Statements Grouped by Functions

This chapter provides an overview of the statements grouped by their functions.

This chapter covers the following topics:

- Database Access and Update

- Arithmetic and Data Movement Operations

- Loop Execution

- Creation of Output Reports

- Screen Generation for Interactive Processing

- Processing of Logical Conditions

- Invoking Programs and Routines

- User-Defined Functions

- Program and Session Termination

- Control of Work Files / PC Files

- Component Based Programming

- Memory Management Control for Dynamic Variables or X-Arrays

- Natural Remote Procedure Call

- Internet and XML

- Miscellaneous

- Reporting Mode Statements

Notes:

1. Certain statements can be used both in structured mode and in reporting mode, while others can be used in reporting mode only. See *Natural Programming Modes* in the *Programming Guide*.
2. The statements `DLOGOFF`, `DLOGON`, `SHOW`, `IMPORT` and `EXPORT` are only available when Entire DB is installed. For a description, see the *Entire DB* documentation.

Database Access and Update

Natural DML Statements

The following Natural Data Manipulation Language (DML) statements are used to access and manipulate information contained in a database.

READ	Reads a database file in physical or logical sequence of records.
FIND	Selects records from a database file based on user-specified criteria.
HISTOGRAM	Reads the values of a database field.
GET	Reads a record with a given ISN (internal sequence number) or RNO (record number).
GET SAME	Re-reads the record currently being processed.
ACCEPT/REJECT	Accepts/reject records based on user-specified criteria.
PASSW	Provides password for access to a password-protected file.
LIMIT	Limits the number of executions of a READ, FIND or HISTOGRAM processing loop.
STORE	Adds a new record to the database.
UPDATE	Updates a record in the database.
DELETE	Deletes a record from the database.
END TRANSACTION	Indicates the end of a logical transaction.
BACKOUT TRANSACTION	Backs out a partially completed logical transaction.
GET TRANSACTION DATA	Reads transaction data stored with a previous END TRANSACTION statement.
RETRY	Attempts to re-read a record which is in hold status for another user.
AT START OF DATA	Specifies statements to be performed when the first of a set of records is processed in a processing loop.
AT END OF DATA	Specifies statements to be performed after the last of a set of records has been processed in a processing loop.
AT BREAK	Specifies statements to be performed when the value of a control field changes (break processing).
BEFORE BREAK PROCESSING	Specifies statements to be performed before performing break processing.
PERFORM BREAK PROCESSING	Immediately invokes break processing.

Natural SQL Statements

In addition to the Natural DML Statements, Natural also provides SQL statements for use in Natural programs so that SQL can be used directly.

The following SQL Statements are available:

CALLDBPROC	Invokes a stored procedure of the SQL database system to which Natural is connected.
COMMIT	Indicates the end of a logical transaction and releases all data locked during the transaction. All data modifications are committed and made permanent.
DELETE	Deletes either rows in a table without using a cursor ("searched" DELETE) or rows in a table to which a cursor is positioned ("positioned" DELETE).
INSERT	Adds one or more new rows to a table.
PROCESS SQL	Issues SQL statements to the underlying database.
READ RESULT SET	Reads a result set which was created by a stored procedure that was invoked by a previous CALLDBPROC statement.
ROLLBACK	Undoes all database modifications made since the beginning of the last recovery unit.
SELECT	Supports both the cursor-oriented selection that is used to retrieve an arbitrary number of rows and the non-cursor selection (singleton SELECT) that retrieves at most one single row.
UPDATE	Performs an update operation on either rows in a table without using a cursor ("searched" UPDATE) or columns in a row to which a cursor is positioned ("positioned" UPDATE).

Arithmetic and Data Movement Operations

The following statements are used for arithmetic and data movement operations:

COMPUTE	Performs arithmetic operations or assigns values to fields.
ADD	Adds two or more operands.
SUBTRACT	Subtracts one or more operands from another operand.
MULTIPLY	Multiplies two or more operands.
DIVIDE	Divides one operand into another.
EXAMINE TRANSLATE	Translates the characters contained in a field into upper-case or lower-case, or into other characters.
MOVE	Moves the value of an operand to one or more fields.
MOVE ALL	Moves multiple occurrences of a value to another field.
COMPRESS	Concatenates the value of two or more fields into a single field.
SEPARATE	Separates the content of a field into two or more fields.
EXAMINE	Scans a field for a specific value and replaces it, and/or counts how often it occurs.
RESET	Sets the value of a field to zero (if numeric) or blank (if alphanumeric), or to its initial value.

