SQL and SPUFI

Objectives

SQL and SPUFI

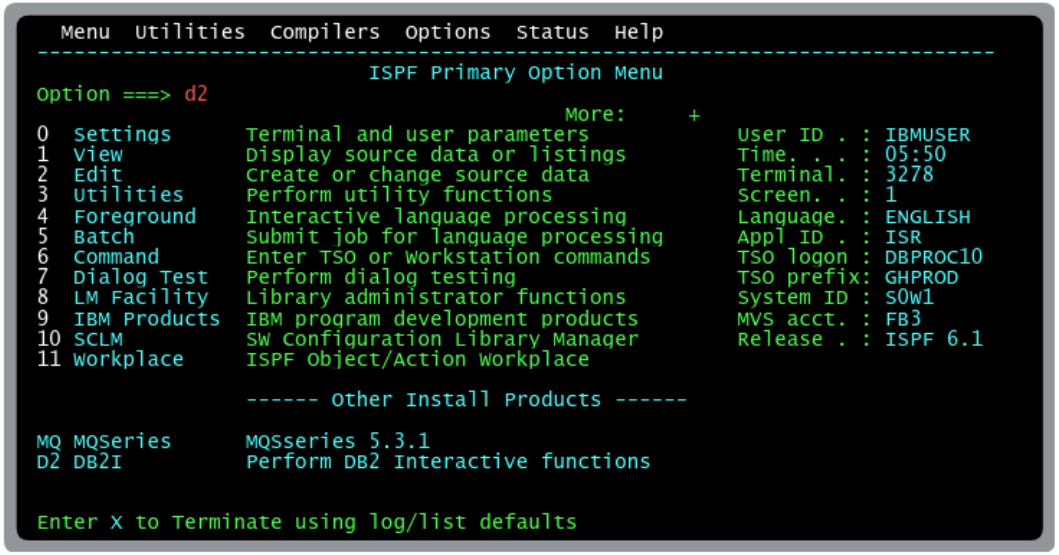
In this module, you will look at invoking some of the SQL statements covered in the previous module interactively using the TSO/ISPF SPUFI facility.

After completing this module, you will be able to:

- Access SPUFI and Modify Panel Defaults
- Code and Execute SQL Statements from an Input Data Set Using SPUFI
- Interpret Output Produced from SPUFI

As you begin coding SQL you will need to test it against Db2 data. A quick method of performing this is to use the SQL Processor Using File Input (SPUFI) facility, which is available through TSO/ISPF.

The following pages describe how SPUFI is accessed and the type of information you need to provide in order to invoke your SQL statements.

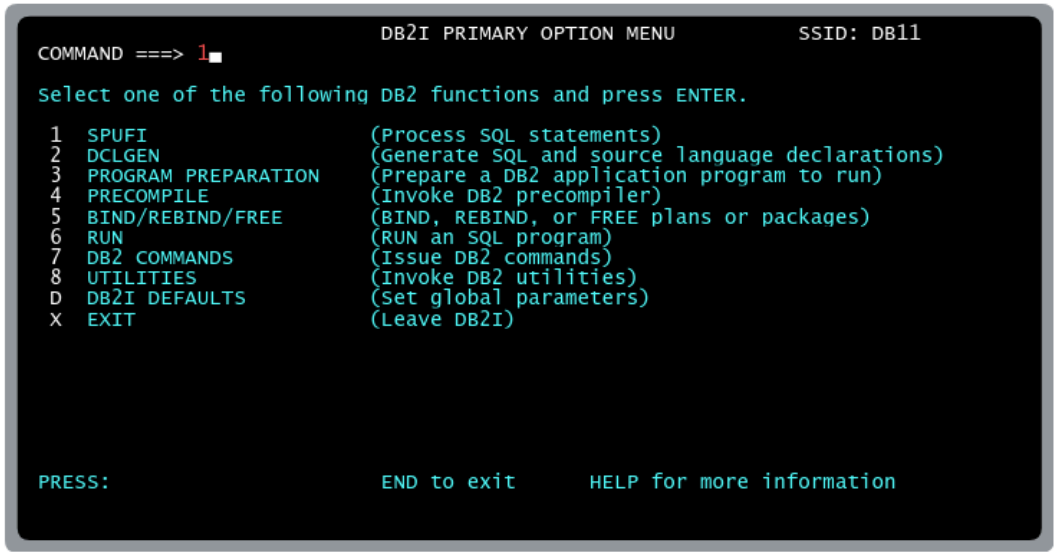


Step 1 of 2

Because SPUFI is run through TSO/ISPF you will need to have been assigned access to this product by your security administrator who will provide you with a logon ID and password. After logging on, an ISPF Primary Option panel similar to the one shown here will be displayed.

In this example you can see option D2 at the bottom of the screen, which is used to access the SPUFI facility on this system. Note that your site's systems programmer may use other panel option selection codes, or that you may need to scroll down (PF8) in your version of this panel to locate the correct option.

