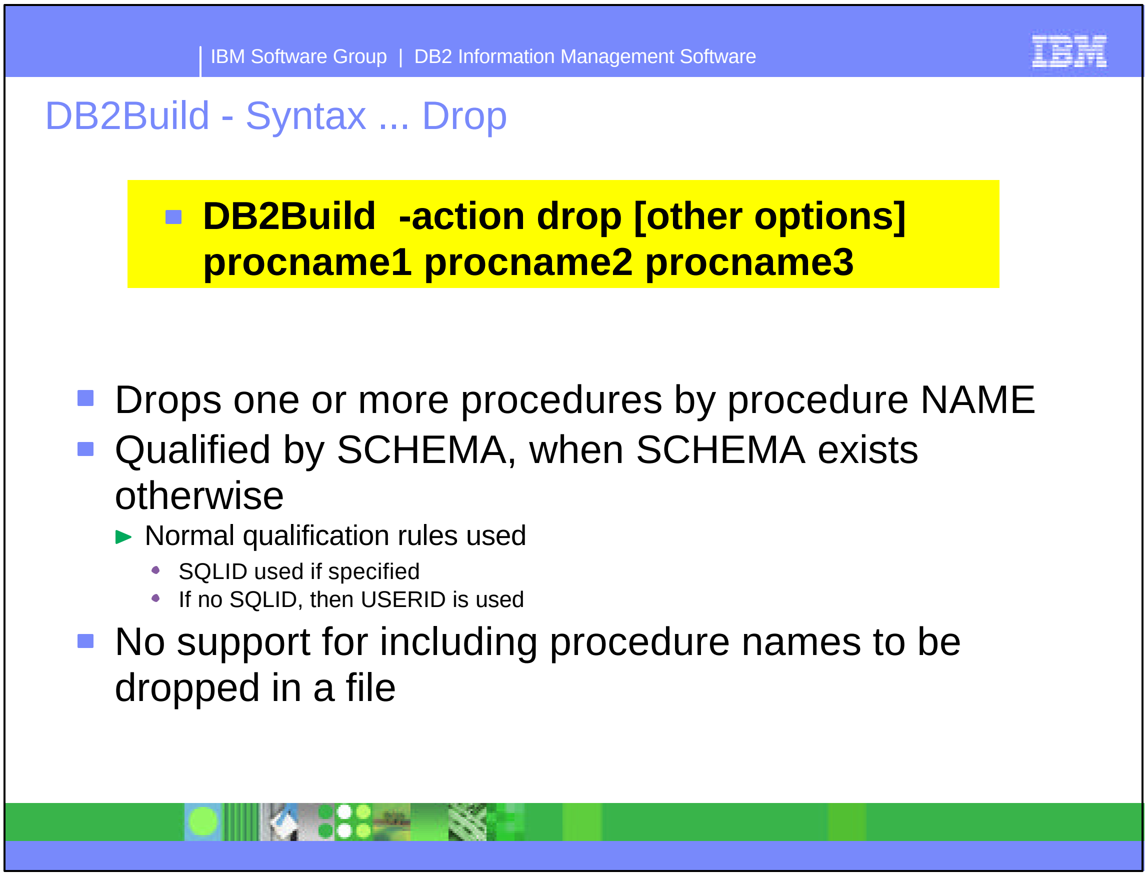
- This slide describes more of the DB2Build options that are supported. These are self explanatory.



DB2Build - Options...

- -linkOpts <opts>
  - Options to be used for the link step
  - Options can be specified more than once in the file
  - The value is valid until a new value is specified
- -bindOpts <opts>
  - Options to be used for the bind step
  - Options can be specified more than once in the file
  - The value is valid until a new value is specified
- -separator <char>
  - Separator character for separating multiple routines in the same file
  - The default separator is @
- -help
  - Displays these options

- This slide describes more of the DB2Build options that are supported. These are self explanatory.

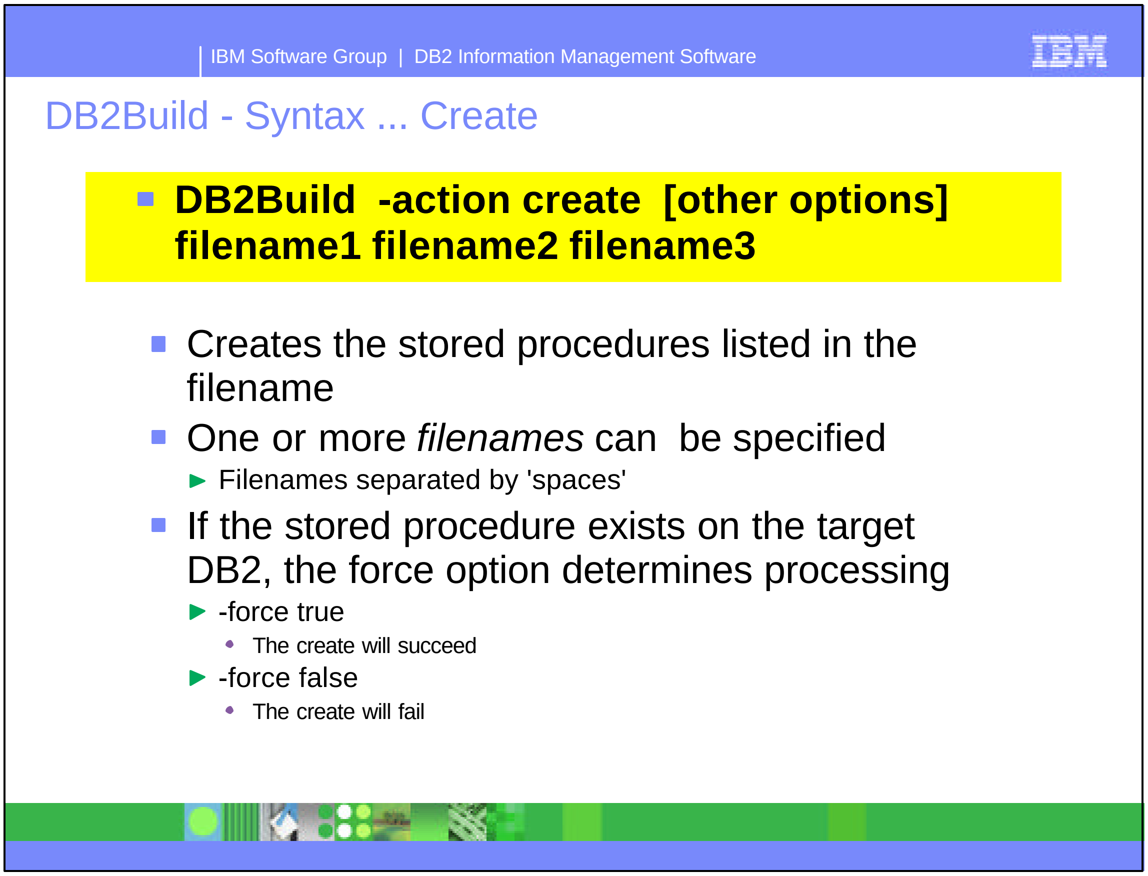


DB2Build - Syntax ... Drop

- **DB2Build -action drop [other options] procname1 procname2 procname3**

- Drops one or more procedures by procedure NAME
- Qualified by SCHEMA, when SCHEMA exists otherwise
  - Normal qualification rules used
    - SQLID used if specified
    - If no SQLID, then USERID is used
- No support for including procedure names to be dropped in a file

- Drop one or more SQL SPs from the DB2Build command.
  - One or more PROCEDURE NAMES, separated by spaces, can be included in the DB2Build command on the command prompt.

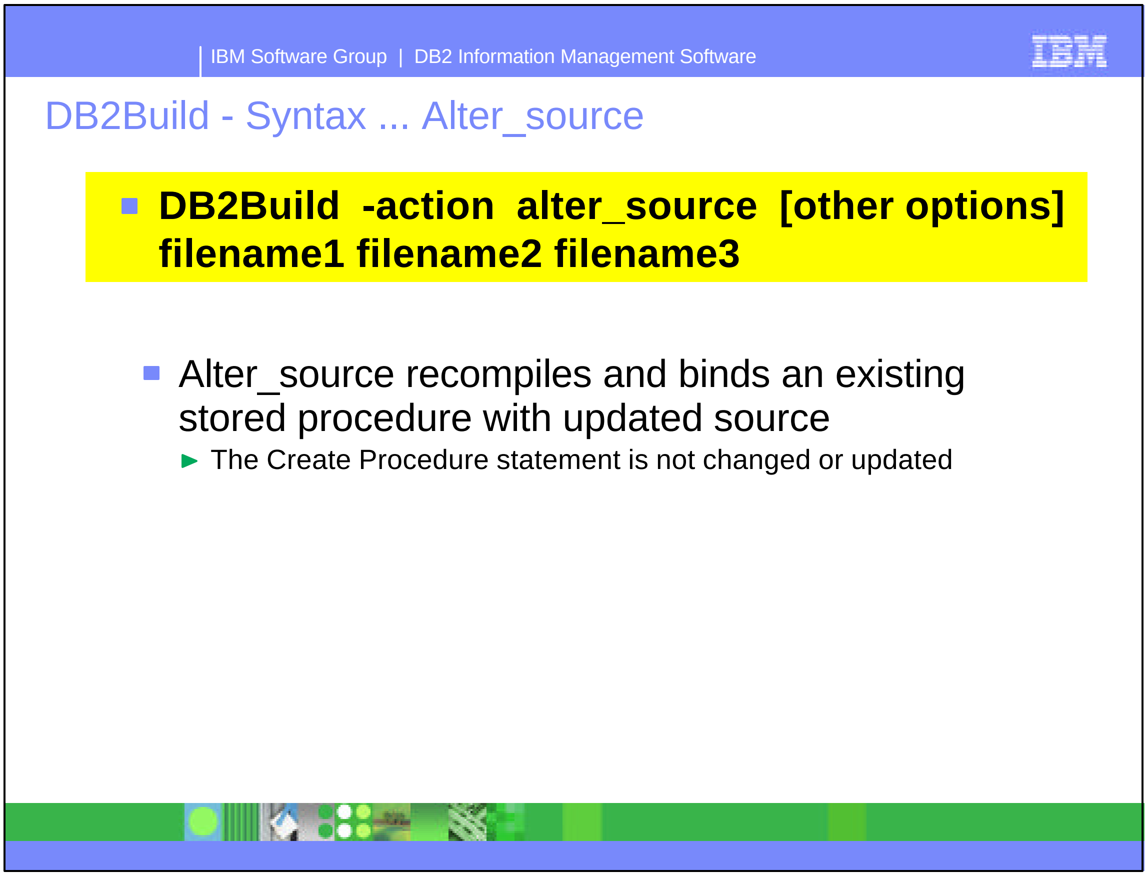


DB2Build - Syntax ... Create

- **DB2Build -action create [other options] filename1 filename2 filename3**

- Creates the stored procedures listed in the filename
- One or more *filenames* can be specified
  - Filenames separated by 'spaces'
- If the stored procedure exists on the target DB2, the force option determines processing
  - -force true
    - The create will succeed
  - -force false
    - The create will fail

- This slide describes the syntax for the Create action.
- One or more filenames can be included on the Command prompt, separated by 'spaces'. One or more stored procedures can be included in multiple filenames that may be used in the command.
- If the stored procedure already exists, using the force option, determines the processing action. If the option, -force true is included the create will succeed. If the -force false (the default) option is specified, the create will fail.

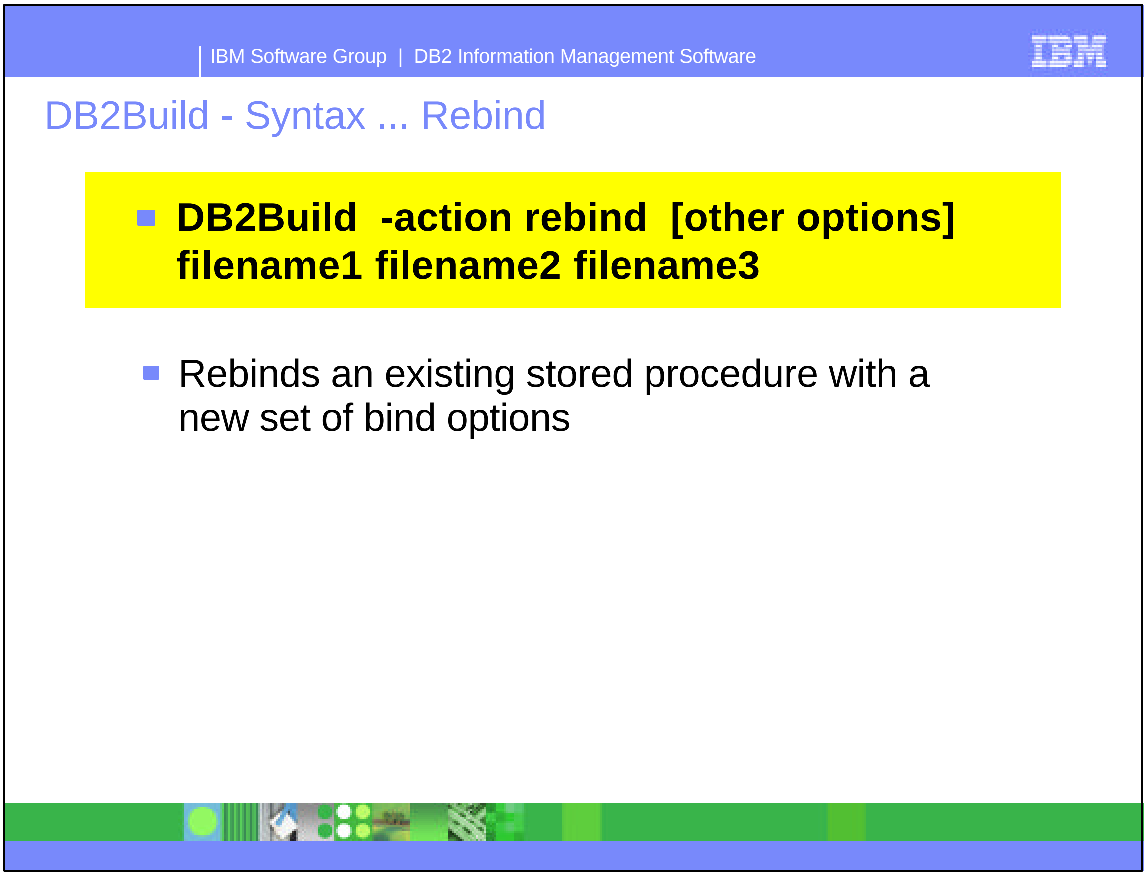


DB2Build - Syntax ... Alter_source

- **DB2Build -action alter_source [other options] filename1 filename2 filename3**

- Alter_source recompiles and binds an existing stored procedure with updated source
  - The Create Procedure statement is not changed or updated

- This slide describes the syntax for the Alter_source action.
- One or more filenames can be included on the Command prompt, separated by 'spaces'. One or more stored procedures can be included in multiple filenames that may be used in the command.
- Only the source of the SP is updated with this command, and the source is rebuilt. No changes are made to the Create Procedure definition.

