

# Programming Language Considerations

This section covers the following topics:

- Natural versus Third Generation Languages
  - Natural Statements with DL/I
  - Natural System Variables with DL/I
- 

## Natural versus Third Generation Languages

With a few exceptions Natural provides all of the functionality of third generation language programming in the DL/I environment.

However, accessing DL/I data using Natural is significantly different from programming techniques used in a third generation language. Natural application programmers do not have to code specific DL/I calls or build the segment search arguments (SSAs). They do not need to concern themselves with PCB mask information or keep track of PCB positioning between Syncpoints.

Natural for DL/I operates as a standard DL/I application and although most of the DL/I call processing is done internally, it is important to realize that all of the required DL/I processing is still performed:

PSBs are scheduled and terminated, PCBs are selected for use, database positioning is maintained, SSAs are created, the most efficient DL/I calls are issued, PCB mask information is evaluated, GET HOLD calls are issued before update or delete operations.

These tasks are all being performed for the application by Natural.

It is important to note that Natural is performing these tasks based on the information available in the application program. If, for example, a READ or FIND statement in a program is lacking essential segment search information, Natural selects a PCB, builds an SSA and issues a certain DL/I call based on this lacking information.

The Natural programmers use the same Natural statements to manipulate data in DL/I as they would for VSAM, Adabas or DB2.

Natural accesses DL/I segments based on the Natural DDM which is being referenced. Since the data access is always for one specific segment type (the one defined by the DDM), Natural does not issue path calls nor unqualified calls; that is, calls where the segment name is not specified.

### Notes:

1. Due to the structure of the Natural programming language, application control over DL/I call command codes is not available.
2. The LOG, STAT and GSCD call functions are not supported for the IMS TM environment.

## Natural Statements with DL/I

- BACKOUT TRANSACTION
- DELETE
- DISPLAY
- END TRANSACTION
- FIND
- GET TRANSACTION DATA
- READ
- RELEASE
- STORE
- UPDATE
- WRITE
- Statements not Available for DL/I

This section mainly consists of information also contained in the Natural Statements documentation, where each Natural statement is described in detail, including notes on DL/I usage where applicable. Summarized below are the particular points a programmer has to bear in mind when using Natural statements with DL/I.

Any Natural statement not mentioned in this section can be used without restrictions with DL/I.

### BACKOUT TRANSACTION

This statement is used to back out all database updates performed during the current logical transaction.

How the statement is translated and which command is actually issued depends on the TP-monitor environment:

- Under CICS, the BACKOUT TRANSACTION statement is translated into an EXEC CICS ROLLBACK command. However, in pseudo-conversational mode (PSEUDO=ON), only changes made to the database since the last terminal I/O are undone. This is due to CICS-specific transaction processing.
- In batch mode and under IMS TM, Natural for DL/I issues ROLB calls without checking the CMPAT setting in the corresponding NSB. However, under IMS TM, only changes made to the database since the last terminal I/O are undone. This is due to IMS TM-specific transaction processing.

Because PSB scheduling is terminated by a Syncpoint/checkpoint request, Natural saves the PCB position before executing the BACKOUT TRANSACTION statement. Before the next command execution, Natural reschedules the PSB and tries to set the PCB position as it was before the backout.

**Note:**

The PCB position might be shifted forward if any pointed segment had been deleted in the time period between the BACKOUT TRANSACTION and the following statement.

**DELETE**

The DELETE statement is used to delete a segment from a DL/I database, which also deletes all descendants of the segment.

**DISPLAY**

The DL/I AIX fields can be displayed only if a PCB is used with the AIX specified in the parameter PROCSEQ. If not, an error message is returned by Natural for DL/I at runtime.

**END TRANSACTION**

This statement indicates the end of a logical transaction and releases all DL/I data locked during the transaction. All data modifications are committed and made permanent.

How the statement is translated and which command is actually issued depends on the TP-monitor environment:

- Under CICS, the END TRANSACTION statement is translated into an EXEC CICS SYNCPOINT command.
- In batch mode and non message-driven BMP environments, Natural for DL/I issues CHKP calls without checking the CMPAT setting in the corresponding NSB.
- In MPP and message-driven BMP environments, the END TRANSACTION statement is not translated into a CHKP call, but is ignored, because CHKP calls imply GU calls. As Natural is a conversational transaction, you must reply to the terminal before requesting the next message (that is, before issuing the next GU call). An implicit end-of-transaction is issued after each terminal I/O.

Because PSB scheduling is terminated by a Syncpoint/checkpoint request, Natural saves the PCB position before executing the END TRANSACTION statement. Before the next command execution, Natural reschedules the PSB and tries to set the PCB position as it was before the END TRANSACTION statement.

**Note:**

The PCB position might be shifted forward if any pointed segment had been deleted in the time period between the END TRANSACTION and the following command.

With batch-oriented BMP regions, user data can be checkpointed on the IMS system log data sets. This is done by supplying an 8-byte checkpoint ID as the first operand in the END TRANSACTION statement, and by specifying the areas to be checkpointed as additional operands.

If the database specified with the profile parameter ETDB (as described in the Natural Parameter Reference documentation) is not the same as the database affected by the transaction, the first operand of the END TRANSACTION statement will be used as checkpoint ID for the ETDB database, while "NATDLICK" will be used as checkpoint ID for the other database *not* specified with the ETDB parameter.

The total area to be checkpointed must not exceed 1992 bytes; see also Symbolic Checkpoint/Restart Functions.