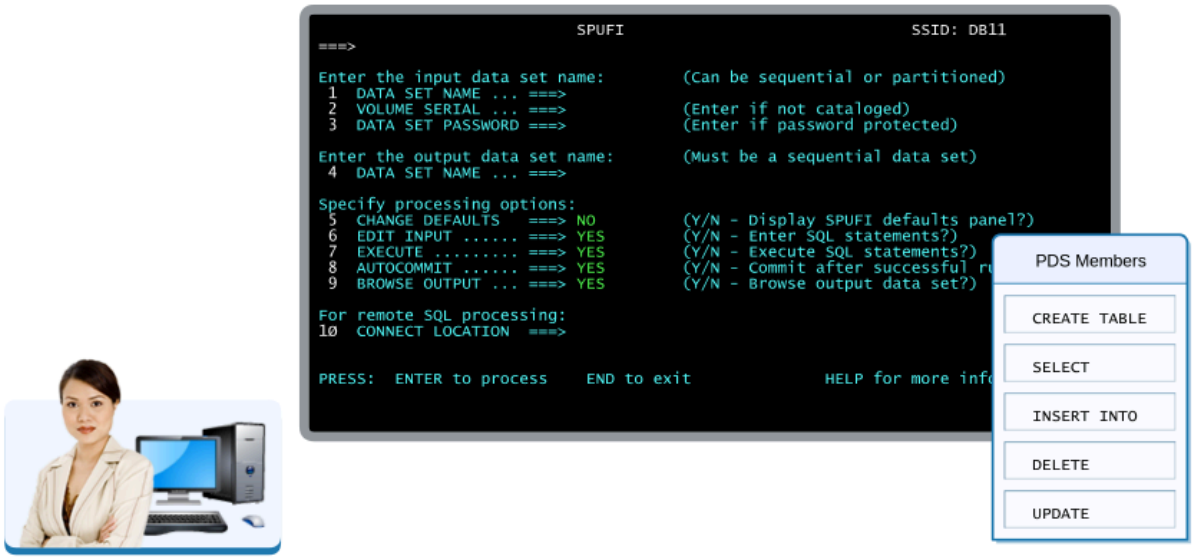
Type D2 in the Option line and **press Enter**.



Step 2 of 2

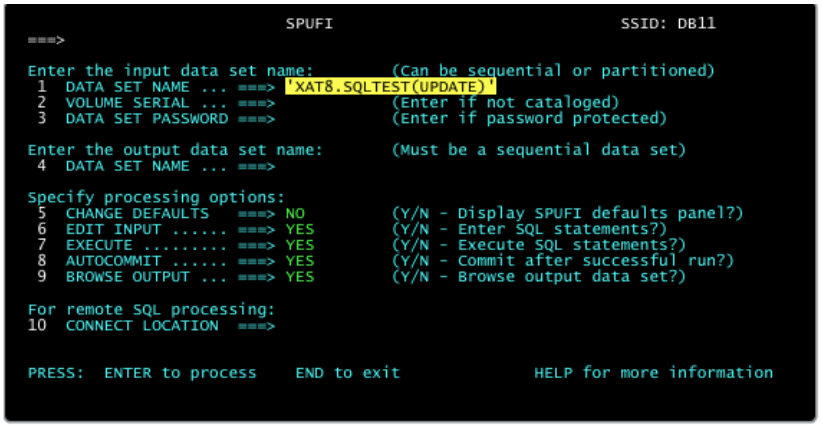
You can see that a number of interactive Db2 functions can be accessed from the resulting DB2I Primary Option Menu. The Db2 subsystem which will perform those functions is identified in the top right of this, and subsequent menu panels. In this case, it is DB11. The identity of the subsystem is just one of the variables which may be set via the panel option: D DB2I DEFAULTS.

Type 1 in the command line and **press Enter** so you can access SPUFI.



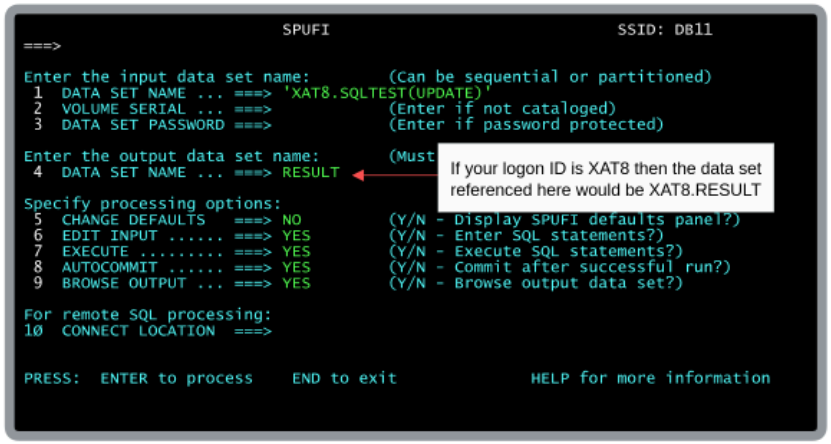
The SPUFI panel allows you to provide details on the input, output and general SPUFI defaults to be used when invoking your SQL statements against Db2 table data and indexes.

Firstly you should create a data set to store the SQL statements you need to test. This can be a sequential data set, but if you need to test different types of statements you may want to create a partitioned data set so that unique code can be stored in different members.