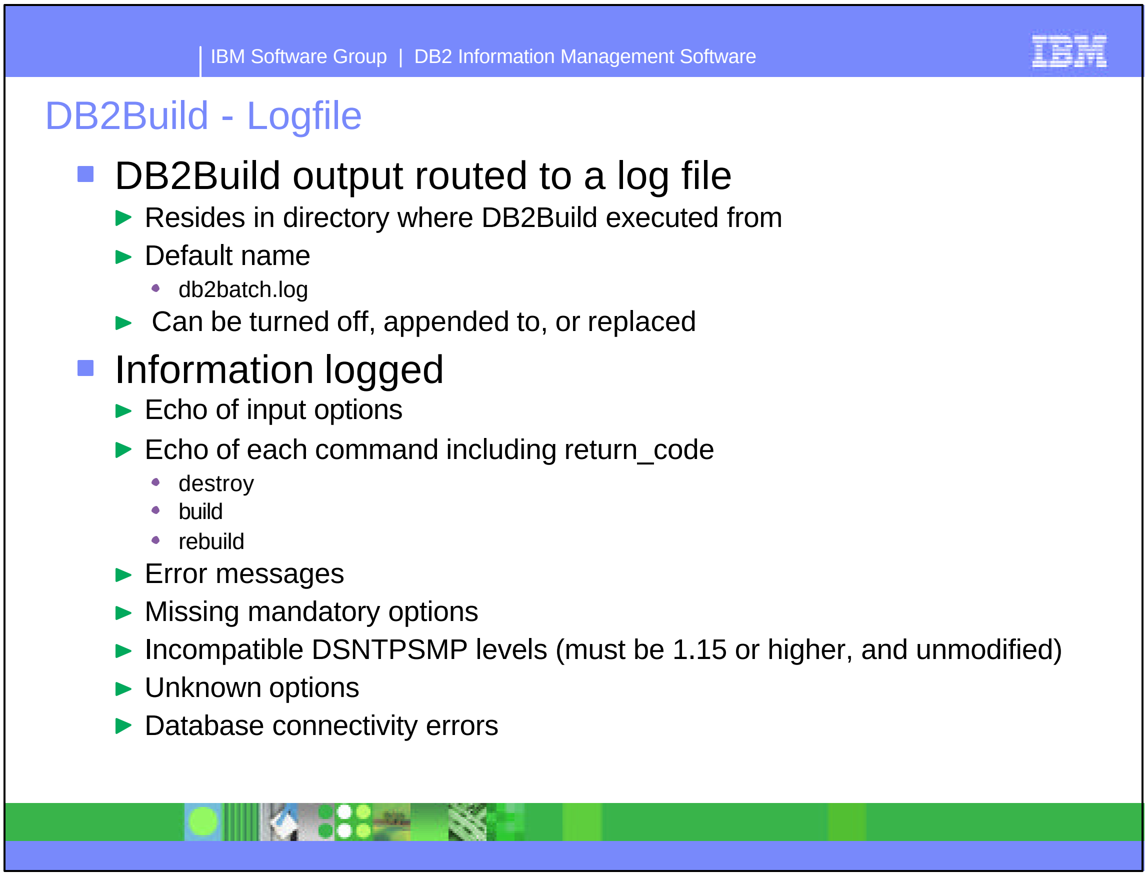


DB2Build - Syntax ... Rebind

- **DB2Build -action rebind [other options] filename1 filename2 filename3**

- Rebinds an existing stored procedure with a new set of bind options

- This slide describes the syntax for the Rebind action.
- One or more filenames can be included on the Command prompt, separated by 'spaces'. One or more stored procedures can be included in multiple filenames that may be used in the command.
- This action rebinds an existing stored procedure with a new set of bind options.



DB2Build - Logfile

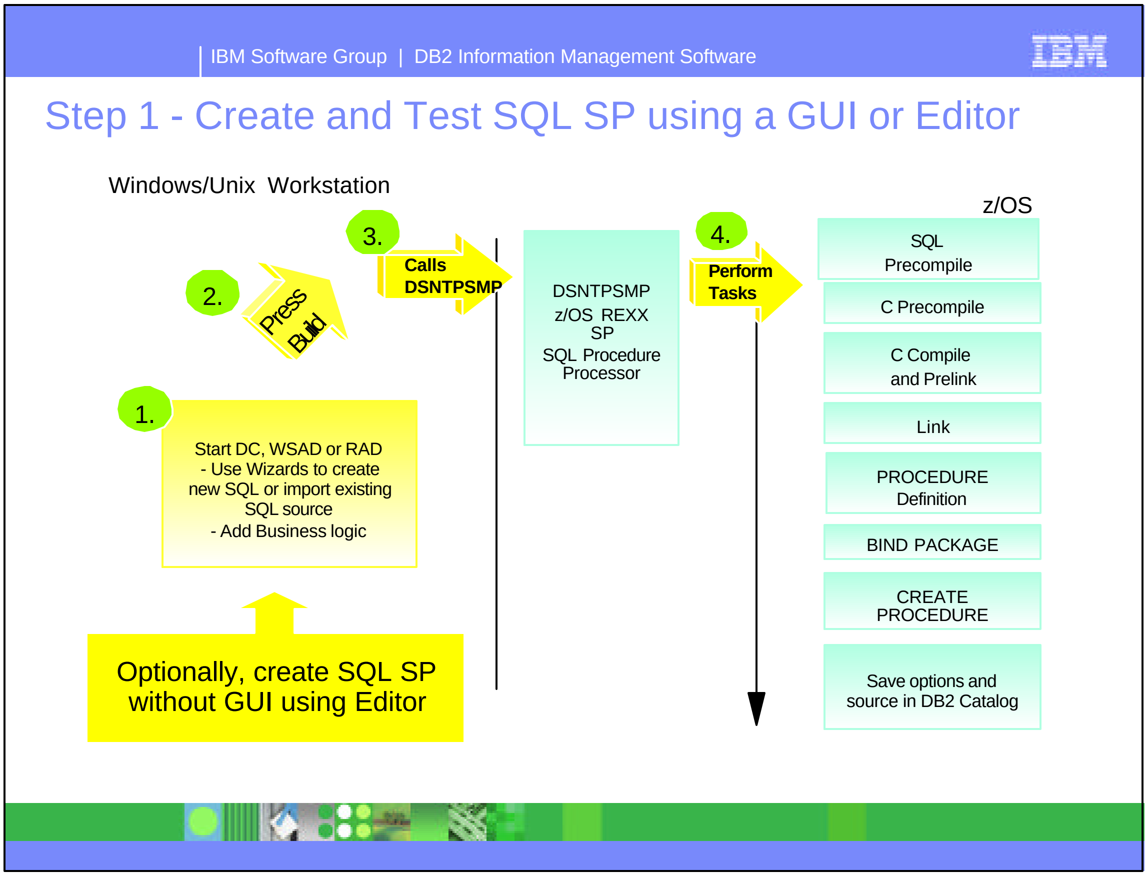
- DB2Build output routed to a log file
  - Resides in directory where DB2Build executed from
  - Default name
    - db2batch.log
  - Can be turned off, appended to, or replaced
- Information logged
  - Echo of input options
  - Echo of each command including return_code
    - destroy
    - build
    - rebuild
  - Error messages
  - Missing mandatory options
  - Incompatible DSNTPSMP levels (must be 1.15 or higher, and unmodified)
  - Unknown options
  - Database connectivity errors

- A log file is created when any DB2Build command is issued. The log file resides in the directory where the DB2Build command is positioned on when the command is submitted. The default name is db2batch.log, which can be turned off, appended to, or replaced. By default, the log is appended to.
- The name of the log file can be changed by including the -logfile option on the command prompt with the DB2Build command.
- Information that is logged is listed on the slide.

Before Migrating SQL SPs between DB2 on z/OS servers

...

Must First Create SQL SP



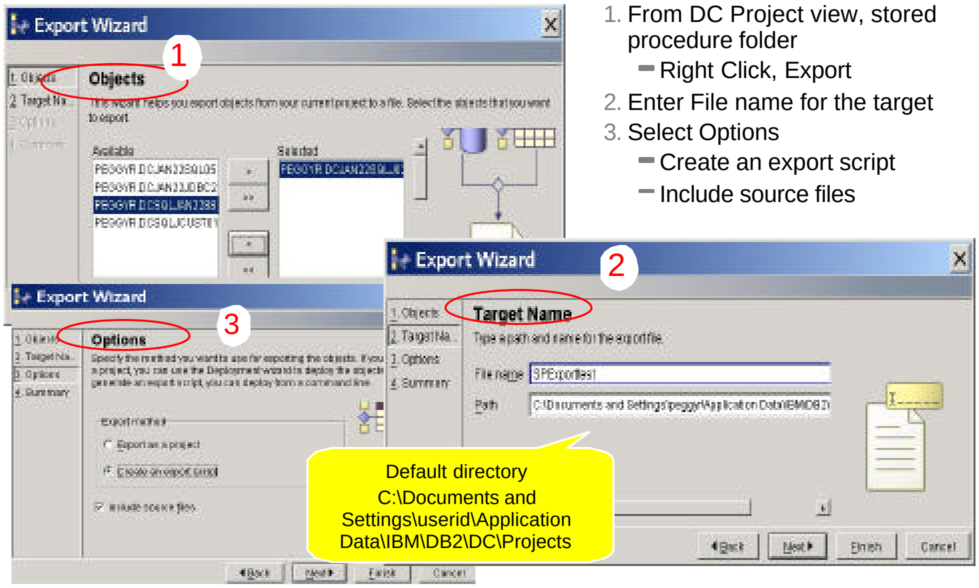
IBM Software Group | DB2 Information Management Software

Step 1 - Create and Test SQL SP using a GUI or Editor

- Create the SQL SP using an editor and test with a user written calling program.
- Alternatively, use one of the GUI tools, DB2 Development Center (DC) in DB2 UDB V8, WSAD V5 or RAD V6 to create and test the SQL SP.
-
- The GUI tool steps (1-3) on the left, occur on the workstation.
  - 1. Start DC using wizards to create a new SQL SP, import an existing SQL Source file. Add any required business logic.
  - 2. Press the Build button to start the build process
  - 3. The DB2 on z/OS DSNTPSMP stored procedure is called.
  - 4. DSNTPSMP performs the steps listed on the right to compile and install the SQL SP on the DB2 on z/OS server.

Step 2 - Create DB2Build Input File

- Using DC - Export SPs to a file



Objects: This wizard helps you export objects from your current project to a file. Select the objects that you want to export.

Target Name: Type a path and name for the export file.

File name: SPExportTest1

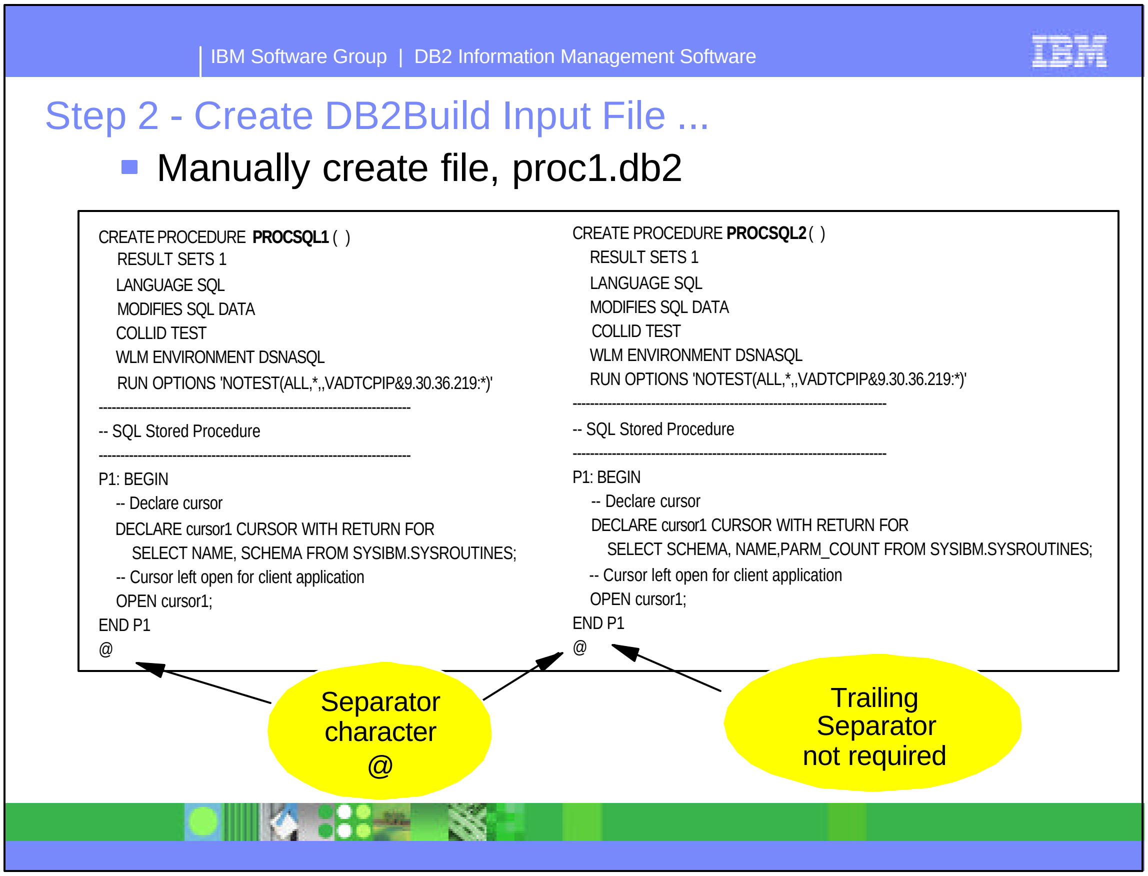
Path: C:\Documents and Settings\peggyr\Application Data\IBM\DB2\

Options: Export method: ○ Export as a project ● Create an export script ☑ Include source files

Default directory
C:\Documents and Settings\userid\Application Data\IBM\DB2\DC\Projects

1. From DC Project view, stored procedure folder
   - Right Click, Export
2. Enter File name for the target
3. Select Options
   - Create an export script
   - Include source files

- This slide shows how Development Center (DC) can be used to export one or more SQL SPs to a file.
- Positioned in Project view, on a DB2 on z/OS server connection, select the stored procedure folder and 'right click, Export' to launch the export wizard
- 4 panels make up the Export wizard.
  - The first panel, is the Object panel which displays the Available SQL SPs in the left window. Select any SP to be exported, click the '>' in the middle which moves this SP to the Selected SQL SPs in the right window.
  - The second panel is the Target name panel. Enter a file name of the export file, and optionally change the default directory where the file will reside.
  - The third panel is the Options panel. Select 'Create an export script' and check the box to 'include source files'.
  - Click Finish.



IBM Software Group | DB2 Information Management Software

Step 2 - Create DB2Build Input File ...