Loop Execution

The following statements are related to the execution of processing loops:

ESCAPE	Stops the execution of a processing loop.
FOR	Initiates a processing loop and controls the number of times the loop is to be processed.
REPEAT	Initiates a processing loop (and terminates it based on a specified condition).
SORT	Sorts records.

Creation of Output Reports

The following statements are used for the creation of output reports:

FORMAT	Specifies output parameter settings.
DISPLAY	Specifies fields to be output in column form.
WRITE / PRINT	Specifies fields to be output in non-column form.
WRITE TITLE	Specifies text to be output at the top of each page of a report.
WRITE TRAILER	Specifies text to be output at the bottom of each page of a report.
AT TOP OF PAGE	Specifies processing to be performed when a new output page is started.
AT END OF PAGE	Specifies processing to be performed when the end of an output page is reached.
SKIP	Generates one or more blank lines in a report.
EJECT	Causes a page advance without titles or headings.
NEWPAGE	Causes a page advance with titles and headings.
SUSPEND IDENTICAL SUPPRESS	Suspends identical suppression for a single record.
DEFINE PRINTER	Allocates a report to a logical output destination.
CLOSE PRINTER	Closes a printer.

Screen Generation for Interactive Processing

The following statements are used to create data screens (maps) for the purpose of interactive processing of data:

INPUT	Creates a formatted screen (map) for data display/ entry.
REINPUT	Re-executes an INPUT statement (if invalid data were entered in response to the previous INPUT statement).
DEFINE WINDOW	Specifies the size, position and attributes of a window.
SET WINDOW	Activates and de-activates a window.
PROCESS PAGE	Creates a data mapping to a web rich GUI screen.
PROCESS PAGE USING	Performs rich GUI I/O processing using an adapter object generated from a page layout.
PROCESS PAGE UPDATE	Re-executes a PROCESS PAGE statement.
PROCESS PAGE MODAL	Initiates a processing block and controls the lifetime of a rich GUI window.

Processing of Logical Conditions

The following statements are used to control the execution of statements based on conditions detected during the execution of a Natural program:

IF	Performs statements depending on a logical condition.
IF SELECTION	Verifies that in a sequence of alphanumeric fields one and only one contains a value.
DECIDE FOR	Performs statements depending on logical conditions.
DECIDE ON	Performs statements depending on the contents of a variable.

Invoking Programs and Routines

The following statements are used in conjunction with the execution of programs and routines:

CALL	Invokes a non-Natural program from a Natural program.
CALLNAT	Invokes a Natural subprogram.
CALL FILE	Invokes a non-Natural program to read a record from a non-Adabas file.
CALL LOOP	Generates a processing loop containing a call to a non-Natural program.
DEFINE SUBROUTINE	Defines a Natural subroutine.
ESCAPE	Stops the execution of a routine.
FETCH	Invokes a Natural program.
PERFORM	Invokes a Natural subroutine.
PROCESS COMMAND	Invokes a command processor.
RUN	Compiles and executes a source program.

User-Defined Functions

The following Natural statements are used to create user-defined functions:

DEFINE FUNCTION	Creates new user-defined functions which may be called instead of operands in the Natural statements. Functions can be defined inside the object type Function only.
DEFINE PROTOTYPE	Specifies a signature according to a certain function call.
Function Call	Used to call user-defined functions which are defined inside special objects of type function. Also used to call Natural functions.

Program and Session Termination

The following Natural statements are used to terminate the execution of an application or to terminate the Natural session.

STOP	Terminates the execution of an application.
TERMINATE	Terminates the Natural session.

Control of Work Files / PC Files

The following Natural statements are used to read/write data to a physical sequential (non-Adabas) work file:

WRITE WORK FILE	Writes data to a work file.
DOWNLOAD PC FILE	Enables transfer data from a mainframe, UNIX or OpenVMS platform to the PC.
READ WORK FILE	Reads data from a work file.
UPLOAD PC FILE	Enables transfer data from a PC to a mainframe, UNIX or OpenVMS platform.
CLOSE WORK FILE	Closes a work file.
CLOSE PC FILE	Closes a specific PC work file.
DEFINE WORK FILE	Assigns a file name to a work file.