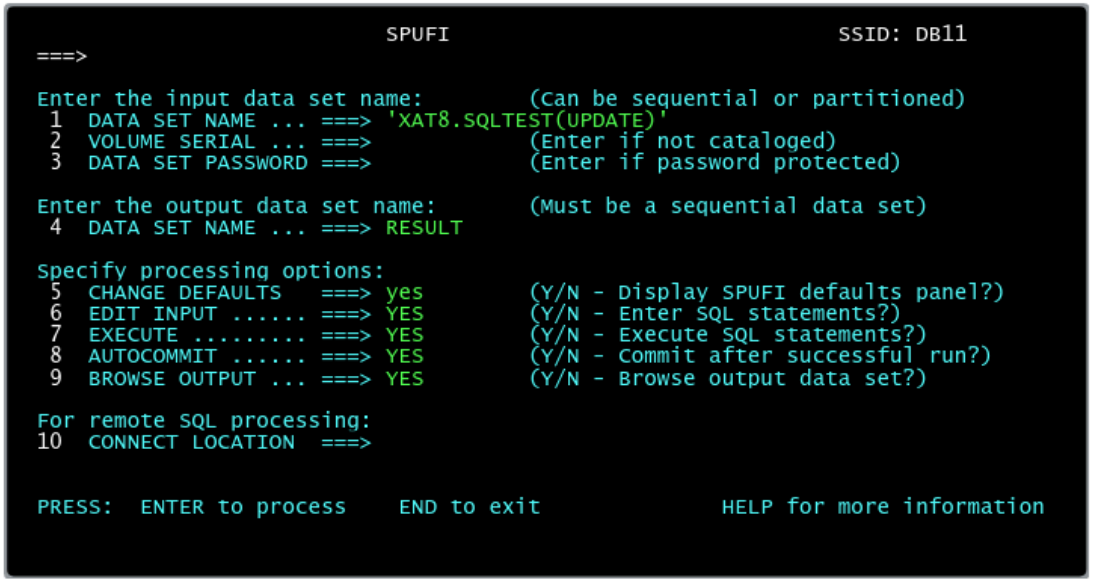
You will enter some SQL statements later but for now, you will code your SQL statements into a data set called XAT8.SQLTEST in a member called UPDATE.

This data set name and member needs to be coded within single quotes in the input data set name field. If the name is spelt incorrectly or you forget to include the single quotes, an error message will be displayed when SPUFI attempts to execute the SQL.



The result from your executed SQL statements will be stored in a sequential data set specified in option 4. If the data set name you provide is an existing one, data in it will be overwritten each time you use it. If the data set name does not exist, it will be created automatically for you.

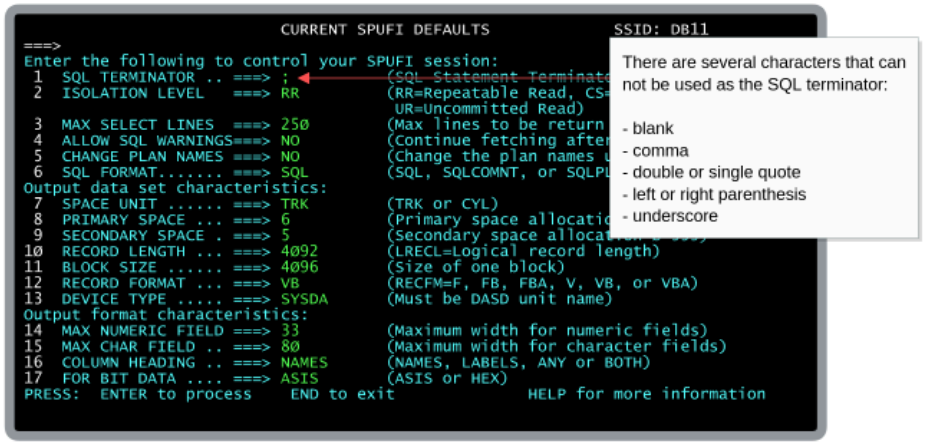
If you forget to insert the single quotes around the data set name, the data set referenced will contain your logon ID as the high level qualifier.



There may be cases where the default processing values used by SPUFI need to be changed before your SQL statements are processed.

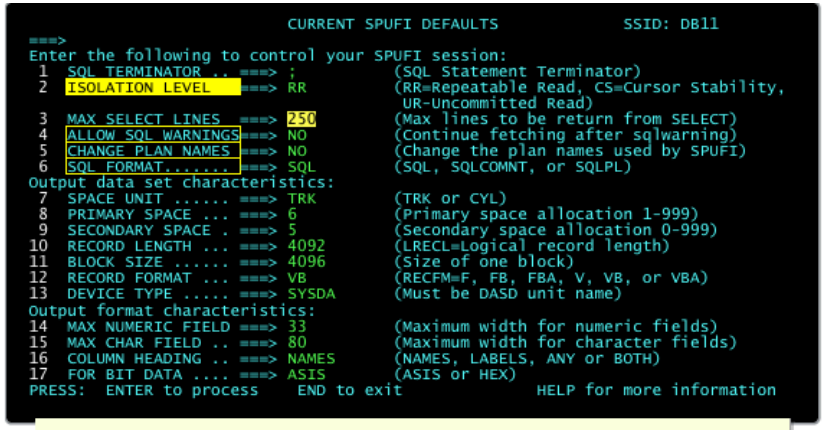
By setting the value for option 5 to Y or YES, the SPUFI default panel will be displayed before executing your SQL statements.

Type this value now and **press** **Enter** to display this panel.



The pages that follow contain details on the defaults you can overtype and update on the CURRENT SPUFI DEFAULTS panel.