## FIND

With DL/I, the Natural FIND statement is typically used when a specific search criterion is known and specific segments are to be retrieved. This issues a DL/I GET UNIQUE call. However, if the FIND statement specifies a lower level segment and is within an active READ or FIND loop for an ancestor segment, it generally results in a DL/I GET NEXT WITHIN PARENT call.

The FIND statement initiates loop processing, which is active until all segment occurrences which match the search criterion have been read.

When accessing a field starting after the last byte of the given segment occurrence, the storage copy of this field is filled according to its format (numeric, blank, etc.).

FIND FIRST, FIND NUMBER and FIND UNIQUE are not permitted. The PASSWORD, CIPHER, COUPLED and RETAIN clauses are not permitted either.

In the WITH clause, you can only use descriptors that are defined as key fields in DL/I and marked with "D" in the DDM.

When connecting search criteria, the following has to be observed:

$[ \text{NOT} ] \left\{ \begin{array}{l} \textit{basic-search-criterion} \\ \textit{(search-expression)} \end{array} \right\} \left[ \left[ \begin{array}{l} \text{OR} \\ \text{NOT} \end{array} \right] \textit{search-expression} \right] \dots$
--

Connecting search criteria for segment type A results in multiple qualification statements within one DL/I segment search argument (SSA). Connecting search criteria for segment types A and B results in multiple SSAs. Therefore, the Boolean operator OR cannot be used to combine search criteria for different segment types.

## GET TRANSACTION DATA

This statement retrieves checkpoint data saved by an END TRANSACTION statement. The first parameter of this statement must be an 8-byte checkpoint ID. The remaining operands must be listed in the same sequence, length and format as in the corresponding END TRANSACTION statement; see also Symbolic Checkpoint/Restart Functions.

## READ

The READ statement should be used to process a set of segment occurrences in sequential order and usually results in a DL/I GET NEXT call.

When the READ statement is used, segments are retrieved based on the sequence field of the root segment or based on a secondary index field. Since the READ statement initiates sequential access of the database, it is important to understand that the EQUAL TO clause means the same thing as the STARTING FROM clause; it initiates a sequential read loop beginning with the key value specified.

The READ statement initiates loop processing. A loop is active until all segment occurrences which match the search criterion have been read.

The PASSWORD and CIPHER clauses are not permitted.

READ IN PHYSICAL SEQUENCE is used to read records in the order in which they are physically stored in a database. The physical sequence is the default sequence.

**Note:**

This is only valid when using Natural with HDAM databases.

READ BY ISN is not valid when using Natural with DL/I.

For Natural, the descriptor used must be either the sequence field of the root segment or a secondary index field. If a secondary index field is specified, it must also be specified in the PROCSEQ parameter of a PCB. Natural uses this PCB and the corresponding hierarchical structure to process the database.

## RELEASE

The RELEASE statement is not applicable for DL/I usage, since it releases sets of records retained by a FIND statement that contained a RETAIN clause, which is not valid when using Natural with DL/I.

## STORE

This statement can be used to add a segment occurrence.

If the segment occurrence is defined with a primary key, a value for the primary key field must be provided.

In the case of a GSAM database, records must be added at the end of the database (due to GSAM restrictions).

The USING/GIVING NUMBER clause is not valid when using Natural with DL/I.

If the SET/WITH clause is used, the following applies with Natural for DL/I:

- Values must be provided for the segment sequence field and for all sequence fields of the ancestors.
- Only I/O (sensitive) fields can be provided.
- A segment of variable length is stored with the minimum length necessary to contain all fields as specified with the STORE statement. The segment length will never be less than the minimum size specified in the SEGM macro of the DBD.
- If a multiple-value field or a periodic group is defined as variable in length, at the end of the segment only the occurrences as specified in the STORE statement are written to the segment and define the segment length.

## UPDATE

This statement can be used to update a segment in a DL/I database. The segment length is increased (if necessary) to accommodate all fields specified with the UPDATE statement. If a multiple-value field or a periodic group is defined as variable in length, only the occurrences as specified in the UPDATE

statement are written to the segment.

The DL/I AIX field name cannot be used in an UPDATE statement. AIX fields, however, can be updated by referring to the source field which comprises the AIX field.

DL/I sequence fields cannot be updated because of DL/I restrictions.

If the SET/WITH clause is used, only I/O (sensitive) fields can be provided. A segment sequence field cannot be updated (DELETE and STORE must be used instead).

Due to GSAM restrictions, the UPDATE statement cannot be used for GSAM databases.

## **WRITE**

With the WRITE statement, the DL/I AIX fields can be displayed only if a PCB is used with the AIX specified in the parameter PROCSEQ. If not, an error message is returned by Natural for DL/I at runtime.

## **Statements not Available for DL/I**

The following Natural statements are not available for DL/I users:

- GET
- GET SAME
- HISTOGRAM
- PASSW
- RELEASE

## **Natural System Variables with DL/I**

With DL/I, the following restrictions apply to the following Natural system variables:

### **\*ISN**

As there is no DL/I equivalent to Adabas ISNs, the system variable \*ISN is not available with Natural for DL/I.

### **\*NUMBER**

With Natural for DL/I, the Natural system variable \*NUMBER does not contain the number of segment occurrences found. It contains 0 if no segment occurrence satisfies the search criterion and a value of 8,388,607=X'7FFFFFF' if at least one segment occurrence satisfies the search criterion.