- Manually create file, proc1.db2

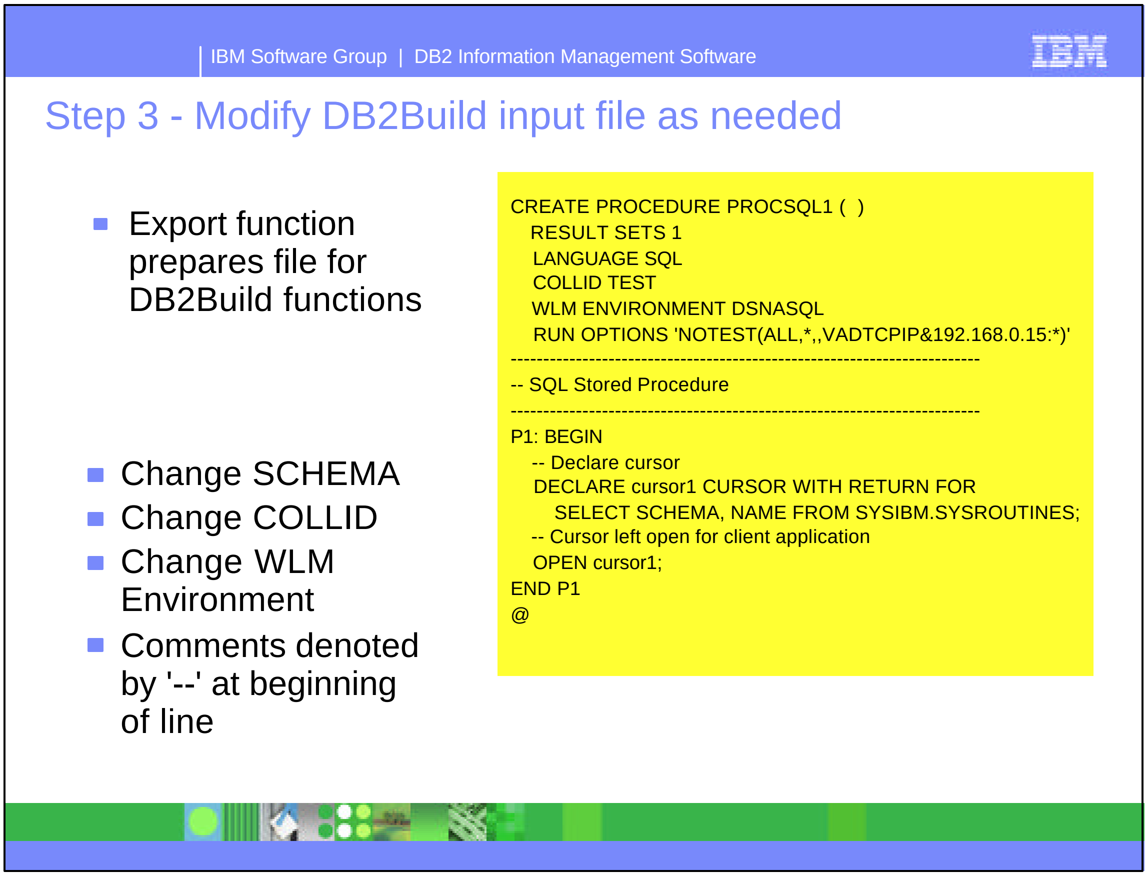
```
CREATE PROCEDURE PROCSQL1 ( )
   RESULT SETS 1
   LANGUAGE SQL
   MODIFIES SQL DATA
   COLLID TEST
   WLM ENVIRONMENT DSNASQL
   RUN OPTIONS 'NOTEST(ALL,*,,VADTCPIP&9.30.36.219:*)'
------------------------------------------------------------------------
-- SQL Stored Procedure
------------------------------------------------------------------------
P1: BEGIN
   -- Declare cursor
   DECLARE cursor1 CURSOR WITH RETURN FOR
      SELECT NAME, SCHEMA FROM SYSIBM.SYSROUTINES;
   -- Cursor left open for client application
   OPEN cursor1;
END P1
@
```

```
CREATE PROCEDURE PROCSQL2 ( )
   RESULT SETS 1
   LANGUAGE SQL
   MODIFIES SQL DATA
   COLLID TEST
   WLM ENVIRONMENT DSNASQL
   RUN OPTIONS 'NOTEST(ALL,*,,VADTCPIP&9.30.36.219:*)'
------------------------------------------------------------------------
-- SQL Stored Procedure
------------------------------------------------------------------------
P1: BEGIN
   -- Declare cursor
   DECLARE cursor1 CURSOR WITH RETURN FOR
      SELECT SCHEMA, NAME,PARM_COUNT FROM SYSIBM.SYSROUTINES;
   -- Cursor left open for client application
   OPEN cursor1;
END P1
@
```

Separator character @

Trailing Separator not required

- Optionally, create the DB2Build input file manually. This slide shows an example of two SQL SPs included in the file, proc1.db2. The two SPs are separated by the '@'.
- A final separator can be included, but is not required. DB2Build operates correctly with or without specifying the separator.



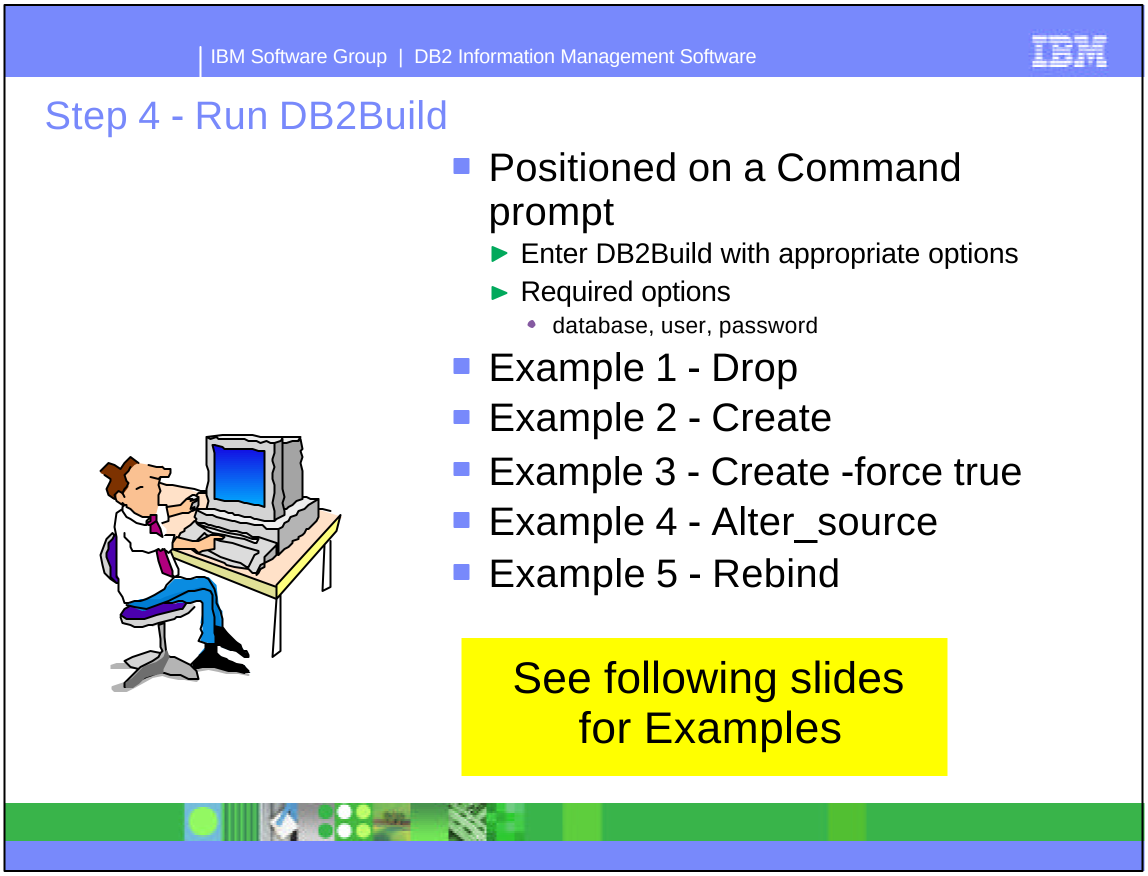
Step 3 - Modify DB2Build input file as needed

- Export function prepares file for DB2Build functions

- Change SCHEMA
- Change COLLID
- Change WLM Environment
- Comments denoted by '--' at beginning of line

```
CREATE PROCEDURE PROCSQL1 ( )
   RESULT SETS 1
    LANGUAGE SQL
    COLLID TEST
    WLM ENVIRONMENT DSNASQL
    RUN OPTIONS 'NOTEST(ALL,*,,VADTCPIP&192.168.0.15:*)'
------------------------------------------------------------------------
-- SQL Stored Procedure
------------------------------------------------------------------------
P1: BEGIN
   -- Declare cursor
    DECLARE cursor1 CURSOR WITH RETURN FOR
       SELECT SCHEMA, NAME FROM SYSIBM.SYSROUTINES;
   -- Cursor left open for client application
    OPEN cursor1;
END P1
@
```

- Prior to migrating the SQL SPs that have been exported, make any changes to this file if needed. Changes for the Procedure parms for SCHEMA, COLLID, and WLM Environment name might be desired.
- Include any comments using two dashes '--' at the beginning of any line.



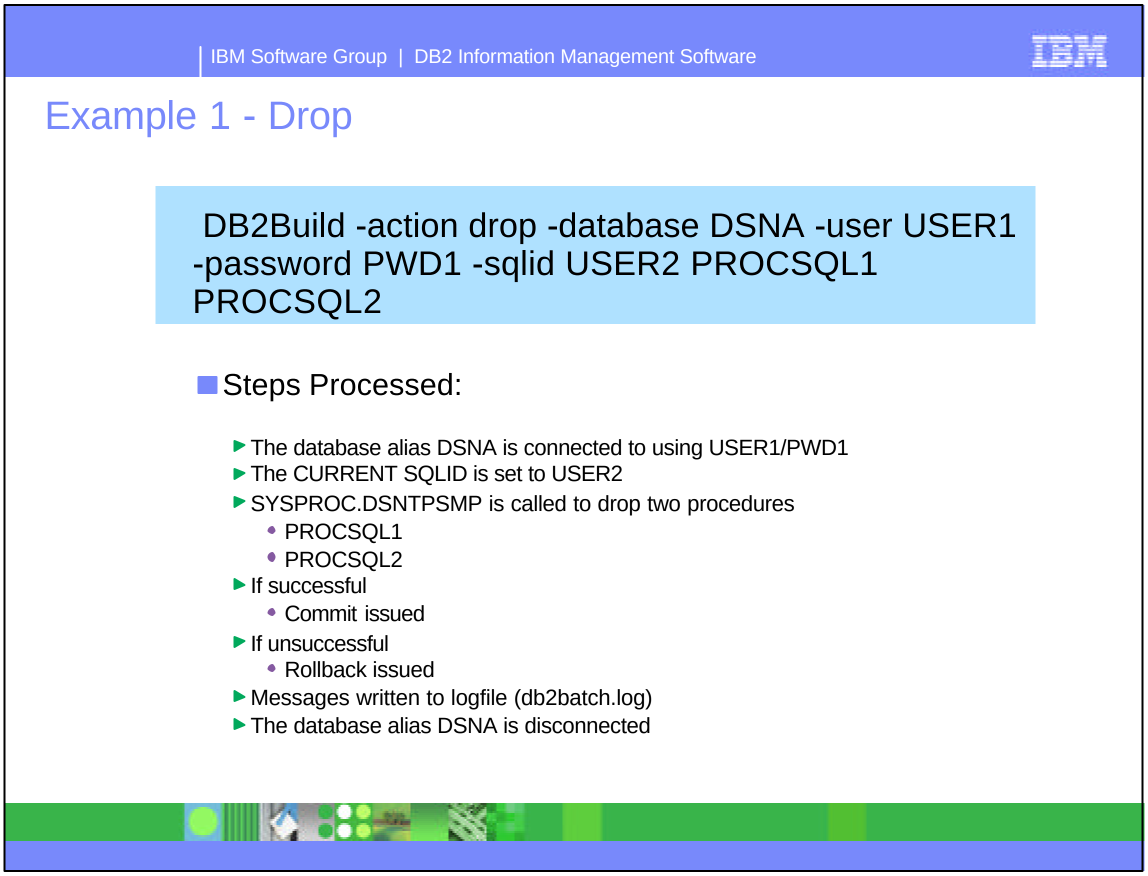
IBM Software Group | DB2 Information Management Software

Step 4 - Run DB2Build

- Positioned on a Command prompt
  - Enter DB2Build with appropriate options
  - Required options
    - database, user, password
- Example 1 - Drop
- Example 2 - Create
- Example 3 - Create -force true
- Example 4 - Alter_source
- Example 5 - Rebind

See following slides for Examples

- Now the build process is ready to run.
- Start a command prompt. Enter the DB2Build command with appropriate options for the requested process.
  - Each DB2Build command requires a minimum of 3 options:
    - database, user and password.
- The remaining slides in this presentation show examples for the actions:
  - Drop
  - Create
  - Create -force true
  - Alter_source
  - Rebind

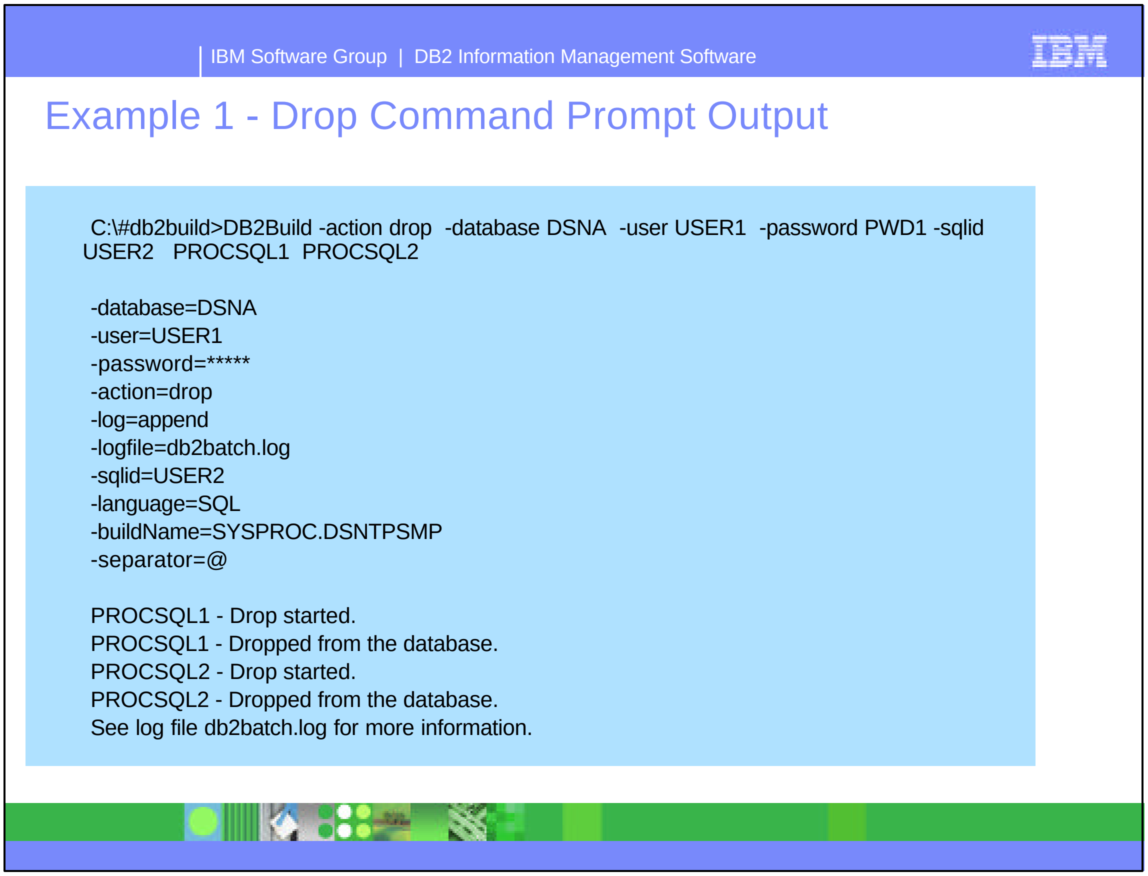


Example 1 - Drop

```
DB2Build -action drop -database DSNA -user USER1
-password PWD1 -sqlid USER2 PROCSQL1
PROCSQL2
```