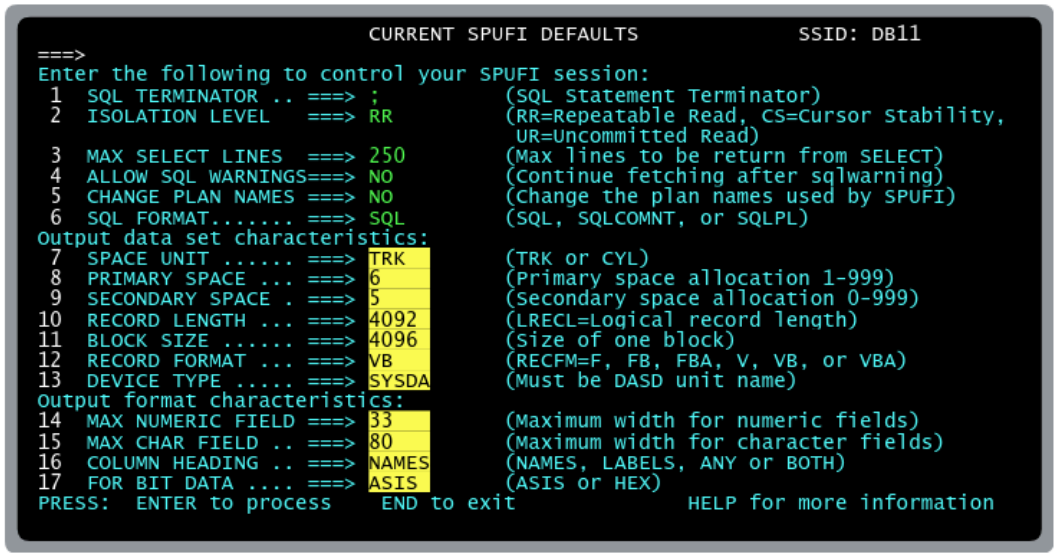
As you have seen in previous modules, the default SQL statement terminator is a semicolon. If your SQL statement does not contain this character as part of its data processing, then SPUFI will not know when to terminate the statement.



SQL statements can be executed at various isolation levels to prevent them from affecting other applications that are running. Refer to the IBM manual for additional information on this topic.

If you are using your SQL SELECT statements for testing purposes, you may want to limit the amount of data that is returned from it. The MAX SELECT LINES field can contain a number from 1 to 99999999 and will prevent any more lines than the number specified from being selected.

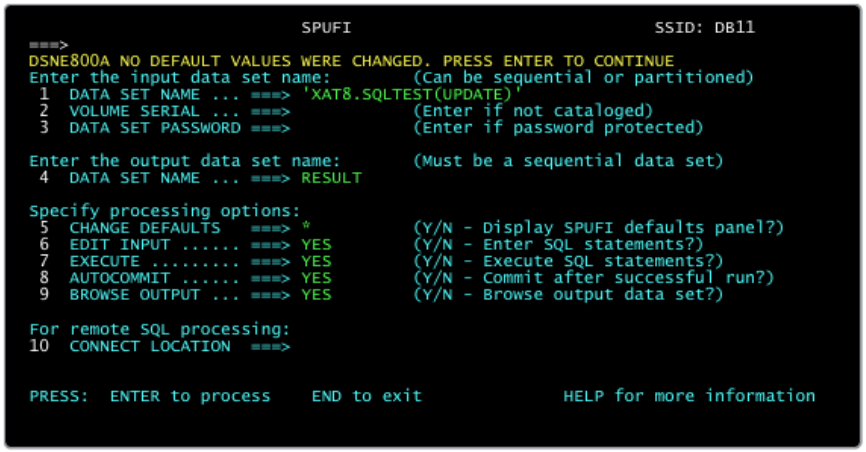
Mouse-over the other highlighted options in this section for a description of their purpose.



If the output produced from your SQL is to be used for other purposes such as being fed as input into another program, you may need to change some of the output data set characteristics or output format characteristics. Normally once all defaults have been defined in this panel, you will not need to access it every time you process your SQL so the CHANGE DEFAULTS option from the previous panel can be set to NO.

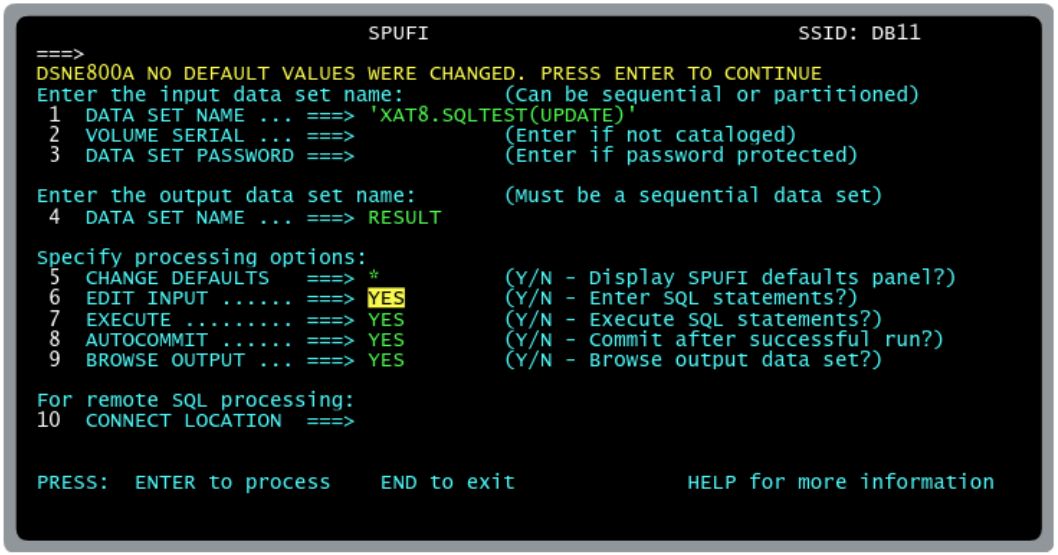
The F3 key can be used to exit from this panel and return to the main SPUFI panel, or if you have made any required changes, the Enter key can be pressed to continue processing of your SQL statements.