Component Based Programming

The following Natural statements are used in conjunction with component based programming:

`DEFINE CLASS`	Specifies a class from within a Natural class module.
`CREATE OBJECT`	Creates an object (also known as an instance) of a given class.
`SEND METHOD`	Invokes a method of an object.
`INTERFACE`	Defines an interface (a collection of methods and properties) for a certain feature of a class.
`METHOD`	Assigns a subprogram as the implementation of a method, outside an interface definition.
`PROPERTY`	Assigns an object data variable as the implementation to a property, outside an interface definition.

Memory Management Control for Dynamic Variables or X-Arrays

`EXPAND`	Expands the allocated memory of dynamic variables to a given size or expands the number of occurrences of X-arrays.
`REDUCE`	Reduces the size of a dynamic variable or the number of occurrences of X-arrays.
`RESIZE`	Adjusts the size of a dynamic variableor the number of occurrences of X-arrays.

Natural Remote Procedure Call

`OPEN CONVERSATION`	Allows the RPC Client to open a conversation and specify the remote subprograms to be included in the conversation.
`CLOSE CONVERSATION`	Allows the client to close conversations. You can close the current conversation, another open conversation, or all open conversations.
`DEFINE DATA CONTEXT`	Defines variables known as context variables, which are meant to be available to multiple remote subprograms within one conversation, without having to explicitly pass the variables as parameters with the corresponding `CALLNAT` statements.

See also the section *Natural Statements Involved* in the *Natural Remote Procedure Call* documentation.

Internet and XML

`PARSE`	Allows you to parse XML documents from a Natural program.
`REQUEST DOCUMENT`	Allows you to access an external system.

Miscellaneous

`DEFINE DATA`	Defines the data elements which are to be used in a Natural program or routine.
`END`	Indicates the end of the source code of a Natural program or routine.
`INCLUDE`	Incorporates Natural copycode at compilation.
`ON ERROR`	Intercepts runtime errors which would otherwise result in a Natural error message, followed by the termination of the Natural program.
`RELEASE`	Deletes the contents of the Natural stack; releases sets of ISN sets retained via a `FIND` statement; releases Natural global variables.
`SET CONTROL`	Performs a Natural terminal command from within a Natural program.
`SET KEY`	Assigns functions to terminal keys.
`SET GLOBALS`	Sets values for session parameters.
`SET TIME`	Establishes a point-in-time reference for a `*TIMD` system variable.
`STACK`	Places data and/or commands into the Natural stack.

Reporting Mode Statements

The following statements are for reporting mode only:

`CLOSE LOOP`	Closes a processing loop.
`DO/DOEND`	Specify a group of statements to be executed based on a logical condition.
`OBTAIN`	Causes on or more fields to be read from a file.
`REDEFINE`	Redefines a field.

The following statements can be used both in structured mode and in reporting mode, however, the statement structure and, with some of them, the functionality is different:

AT START OF DATA	Specifies statements to be performed when the first of a set of records is processed in a processing loop.
AT END OF DATA	Specifies statements to be performed after the last of a set of records has been processed in a processing loop.
AT BREAK	Specifies statements to be performed when the value of a control field changes (break processing).
AT TOP OF PAGE	Specifies processing to be performed when a new output page is started.
AT END OF PAGE	Specifies processing to be performed when the end of an output page is reached.
BEFORE BREAK PROCESSING	Specifies statements to be performed before performing break processing.
CALL LOOP	Generates a processing loop containing a call to a non-Natural program.
CALL FILE	Invokes a non-Natural program to read a record from a non-Adabas file.
COMPUTE	Performs arithmetic operations or assigns values to fields.
DEFINE SUBROUTINE	Defines a Natural subroutine.
ESCAPE	Stops the execution of a processing loop.
FIND	Selects records from a database file based on user-specified criteria.
GET SAME	Re-reads the record currently being processed.
HISTOGRAM	Reads the values of a database field.
IF	Performs statements depending on a logical condition.
IF SELECTION	Verifies that in a sequence of alphanumeric fields one and only one contains a value.
ON ERROR	Intercepts runtime errors which would otherwise result in a Natural error message, followed by the termination of the Natural program.
READ	Reads a database file in physical or logical sequence of records.
READ WORK FILE	Reads data from a work file.
REPEAT	Initiates a processing loop (and terminates it based on a specified condition).
SORT	Sorts records.
STORE	Adds a new record to the database.
UPDATE	Updates a record in the database.
UPLOAD PC FILE	Enables transfer data from a PC to a mainframe, UNIX or OpenVMS platform.