- Steps Processed:
  - The database alias DSNA is connected to using USER1/PWD1
  - The CURRENT SQLID is set to USER2
  - SYSPROC.DSNTPSMP is called to drop two procedures
    - PROCSQL1
    - PROCSQL2
  - If successful
    - Commit issued
  - If unsuccessful
    - Rollback issued
  - Messages written to logfile (db2batch.log)
  - The database alias DSNA is disconnected

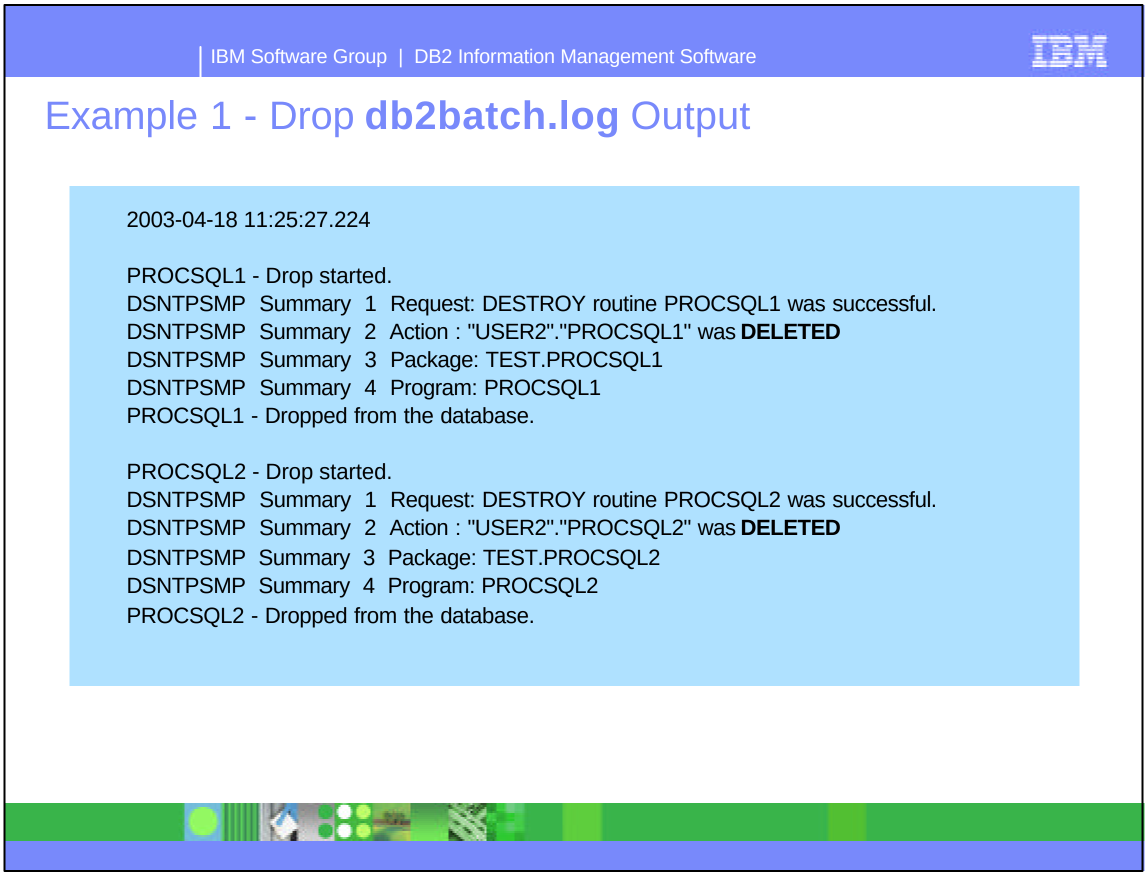
- The Drop action includes all processing information with the DB2Build command. There is no file passed with this action.
- This example is dropping two SPs, PROCSQL1 and PROCSQL2 from the DB2 z/OS server referenced by the Client Configuration Assistant alias, DSNA. The Schema is USER2 for both procedures.
- First the database alias, DSNA is connected to. Each drop is processed individually. If the drop is successful, then it is committed, otherwise the drop is rolled back.
- Messages are written to the db2batch.log file.
- The final step has the database alias, DSNA disconnected.



Example 1 - Drop Command Prompt Output

```
 C:\#db2build>DB2Build -action drop  -database DSNA  -user USER1  -password PWD1 -sqlid
USER2   PROCSQL1  PROCSQL2

 -database=DSNA
 -user=USER1
 -password=*****
 -action=drop
 -log=append
 -logfile=db2batch.log
 -sqlid=USER2
 -language=SQL
 -buildName=SYSPROC.DSNTPSMP
 -separator=@

 PROCSQL1 - Drop started.
 PROCSQL1 - Dropped from the database.
 PROCSQL2 - Drop started.
 PROCSQL2 - Dropped from the database.
 See log file db2batch.log for more information.
```

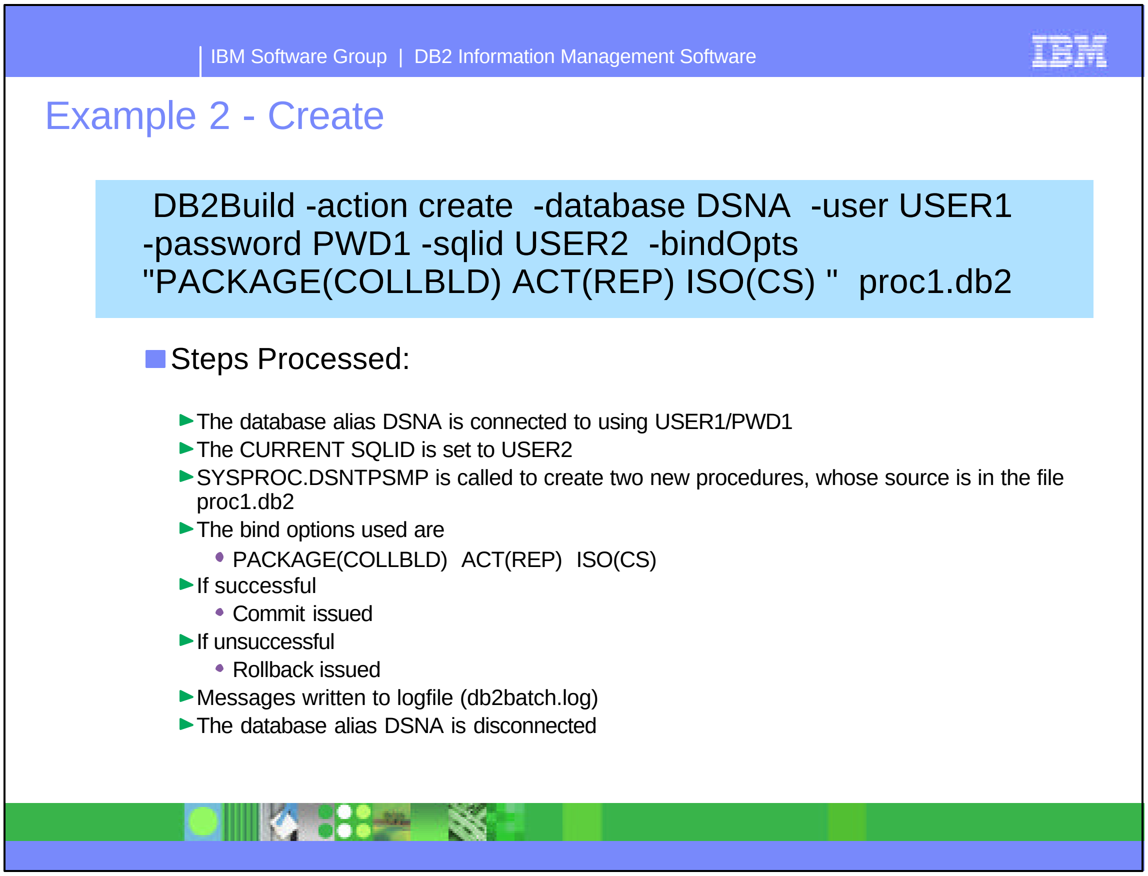
- This slide shows the output of the command processing in the command prompt window. Options that were specified, or other defaults are listed. The command shows the processing of PROCSQL1 drop started, then was dropped from the database, followed by PROCSQL2 drop started and dropped from the database.
- Output for the processing is written to the db2batch.log.



Example 1 - Drop **db2batch.log** Output

```
2003-04-18 11:25:27.224

PROCSQL1 - Drop started.
DSNTPSMP  Summary  1  Request: DESTROY routine PROCSQL1 was successful.
DSNTPSMP  Summary  2  Action : "USER2"."PROCSQL1" was DELETED
DSNTPSMP  Summary  3  Package: TEST.PROCSQL1
DSNTPSMP  Summary  4  Program: PROCSQL1
PROCSQL1 - Dropped from the database.

PROCSQL2 - Drop started.
DSNTPSMP  Summary  1  Request: DESTROY routine PROCSQL2 was successful.
DSNTPSMP  Summary  2  Action : "USER2"."PROCSQL2" was DELETED
DSNTPSMP  Summary  3  Package: TEST.PROCSQL2
DSNTPSMP  Summary  4  Program: PROCSQL2
PROCSQL2 - Dropped from the database.
```

- This slide shows the information written to the db2batch.log for this drop process.
- While not shown here, the options, listed on the previous slide are also written to the log file.

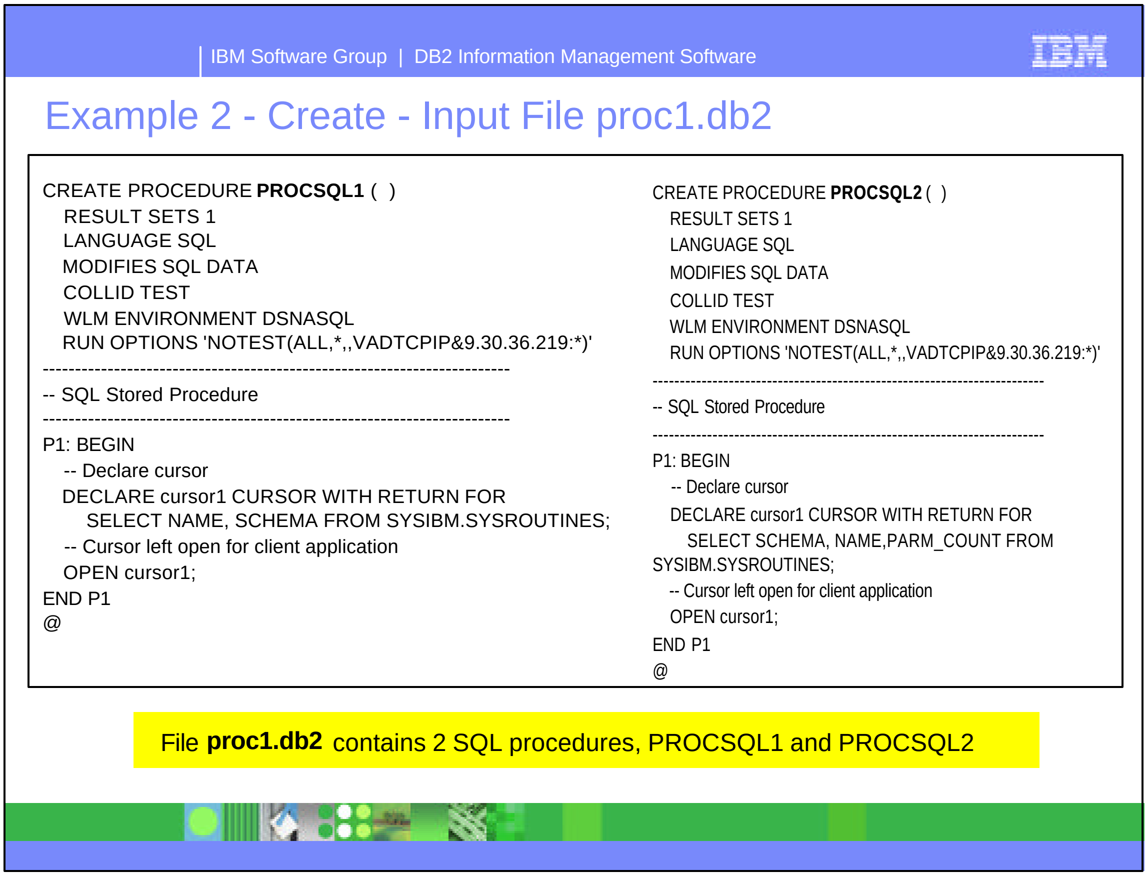


Example 2 - Create

```
DB2Build -action create  -database DSNA  -user USER1
-password PWD1 -sqlid USER2  -bindOpts
"PACKAGE(COLLBLD) ACT(REP) ISO(CS) "  proc1.db2
```

- Steps Processed:
  - The database alias DSNA is connected to using USER1/PWD1
  - The CURRENT SQLID is set to USER2
  - SYSPROC.DSNTPSMP is called to create two new procedures, whose source is in the file proc1.db2
  - The bind options used are
    - PACKAGE(COLLBLD) ACT(REP) ISO(CS)
  - If successful
    - Commit issued
  - If unsuccessful
    - Rollback issued
  - Messages written to logfile (db2batch.log)
  - The database alias DSNA is disconnected

- The Create action includes one or more files on the command. Each file can have one or more SQL SPs included in the file, separated by '@'.
- This example uses the file, proc1.db2. The SQL SPs that are included in this file are listed on the following slide.
- All DB2Build actions first connect to the database alias. If sqlid is specified, then that value is set as CURRENT SQLID. The listed bind options are used. If the create is successful, then it is committed, otherwise the create is rolled back.
- Messages are written to the db2batch.log file.
- The final step has the database alias, DSNA disconnected.