Press F3 to return to the SPUFI panel.



In this scenario there were no changes made to the defaults, and a message informing you of that fact is displayed at the top of the SPUFI panel.

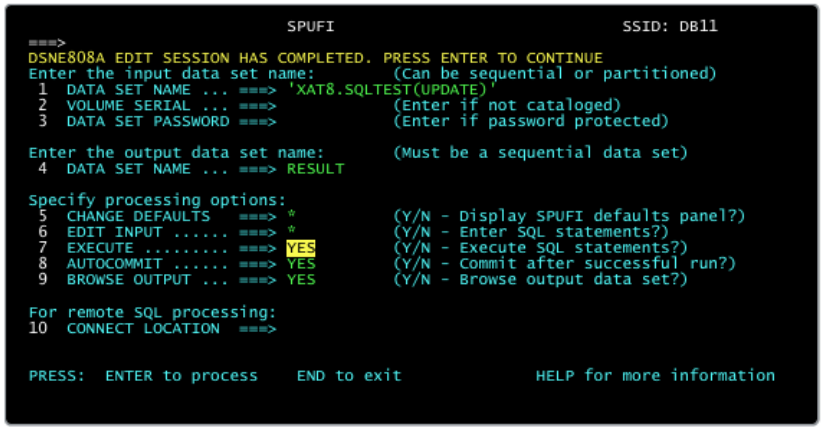
The CHANGE DEFAULTS field is also changed to * to indicate that it has been processed and that when the Enter key is pressed the defaults panel will not be displayed.



Step 1 of 2

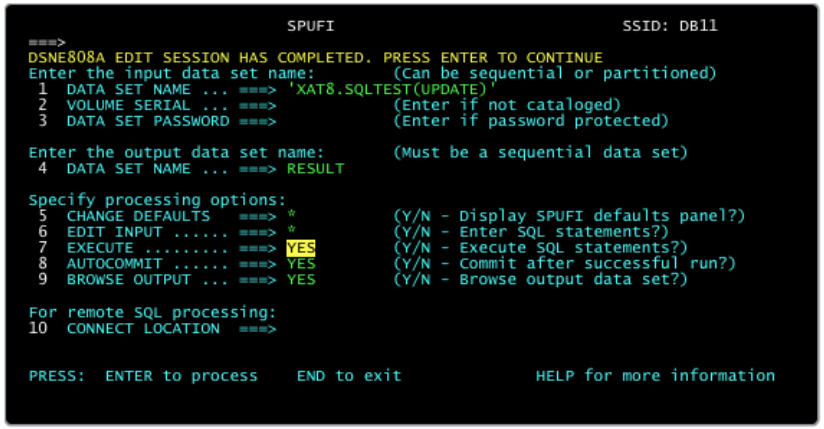
There are two methods of entering your SQL statements through SPUFI. You can create your input data set or data set member and using ISPF, code all your required statements prior to invoking them through SPUFI. Another method is to type YES into the EDIT INPUT field so that when SPUFI is invoked, the input data set will be opened using ISPF Edit and you can type in the statements required for that invocation.

This option is already set to YES on this panel so **press** **Enter** so you can see how SPUFI displays the input data.



The message at the top of the SPUFI panel indicates that the editing has been performed on the input data set. Normally you will want to begin processing the SQL statements that you have entered in your input data set, so the EXECUTE field should be set to Y or YES before pressing the Enter key.

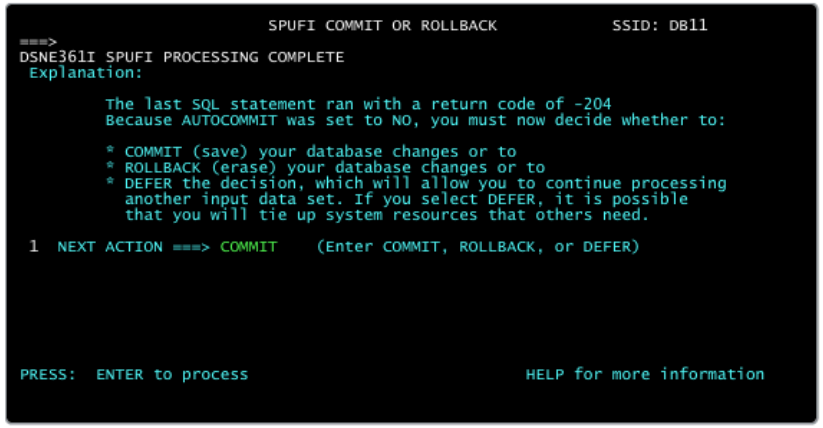
If this field is set to NO and BROWSE OUTPUT field is set to YES, then no SQL processing will be performed but any existing data in the output data set specified will be displayed.



