PROCESS SQL PROCESS SQL

PROCESS SQL

PROCESS SQL ddm-name <<statement-string>>

This chapter covers the following topics:

- Function
- Syntax Description
- Entire Access Options
- Examples

For an explanation of the symbols used in the syntax diagram, see *Syntax Symbols*.

Belongs to Function Group: Database Access and Update

Function

The PROCESS SQL statement is used to issue SQL statements to the underlying database.

Syntax Description

Syntax Element	Description
ddm-name	The name of a DDM must be specified to provide the "address" of the database which executes the stored procedure. For more information see ddm-name.
statement-string	The statements which can be specified in the <code>statement-string</code> are the same statements which can be issued with the SQL statement <code>EXECUTE</code> (see also <code>Flexible SQL</code>).
	Warning: To avoid transaction synchronization problems between the Natural environment and the underlying database, the COMMIT and ROLLBACK statements must not be used within PROCESS SQL.
	The statement string can cover several statement lines without any continuation character to be specified. Comments at the end of a line as well as entire comment lines are possible.
	The statement string can also include parameters; see <i>Parameters</i> below.

Parameters

Γ	:ប	: host-variable [INDICATOR: host-variable] [LINIDICATOR: host-variable]
L	:G	

Unlike with the *parameter* described, in this context *host-variables* must be prefixed by a colon (:). In addition, they can be preceded by a further qualifier (:U or :G).

See further details on host-variable.

Syntax Element Description:

Syntax Element	Description
:U:host-variable	The prefix: U qualifies the host variable as a so-called "Using" variable. Such a variable indicates that its value is to be <i>passed to</i> the database. U is the default specification.
:G:host-variable	The prefix : G qualifies the host variable as a so-called "Giving" variable. Such a variable indicates that it is to <i>receive</i> a value <i>from</i> the database.

Entire Access Options

With Entire Access, you can also specify the following as *statement-string*:

- SET SQLOPTION option = value
- SQLCONNECT option = value
- SQLDISCONNECT

These options are only possible with Entire Access, and are described in the section *Accessing Data in an SQL Database* (in the *Programming Guide*).

Examples

Example for Adabas D:

PROCESS SQL ADABAS_D_DDM << LOCK TABLE EMPLOYEES IN SHARE MODE >>

Example of Calling a Procedure Stored in Adabas D:

The called procedure computes the sum of two numbers.

Examples PROCESS SQL

```
COMPUTE #N1 = 1
COMPUTE #N2 = 2
COMPUTE #SUM = 0
...

PROCESS SQL ADABAS_D_DDM << DBPROCEDURE DEMO.SUM (:#N1, :#N2, :G:#SUM) >>
...

WRITE #N1 '+' #N2 ' =' #SUM
...
```