IBM Software Group | DB2 Information Management Software

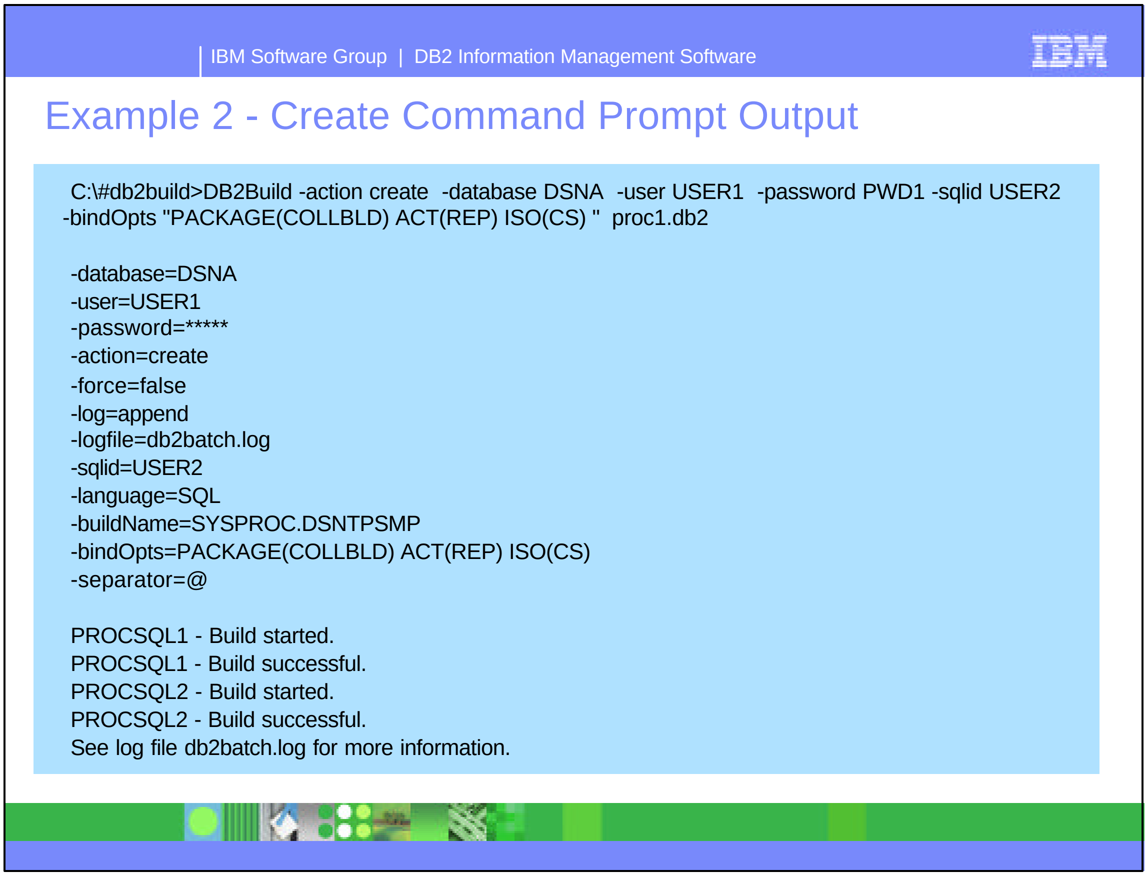
Example 2 - Create - Input File proc1.db2

```
CREATE PROCEDURE PROCSQL1 ( )
   RESULT SETS 1
   LANGUAGE SQL
   MODIFIES SQL DATA
   COLLID TEST
   WLM ENVIRONMENT DSNASQL
   RUN OPTIONS 'NOTEST(ALL,*,,VADTCPIP&9.30.36.219:*)'
------------------------------------------------------------------------
-- SQL Stored Procedure
------------------------------------------------------------------------
P1: BEGIN
   -- Declare cursor
   DECLARE cursor1 CURSOR WITH RETURN FOR
      SELECT NAME, SCHEMA FROM SYSIBM.SYSROUTINES;
   -- Cursor left open for client application
   OPEN cursor1;
END P1
@
```

```
CREATE PROCEDURE PROCSQL2 ( )
   RESULT SETS 1
   LANGUAGE SQL
   MODIFIES SQL DATA
   COLLID TEST
   WLM ENVIRONMENT DSNASQL
   RUN OPTIONS 'NOTEST(ALL,*,,VADTCPIP&9.30.36.219:*)'
------------------------------------------------------------------------
-- SQL Stored Procedure
------------------------------------------------------------------------
P1: BEGIN
   -- Declare cursor
   DECLARE cursor1 CURSOR WITH RETURN FOR
      SELECT SCHEMA, NAME,PARM_COUNT FROM
SYSIBM.SYSROUTINES;
   -- Cursor left open for client application
   OPEN cursor1;
END P1
@
```

File **proc1.db2** contains 2 SQL procedures, PROCSQL1 and PROCSQL2

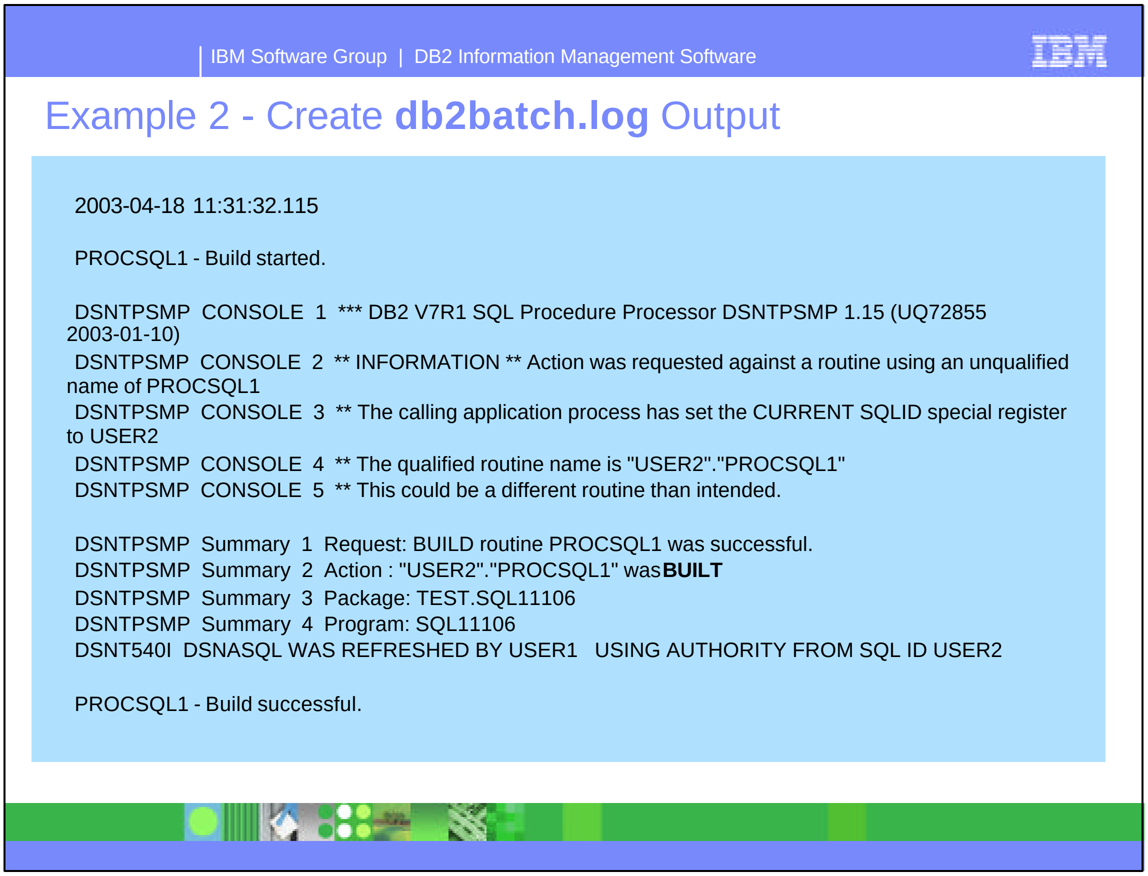
- This slide shows the contents of the PROCSQL1 and PROCSQL2 stored procedures included in the file, proc1.db2.



Example 2 - Create Command Prompt Output

```
 C:\#db2build>DB2Build -action create  -database DSNA  -user USER1  -password PWD1 -sqlid USER2
-bindOpts "PACKAGE(COLLBLD) ACT(REP) ISO(CS) "  proc1.db2

 -database=DSNA
 -user=USER1
 -password=*****
 -action=create
 -force=false
 -log=append
 -logfile=db2batch.log
 -sqlid=USER2
 -language=SQL
 -buildName=SYSPROC.DSNTPSMP
 -bindOpts=PACKAGE(COLLBLD) ACT(REP) ISO(CS)
 -separator=@

 PROCSQL1 - Build started.
 PROCSQL1 - Build successful.
 PROCSQL2 - Build started.
 PROCSQL2 - Build successful.
 See log file db2batch.log for more information.
```

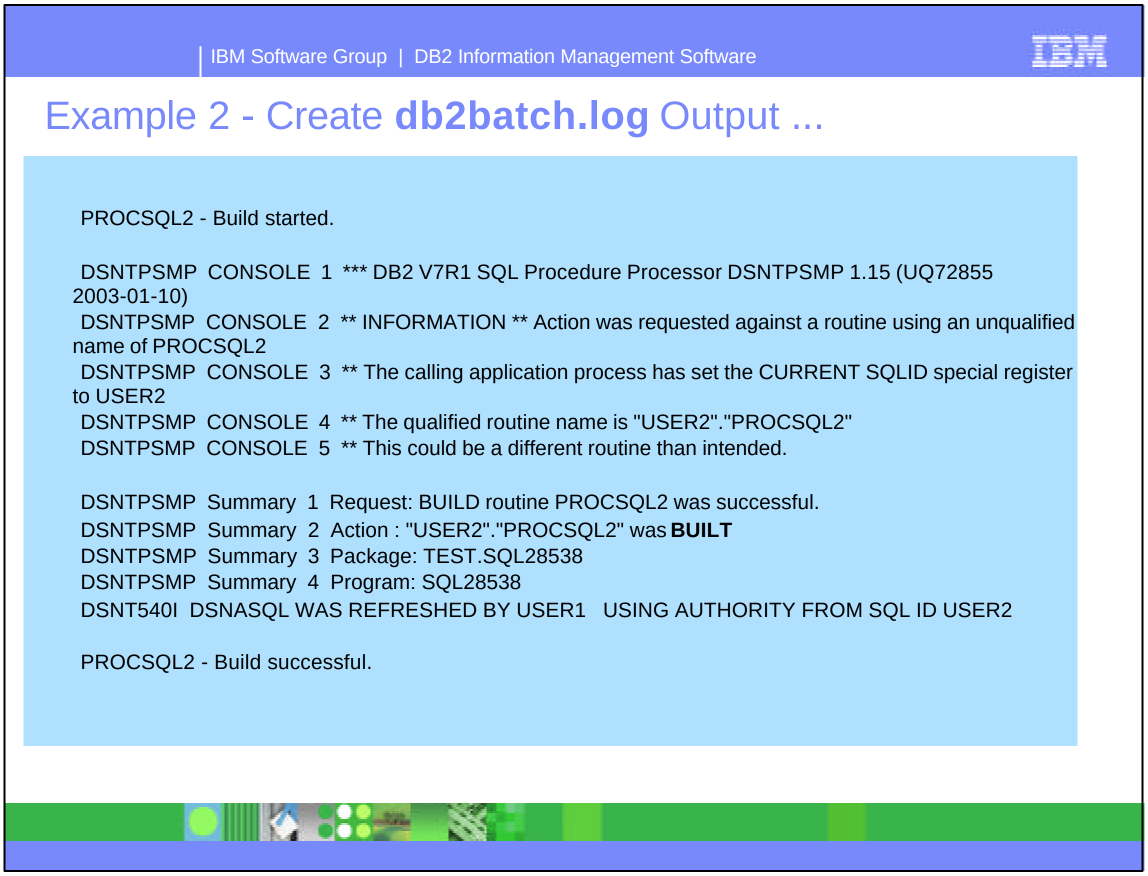
- This slide shows the output of the command processing in the command prompt window. Options that were specified, or other defaults are listed. The command shows the processing of PROCSQL1 build started, then successful, followed by PROCSQL2 build started and successful.
- Output for the processing is written to the db2batch.log.



Example 2 - Create **db2batch.log** Output

```
 2003-04-18 11:31:32.115

 PROCSQL1 - Build started.

 DSNTPSMP  CONSOLE  1  *** DB2 V7R1 SQL Procedure Processor DSNTPSMP 1.15 (UQ72855
2003-01-10)
 DSNTPSMP  CONSOLE  2  ** INFORMATION ** Action was requested against a routine using an unqualified
name of PROCSQL1
 DSNTPSMP  CONSOLE  3  ** The calling application process has set the CURRENT SQLID special register
to USER2
 DSNTPSMP  CONSOLE  4  ** The qualified routine name is "USER2"."PROCSQL1"
 DSNTPSMP  CONSOLE  5  ** This could be a different routine than intended.

 DSNTPSMP  Summary  1  Request: BUILD routine PROCSQL1 was successful.
 DSNTPSMP  Summary  2  Action : "USER2"."PROCSQL1" was BUILT
 DSNTPSMP  Summary  3  Package: TEST.SQL11106
 DSNTPSMP  Summary  4  Program: SQL11106
 DSNT540I  DSNASQL WAS REFRESHED BY USER1   USING AUTHORITY FROM SQL ID USER2

 PROCSQL1 - Build successful.
```

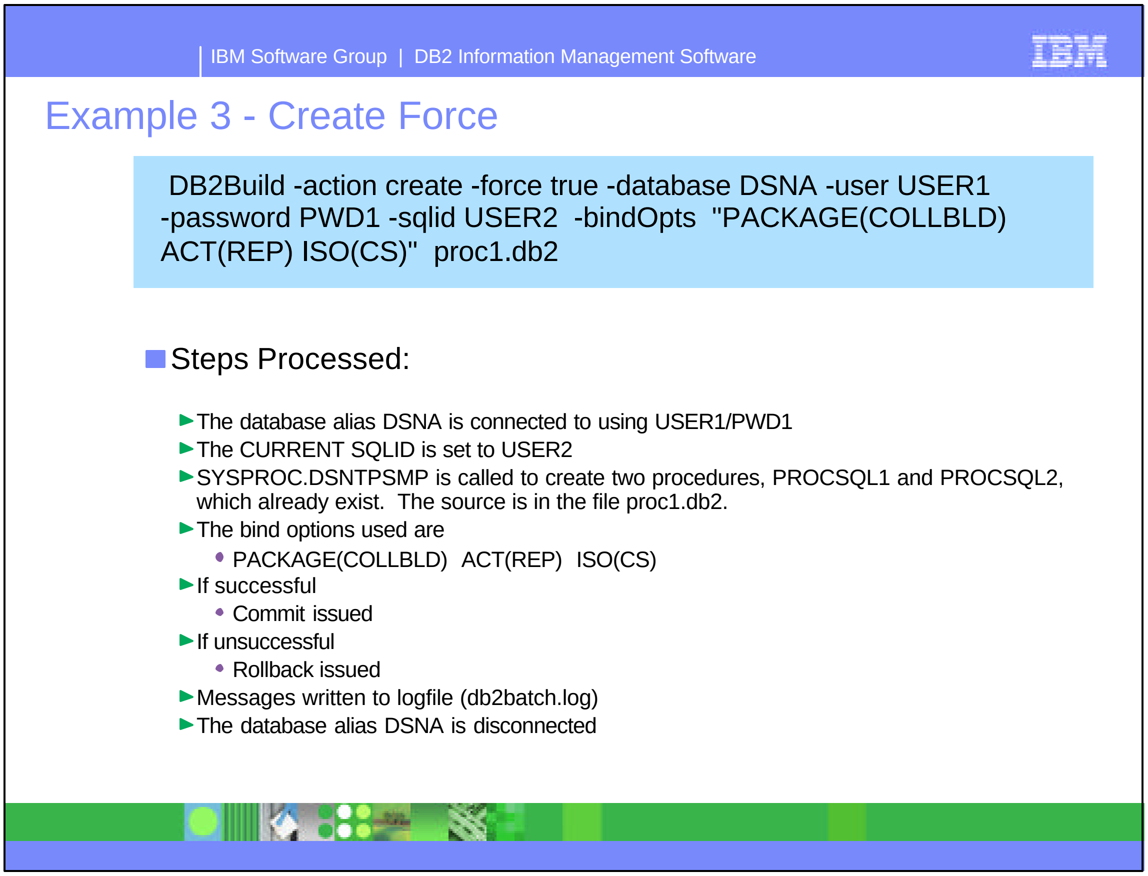
- This slide shows the information written to the db2batch.log for this create process for PROCSQL1. The **built** summary value is used for create of new stored procedures.
- While not shown here, the options, listed on the previous slide are also written to the log file.