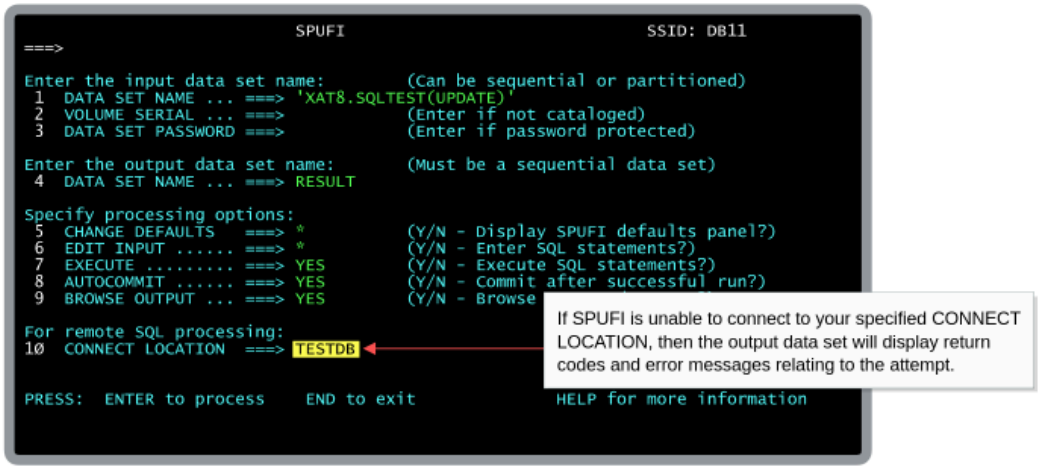
If the EXECUTE field is set to YES in your SPUFI panel, the next stage when you press Enter will see the execution of your SQL statements.

If there are many statements in your input data set or the table you are interrogating is large, it may take considerable time to process the statements. If this is unacceptable, you can interrupt the processing using the PA1 key with the amount of output created depending on how much processing the statement(s) were able to perform before being interrupted.

If SPUFI terminates while executing your statements it could indicate that Db2 resource limits have been defined by your Db2 administrator and that your SQL execution has exceeded them.



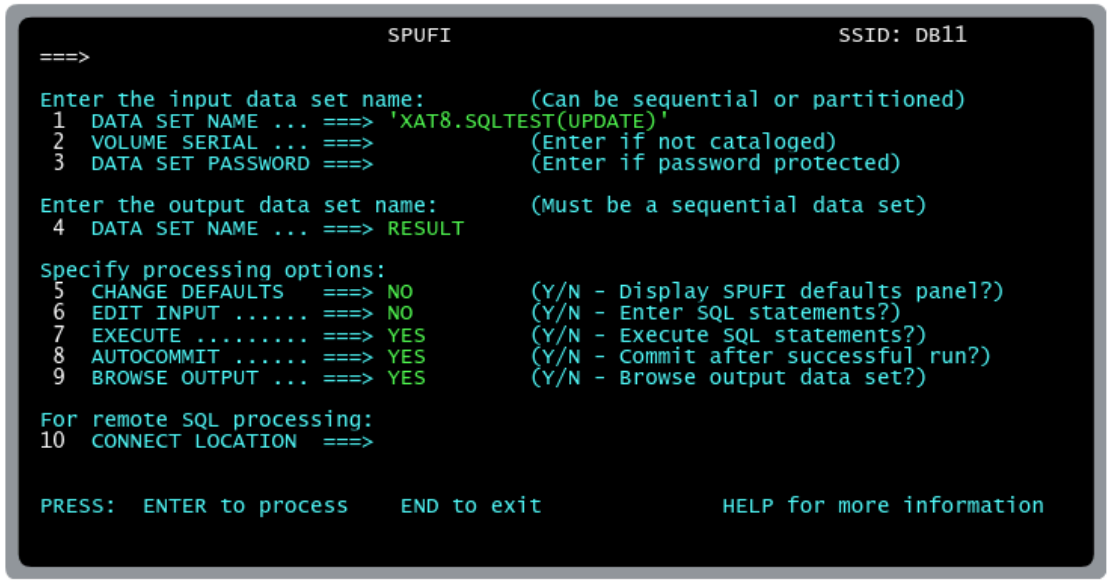
Normally when processing SQL statements you will want the results saved to the Db2 tables if they have executed successfully. If you are testing statements then you can be given the option to commit them to the database by specifying N or NO in the AUTOCOMMIT field on the SPUFI panel. This will result in the screen shown here being displayed.



If your SQL statements need to be processed on a remote database server, then the location of that server can be entered into the CONNECT LOCATION field.