Example 2 - Create **db2batch.log** Output ...

```
 PROCSQL2 - Build started.

 DSNTPSMP  CONSOLE  1  *** DB2 V7R1 SQL Procedure Processor DSNTPSMP 1.15 (UQ72855
2003-01-10)
 DSNTPSMP  CONSOLE  2  ** INFORMATION ** Action was requested against a routine using an unqualified
name of PROCSQL2
 DSNTPSMP  CONSOLE  3  ** The calling application process has set the CURRENT SQLID special register
to USER2
 DSNTPSMP  CONSOLE  4  ** The qualified routine name is "USER2"."PROCSQL2"
 DSNTPSMP  CONSOLE  5  ** This could be a different routine than intended.

 DSNTPSMP  Summary  1  Request: BUILD routine PROCSQL2 was successful.
 DSNTPSMP  Summary  2  Action : "USER2"."PROCSQL2" was BUILT
 DSNTPSMP  Summary  3  Package: TEST.SQL28538
 DSNTPSMP  Summary  4  Program: SQL28538
 DSNT540I  DSNASQL WAS REFRESHED BY USER1   USING AUTHORITY FROM SQL ID USER2

 PROCSQL2 - Build successful.
```

- This slide shows the information written to the db2batch.log for the create process for PROCSQL2.

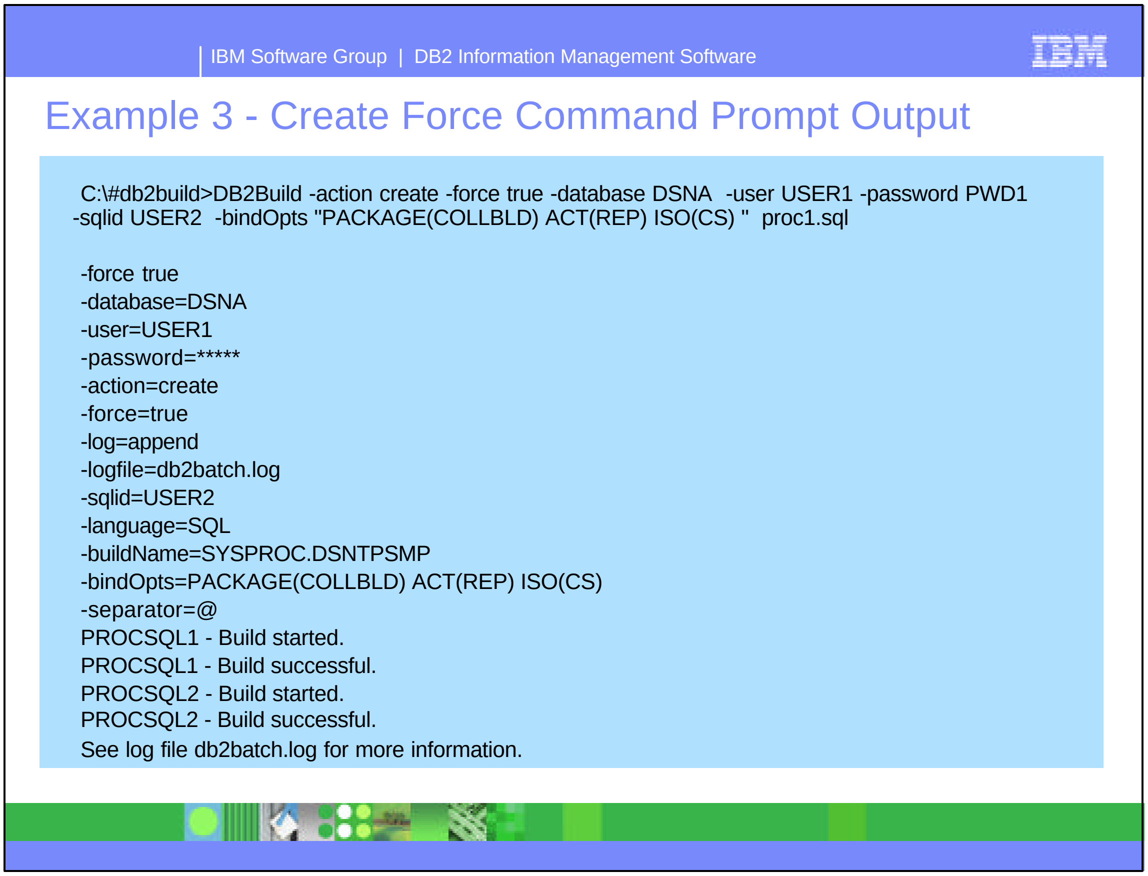


Example 3 - Create Force

```
DB2Build -action create -force true -database DSNA -user USER1
-password PWD1 -sqlid USER2  -bindOpts  "PACKAGE(COLLBLD)
ACT(REP) ISO(CS)"  proc1.db2
```

- Steps Processed:
  - The database alias DSNA is connected to using USER1/PWD1
  - The CURRENT SQLID is set to USER2
  - SYSPROC.DSNTPSMP is called to create two procedures, PROCSQL1 and PROCSQL2, which already exist. The source is in the file proc1.db2.
  - The bind options used are
    - PACKAGE(COLLBLD) ACT(REP) ISO(CS)
  - If successful
    - Commit issued
  - If unsuccessful
    - Rollback issued
  - Messages written to logfile (db2batch.log)
  - The database alias DSNA is disconnected

- The processing for the Create with the '-force true' option is the same as an initial create except the SQL SP already exists. As with Create, one or more files can be specified on the command. Each file can have one or more SQL SPs included in the file, separated by '@'.
- This example uses the file, proc1.db2 which includes PROCSQL1 and PROCSQL2.
- All DB2Build actions first connect to the database alias. If sqlid is specified, then that value is set as CURRENT SQLID. The listed bind options are used. If the create with force is successful, then it is committed, otherwise the create is rolled back.
- Messages are written to the db2batch.log file.
- The final step has the database alias, DSNA disconnected.

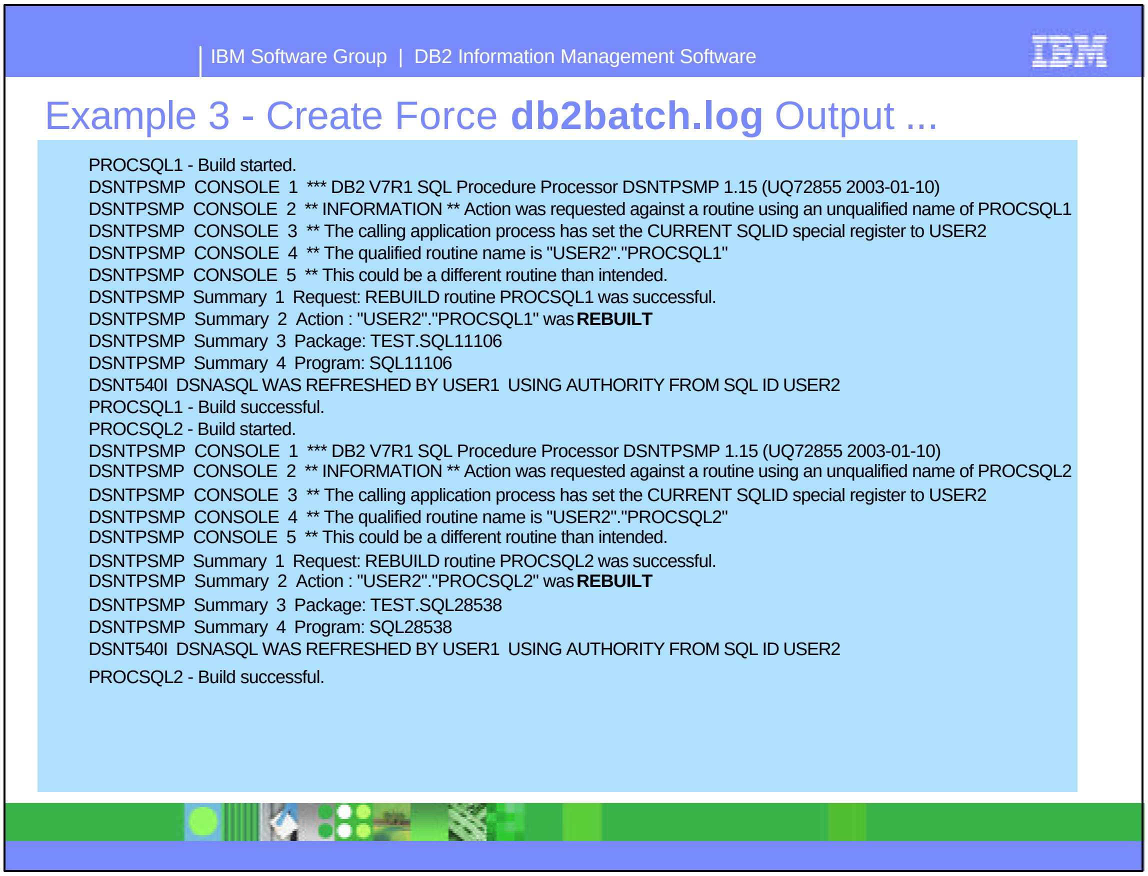


IBM Software Group | DB2 Information Management Software

Example 3 - Create Force Command Prompt Output

```
 C:\#db2build>DB2Build -action create -force true -database DSNA  -user USER1 -password PWD1
-sqlid USER2  -bindOpts "PACKAGE(COLLBLD) ACT(REP) ISO(CS) "  proc1.sql

 -force true
 -database=DSNA
 -user=USER1
 -password=*****
 -action=create
 -force=true
 -log=append
 -logfile=db2batch.log
 -sqlid=USER2
 -language=SQL
 -buildName=SYSPROC.DSNTPSMP
 -bindOpts=PACKAGE(COLLBLD) ACT(REP) ISO(CS)
 -separator=@
 PROCSQL1 - Build started.
 PROCSQL1 - Build successful.
 PROCSQL2 - Build started.
 PROCSQL2 - Build successful.
 See log file db2batch.log for more information.
```

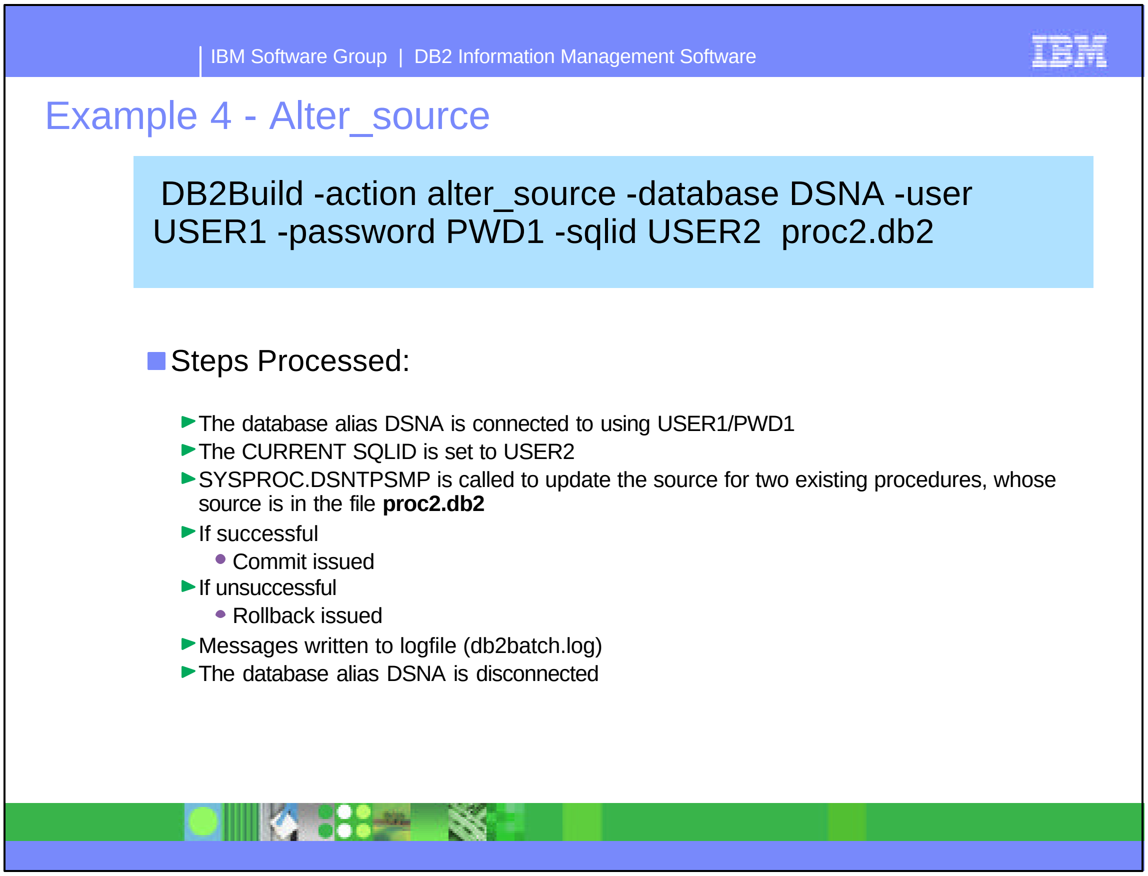
- This slide shows the output of the command processing in the command prompt window for the Create with force true option. Options that were specified, or other defaults are listed. The command shows the processing of PROCSQL1 build started, then successful, followed by PROCSQL2 build started and successful..
- Output for the processing is written to the db2batch.log.



Example 3 - Create Force **db2batch.log** Output ...

```
PROCSQL1 - Build started.
DSNTPSMP  CONSOLE  1  *** DB2 V7R1 SQL Procedure Processor DSNTPSMP 1.15 (UQ72855 2003-01-10)
DSNTPSMP  CONSOLE  2  ** INFORMATION ** Action was requested against a routine using an unqualified name of PROCSQL1
DSNTPSMP  CONSOLE  3  ** The calling application process has set the CURRENT SQLID special register to USER2
DSNTPSMP  CONSOLE  4  ** The qualified routine name is "USER2"."PROCSQL1"
DSNTPSMP  CONSOLE  5  ** This could be a different routine than intended.
DSNTPSMP  Summary  1  Request: REBUILD routine PROCSQL1 was successful.
DSNTPSMP  Summary  2  Action : "USER2"."PROCSQL1" was REBUILT
DSNTPSMP  Summary  3  Package: TEST.SQL11106
DSNTPSMP  Summary  4  Program: SQL11106
DSNT540I  DSNASQL WAS REFRESHED BY USER1  USING AUTHORITY FROM SQL ID USER2
PROCSQL1 - Build successful.
PROCSQL2 - Build started.
DSNTPSMP  CONSOLE  1  *** DB2 V7R1 SQL Procedure Processor DSNTPSMP 1.15 (UQ72855 2003-01-10)
DSNTPSMP  CONSOLE  2  ** INFORMATION ** Action was requested against a routine using an unqualified name of PROCSQL2
DSNTPSMP  CONSOLE  3  ** The calling application process has set the CURRENT SQLID special register to USER2
DSNTPSMP  CONSOLE  4  ** The qualified routine name is "USER2"."PROCSQL2"
DSNTPSMP  CONSOLE  5  ** This could be a different routine than intended.
DSNTPSMP  Summary  1  Request: REBUILD routine PROCSQL2 was successful.
DSNTPSMP  Summary  2  Action : "USER2"."PROCSQL2" was REBUILT
DSNTPSMP  Summary  3  Package: TEST.SQL28538
DSNTPSMP  Summary  4  Program: SQL28538
DSNT540I  DSNASQL WAS REFRESHED BY USER1  USING AUTHORITY FROM SQL ID USER2
PROCSQL2 - Build successful.
```

- This slide shows the information written to the db2batch.log for this create with force true specified for processing PROCSQL1. The **rebuilt** summary value is used for create with force true.
- While not shown here, the options, listed on the previous slide are also written to the log file.

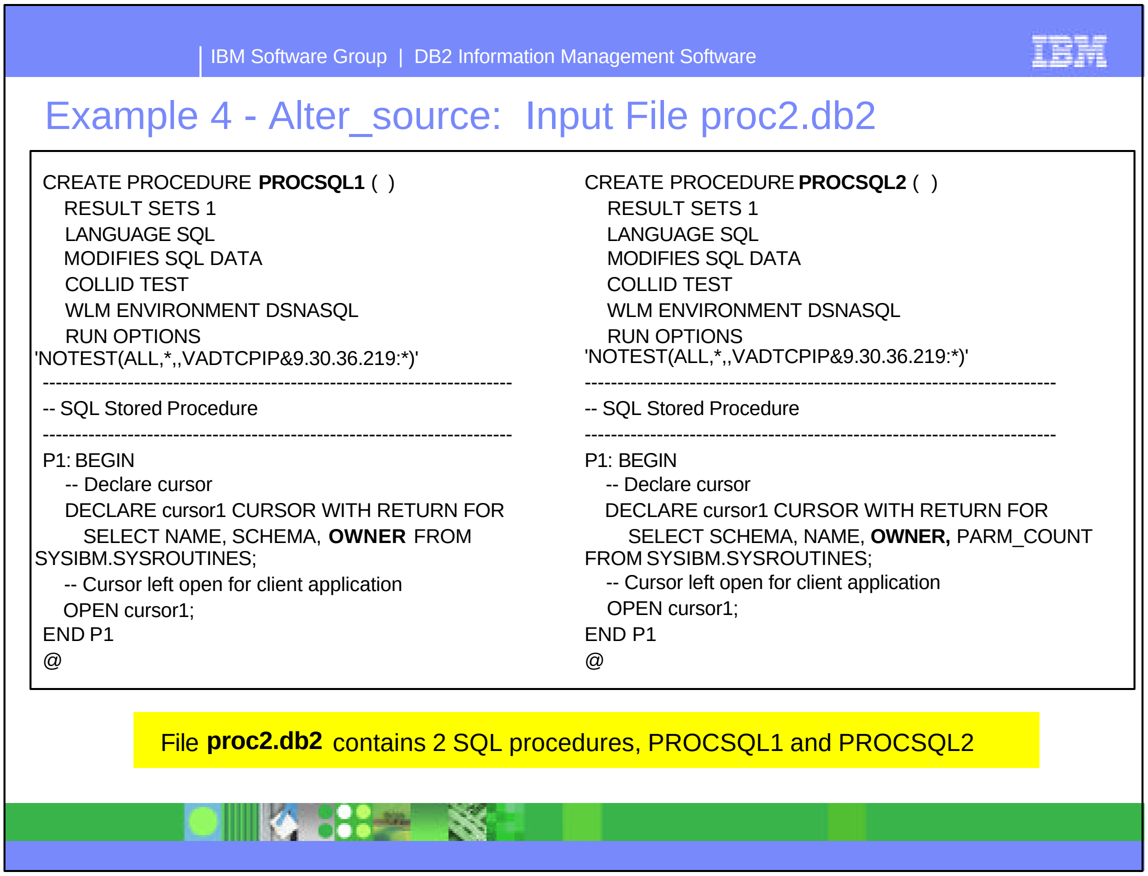


Example 4 - Alter_source

```
DB2Build -action alter_source -database DSNA -user
USER1 -password PWD1 -sqlid USER2  proc2.db2
```

- Steps Processed:
  - The database alias DSNA is connected to using USER1/PWD1
  - The CURRENT SQLID is set to USER2
  - SYSPROC.DSNTPSMP is called to update the source for two existing procedures, whose source is in the file **proc2.db2**
  - If successful
    - Commit issued
  - If unsuccessful
    - Rollback issued
  - Messages written to logfile (db2batch.log)
  - The database alias DSNA is disconnected

- The processing for the Alter_source option. This option only changes the source of the SQL SP. Each file can have one or more SQL SPs included in the file, separated by '@'.
- This example uses the file, proc2.db2 which includes PROCSQL1 and PROCSQL2. The following slide includes the source for proc2.db2.
- All DB2Build actions first connect to the database alias. If sqlid is specified, then that value is set as CURRENT SQLID. If the alter_source is successful, then the updates are committed, otherwise the alter_source is rolled back.
- Messages are written to the db2batch.log file.
- The final step has the database alias, DSNA disconnected.



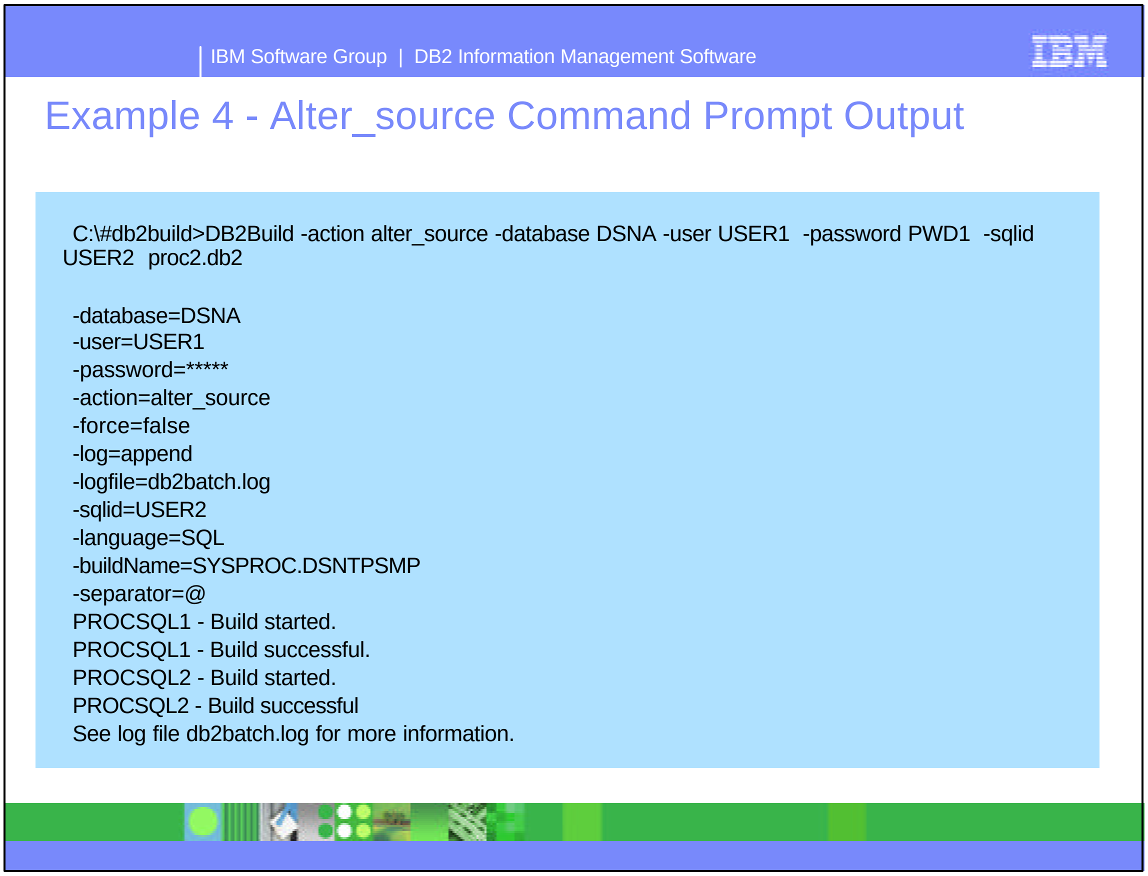
Example 4 - Alter_source: Input File proc2.db2

```
CREATE PROCEDURE PROCSQL1 ( )
   RESULT SETS 1
   LANGUAGE SQL
   MODIFIES SQL DATA
   COLLID TEST
   WLM ENVIRONMENT DSNASQL
   RUN OPTIONS
'NOTEST(ALL,*,,VADTCPIP&9.30.36.219:*)'
------------------------------------------------------------------------
-- SQL Stored Procedure
------------------------------------------------------------------------
P1: BEGIN
   -- Declare cursor
   DECLARE cursor1 CURSOR WITH RETURN FOR
      SELECT NAME, SCHEMA, OWNER FROM
SYSIBM.SYSROUTINES;
   -- Cursor left open for client application
   OPEN cursor1;
END P1
@
```

```
CREATE PROCEDURE PROCSQL2 ( )
   RESULT SETS 1
   LANGUAGE SQL
   MODIFIES SQL DATA
   COLLID TEST
   WLM ENVIRONMENT DSNASQL
   RUN OPTIONS
'NOTEST(ALL,*,,VADTCPIP&9.30.36.219:*)'
------------------------------------------------------------------------
-- SQL Stored Procedure
------------------------------------------------------------------------
P1: BEGIN
   -- Declare cursor
   DECLARE cursor1 CURSOR WITH RETURN FOR
      SELECT SCHEMA, NAME, OWNER, PARM_COUNT
FROM SYSIBM.SYSROUTINES;
   -- Cursor left open for client application
   OPEN cursor1;
END P1
@
```

File **proc2.db2** contains 2 SQL procedures, PROCSQL1 and PROCSQL2

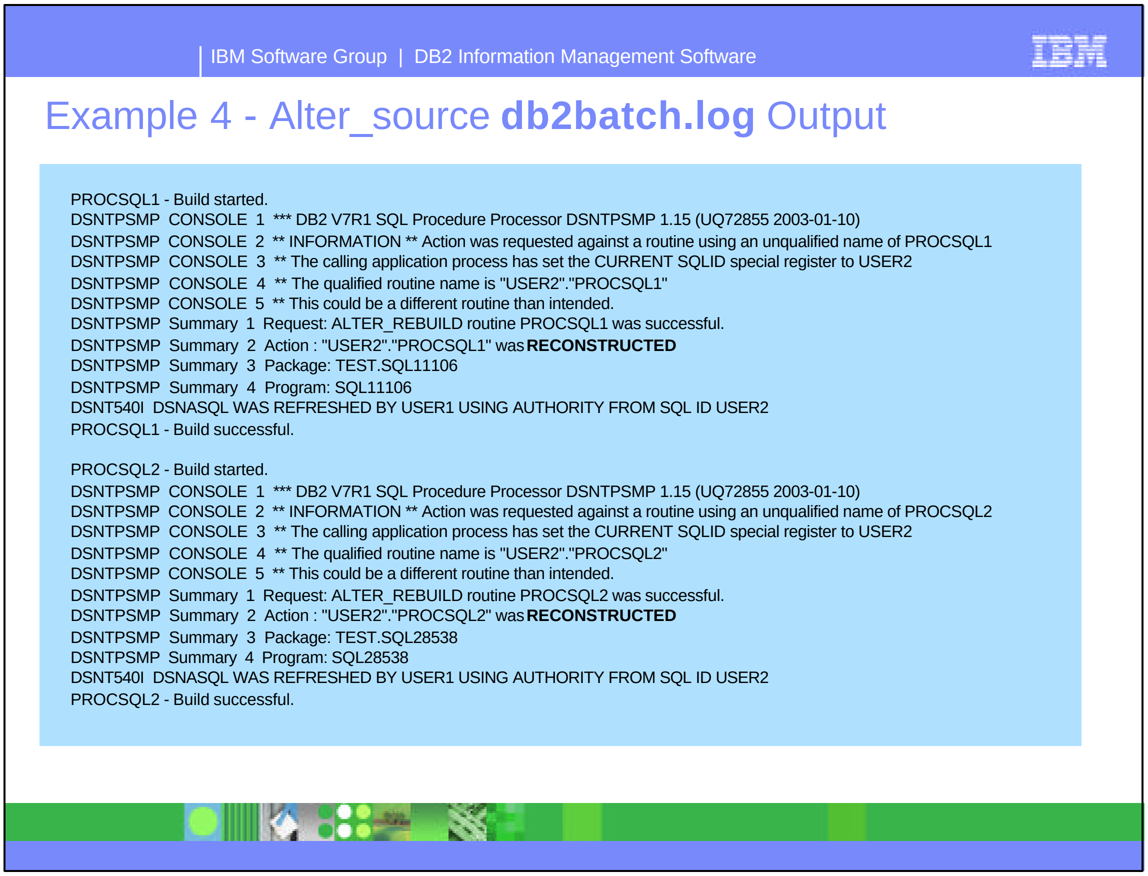
- File, proc2.db2 includes the source for the PROCSQL1 and PROCSQL2. The source has been updated to add 'OWNER' for a column to be selected in the SQL statement.