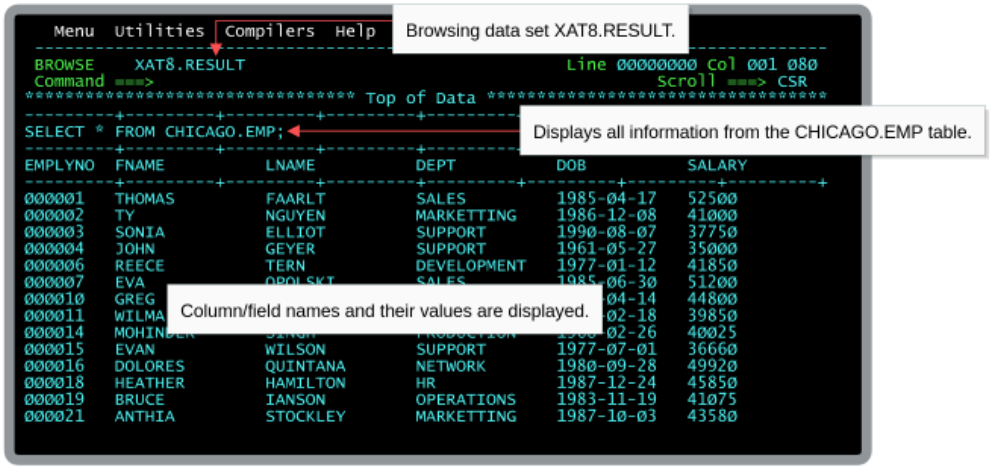
In this section you have seen how SPUFI is used to execute SQL statements and the defaults and processing options that can be invoked as part of this process.

In the next section you will look at the format of output produced by SPUFI and how to interpret messages displayed.



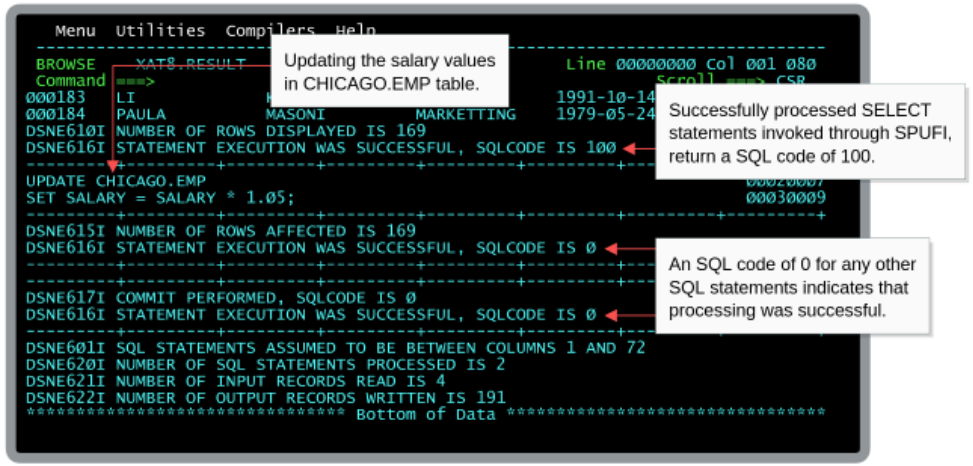
Now that you have identified the input and output data sets to be used by SPUFI you will look at interpreting the output produced from the execution of your SQL statements.

With the EXECUTE field set to YES, **press** **Enter** to execute the SQL statements in the XAT8.SQLTEST(UPDATE) data set.

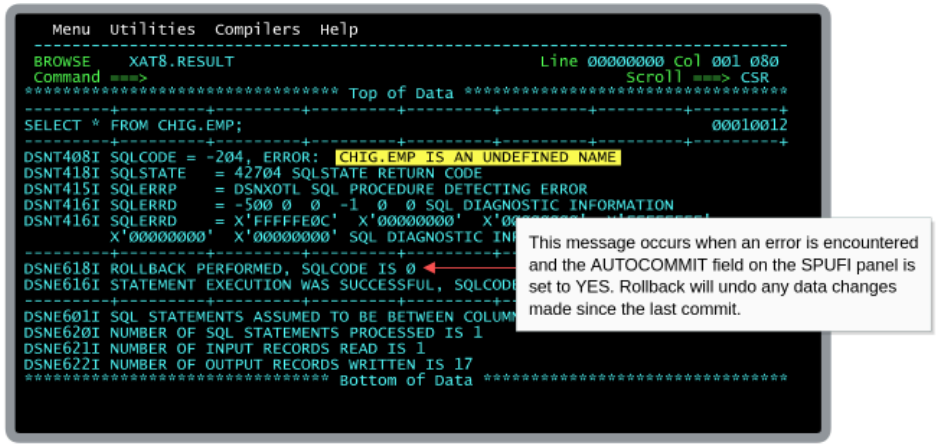


The ISPF Browse facility is opened and the result of your SQL execution is displayed. At the top of the output produced, the following data is displayed:

- A copy of the SQL statements from the input data set
- The data produced as a result of the processed SQL statements

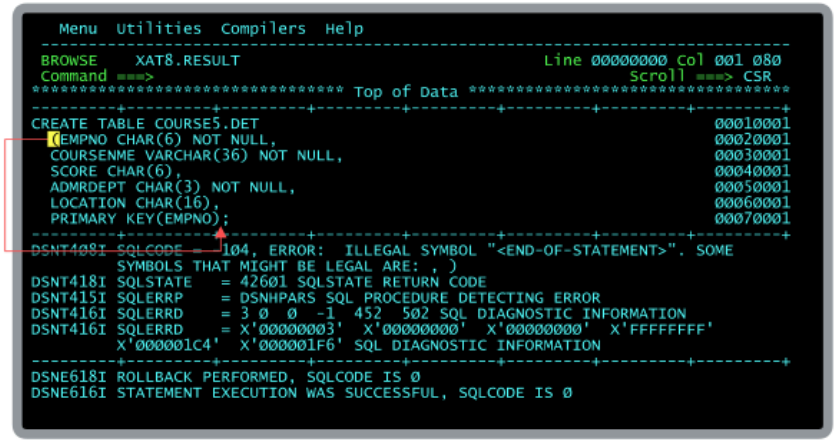


As you scroll towards the bottom of the data, you may encounter warning or error messages associated with the invocation, and will also see summary statistics.



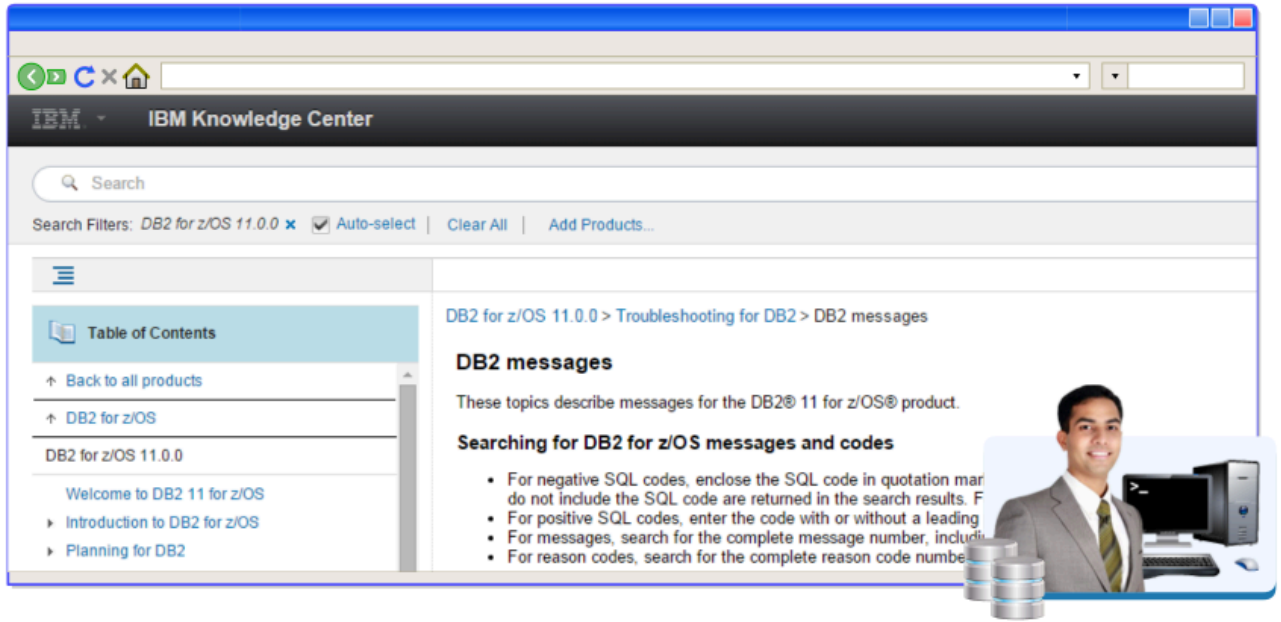
The example shown here highlights a message indicating that there is a problem. The message says that the table name specified in the SELECT statement is undefined, which indicates that a table of that name could not be located.

As a result of this error, rollback processing is performed.



In this example the messages indicate that there is an illegal symbol being used to terminate one of the SQL statements. In fact, the problem is associated with a missing corresponding closed parenthesis on the last line.

This highlights that at times it can be difficult to determine the actual cause of a problem, from the messages displayed.



Additional information on messages produced by Db2 can be obtained through IBM's Db2 for z/OS 11 Knowledge Center portal.

```
  File  Edit  Edit_Settings  Menu  Utilities  Compilers  Test  Help
-------------------------------------------------------------------------------
EDIT       XA8T.CNTL(XABSPUFI) - 01.03                  Columns 00001 00072
Command ===>                                              Scroll ===> CSR
****** ***************************** Top of Data ******************************
000100 //XABSPUFI JOB CLASS=A,MSGCLASS=X,REGION=64M
000200 //DSN     EXEC PGM=IKJEFT01,DYNAMNBR=20
000300 //STEPLIB  DD  DSN=DSN110.SDSNLOAD,DISP=SHR
000310 //SYSPRINT DD  SYSOUT=*
000400 //SYSTSPRT DD  SYSOUT=*
000500 //SYSTSIN  DD  *
000600  DSN SYSTEM(DB11)
000700  RUN  PROGRAM(DSNTEP2) PLAN(DSNTEP10) -
000800         LIB('DSN110.DB11.RUNLIB.LOAD') -
000810         PARMS('/ALIGN(LHS) MIXED TOLWARN(YES)')
000900  END
001000 //SYSIN    DD  *
001100 SELECT * FROM CHICAGO.EMP;
001200 /*
****** **************************** Bottom of Data ****************************
```

Before concluding this module on SPUFI, it should be mentioned that there is a batch equivalent that you may prefer to use if you have a large amount of SQL statements you need to invoke or there is a big Db2 table you need to access. Several batch programs are available that perform functions similar to those found in SPUFI, with DSNTEP2 and DSNTEP4 being the most popular.

An example of batch SPUFI is shown above.

Summary

SQL and SPUFI

In this module, you discovered how SPUFI can be used to invoke SQL statements interactively.

You should now be able to:

- Access SPUFI and Modify Panel Defaults
- Code and Execute SQL Statements from an Input Data Set Using SPUFI
- Interpret Output Produced from SPUFI

interskill learning

Db2 Fundamentals Mastery Test