Example 4 - Alter_source Command Prompt Output

```
  C:\#db2build>DB2Build -action alter_source -database DSNA -user USER1  -password PWD1  -sqlid
USER2  proc2.db2

  -database=DSNA
  -user=USER1
  -password=*****
  -action=alter_source
  -force=false
  -log=append
  -logfile=db2batch.log
  -sqlid=USER2
  -language=SQL
  -buildName=SYSPROC.DSNTPSMP
  -separator=@
  PROCSQL1 - Build started.
  PROCSQL1 - Build successful.
  PROCSQL2 - Build started.
  PROCSQL2 - Build successful
  See log file db2batch.log for more information.
```

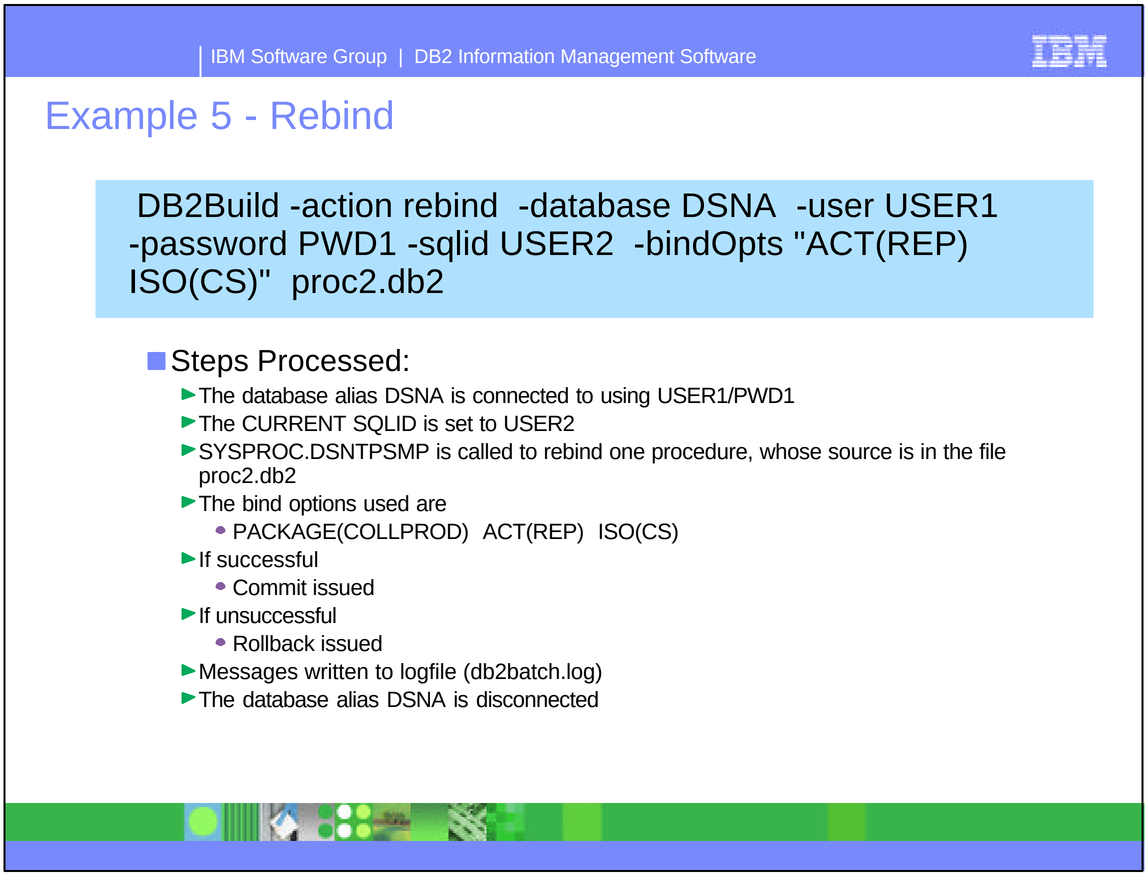
- This slide shows the output of the command processing in the command prompt window for the alter_source option. Options that were specified, or other defaults are listed. The command shows the processing of PROCSQL1 build started, then successful, followed by PROCSQL2 build started and successful.
- Output for the processing is written to the db2batch.log.



Example 4 - Alter_source **db2batch.log** Output

```
PROCSQL1 - Build started.
DSNTPSMP  CONSOLE  1  *** DB2 V7R1 SQL Procedure Processor DSNTPSMP 1.15 (UQ72855 2003-01-10)
DSNTPSMP  CONSOLE  2  ** INFORMATION ** Action was requested against a routine using an unqualified name of PROCSQL1
DSNTPSMP  CONSOLE  3  ** The calling application process has set the CURRENT SQLID special register to USER2
DSNTPSMP  CONSOLE  4  ** The qualified routine name is "USER2"."PROCSQL1"
DSNTPSMP  CONSOLE  5  ** This could be a different routine than intended.
DSNTPSMP  Summary  1  Request: ALTER_REBUILD routine PROCSQL1 was successful.
DSNTPSMP  Summary  2  Action : "USER2"."PROCSQL1" was RECONSTRUCTED
DSNTPSMP  Summary  3  Package: TEST.SQL11106
DSNTPSMP  Summary  4  Program: SQL11106
DSNT540I  DSNASQL WAS REFRESHED BY USER1 USING AUTHORITY FROM SQL ID USER2
PROCSQL1 - Build successful.

PROCSQL2 - Build started.
DSNTPSMP  CONSOLE  1  *** DB2 V7R1 SQL Procedure Processor DSNTPSMP 1.15 (UQ72855 2003-01-10)
DSNTPSMP  CONSOLE  2  ** INFORMATION ** Action was requested against a routine using an unqualified name of PROCSQL2
DSNTPSMP  CONSOLE  3  ** The calling application process has set the CURRENT SQLID special register to USER2
DSNTPSMP  CONSOLE  4  ** The qualified routine name is "USER2"."PROCSQL2"
DSNTPSMP  CONSOLE  5  ** This could be a different routine than intended.
DSNTPSMP  Summary  1  Request: ALTER_REBUILD routine PROCSQL2 was successful.
DSNTPSMP  Summary  2  Action : "USER2"."PROCSQL2" was RECONSTRUCTED
DSNTPSMP  Summary  3  Package: TEST.SQL28538
DSNTPSMP  Summary  4  Program: SQL28538
DSNT540I  DSNASQL WAS REFRESHED BY USER1 USING AUTHORITY FROM SQL ID USER2
PROCSQL2 - Build successful.
```

- This slide shows the information written to the db2batch.log for this create process for PROCSQL1. The **reconstructed** summary value is used for the alter_source action.
- While not shown here, the options, listed on the previous slide are also written to the log file.

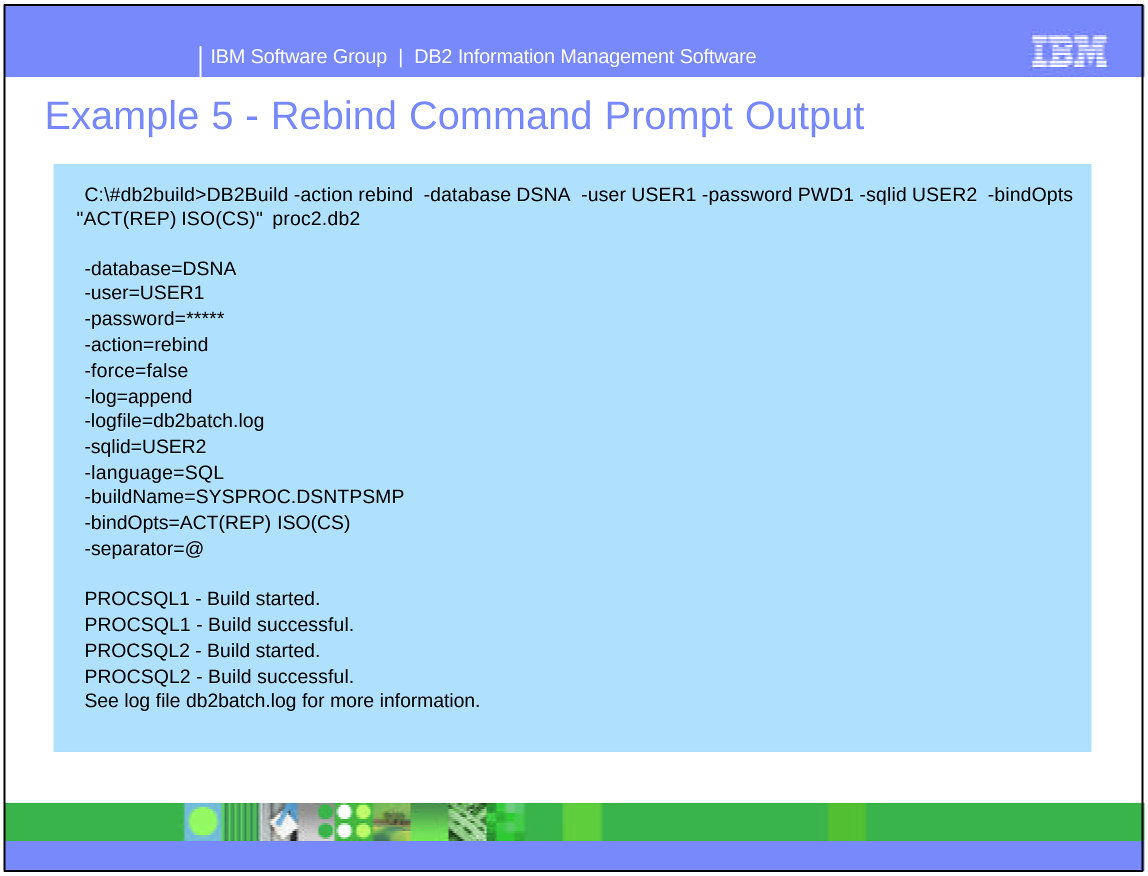


Example 5 - Rebind

```
DB2Build -action rebind  -database DSNA  -user USER1
-password PWD1 -sqlid USER2  -bindOpts "ACT(REP)
ISO(CS)"  proc2.db2
```

- Steps Processed:
  - The database alias DSNA is connected to using USER1/PWD1
  - The CURRENT SQLID is set to USER2
  - SYSPROC.DSNTPSMP is called to rebind one procedure, whose source is in the file proc2.db2
  - The bind options used are
    - PACKAGE(COLLPROD) ACT(REP) ISO(CS)
  - If successful
    - Commit issued
  - If unsuccessful
    - Rollback issued
  - Messages written to logfile (db2batch.log)
  - The database alias DSNA is disconnected

- When using the Rebind action, one or more files can be specified on the command. Each file can have one or more SQL SPs included in the file, separated by '@'.
- This example uses the file, proc2.db2 which includes PROCSQL1 and PROCSQL2.
- All DB2Build actions first connect to the database alias. If sqlid is specified, then that value is set as CURRENT SQLID. The listed bind options are used. If the rebind is successful, then it is committed, otherwise the create is rolled back.
- Messages are written to the db2batch.log file.
- The final step has the database alias, DSNA disconnected.

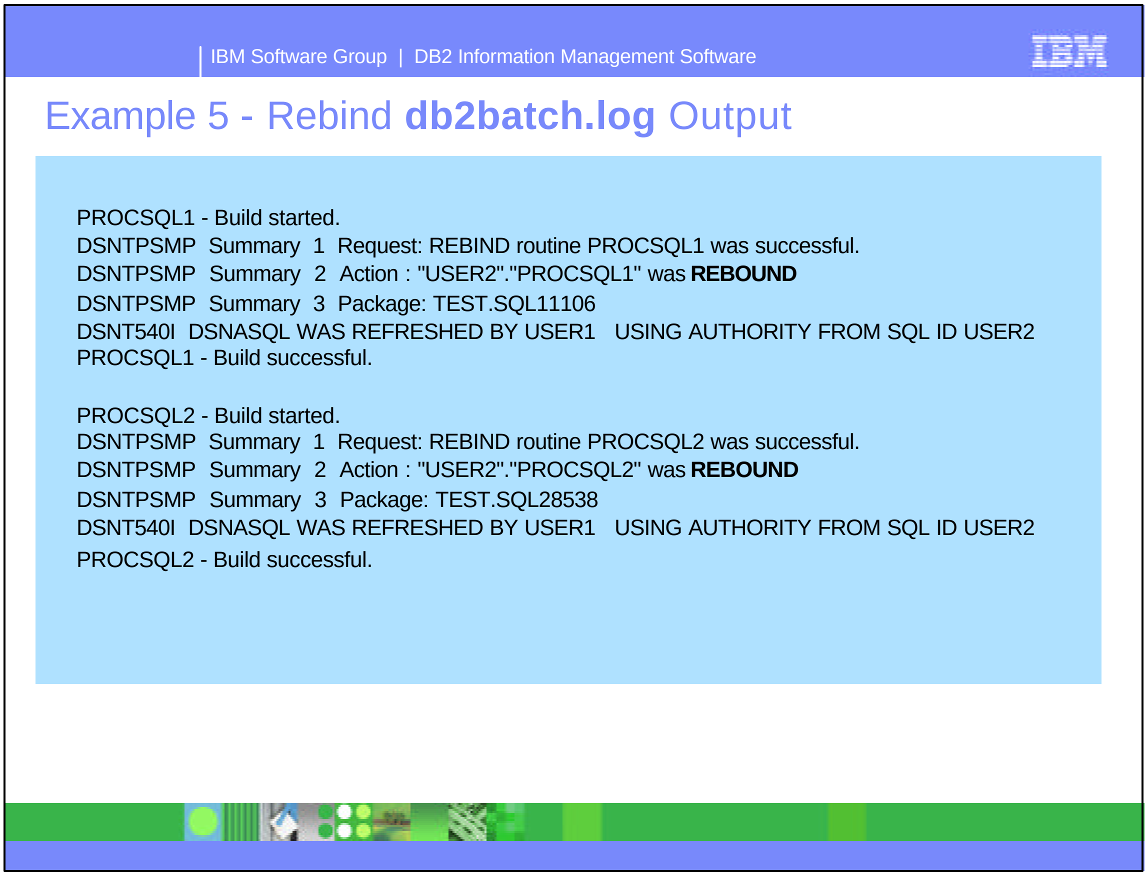


IBM Software Group | DB2 Information Management Software

Example 5 - Rebind Command Prompt Output

```
  C:\#db2build>DB2Build -action rebind  -database DSNA  -user USER1 -password PWD1 -sqlid USER2  -bindOpts
"ACT(REP) ISO(CS)"  proc2.db2

  -database=DSNA
  -user=USER1
  -password=*****
  -action=rebind
  -force=false
  -log=append
  -logfile=db2batch.log
  -sqlid=USER2
  -language=SQL
  -buildName=SYSPROC.DSNTPSMP
  -bindOpts=ACT(REP) ISO(CS)
  -separator=@

  PROCSQL1 - Build started.
  PROCSQL1 - Build successful.
  PROCSQL2 - Build started.
  PROCSQL2 - Build successful.
  See log file db2batch.log for more information.
```

- This slide shows the output of the command processing in the command prompt window for the Rebind option. Options that were specified, or other defaults are listed. The command shows the processing of PROCSQL1 build started, then successful, followed by PROCSQL2 build started and successful.
- Output for the processing is written to the db2batch.log.



Example 5 - Rebind **db2batch.log** Output

```
PROCSQL1 - Build started.
DSNTPSMP  Summary  1  Request: REBIND routine PROCSQL1 was successful.
DSNTPSMP  Summary  2  Action : "USER2"."PROCSQL1" was REBOUND
DSNTPSMP  Summary  3  Package: TEST.SQL11106
DSNT540I  DSNASQL WAS REFRESHED BY USER1   USING AUTHORITY FROM SQL ID USER2
PROCSQL1 - Build successful.

PROCSQL2 - Build started.
DSNTPSMP  Summary  1  Request: REBIND routine PROCSQL2 was successful.
DSNTPSMP  Summary  2  Action : "USER2"."PROCSQL2" was REBOUND
DSNTPSMP  Summary  3  Package: TEST.SQL28538
DSNT540I  DSNASQL WAS REFRESHED BY USER1   USING AUTHORITY FROM SQL ID USER2
PROCSQL2 - Build successful.
```

- This slide shows the information written to the db2batch.log for this rebind process for PROCSQL1. The **rebound** summary value is used for the rebind action.
- While not shown here, the options listed on the previous slide are also written to the log file.