



NATURAL

Natural Planning Guide

Version 4.1 for Mainframes

Status: June 2002

Specifications contained herein are subject to change.
These changes will be reported in new editions.

 **SOFTWARE AG**



This document applies to Natural Version 4.1 for Mainframes. Specifications contained herein are subject to change and these changes will be reported in new editions.

© June 2002, Software AG
All rights reserved

Software AG and/or all Software AG products are either trademarks or registered trademarks of Software AG. Other products and company names mentioned herein may be the trademarks of their respective owners.

Table of Contents

| | |
|--|----|
| Natural Version 4.1 Planning Guide for Mainframes | 1 |
| Natural Version 4.1 Planning Guide for Mainframes | 1 |
| General Information | 2 |
| General Information | 2 |
| Documentation | 2 |
| Availability | 2 |
| Compatibility | 2 |
| Intented Incompatibility | 2 |
| Prerequisites | 3 |
| Other Software AG Products with Natural 4.1 | 3 |
| Examples | 3 |
| Programming Language | 4 |
| Programming Language | 4 |
| New Statements | 4 |
| EXPAND Statement | 4 |
| REDUCE Statement | 4 |
| Enhanced Statements | 4 |
| CALL Statement | 5 |
| CALLNAT Statement | 5 |
| DEFINE DATA Statement | 5 |
| DEFINE WORK FILE Statement | 5 |
| ESCAPE Statement | 5 |
| FIND Statement | 6 |
| HISTOGRAM Statement | 6 |
| INPUT Statement | 8 |
| INTERFACE Statement | 8 |
| METHOD Statement | 8 |
| PERFORM Statement | 8 |
| PROPERTY Statement | 8 |
| READ Statement | 8 |
| SEND METHOD Statement | 10 |
| Dynamic Variables | 10 |
| Optional Parameters | 10 |
| SPECIFIED Option in Logical Condition | 11 |
| MASK Option in Logical Condition | 11 |
| New System Variables | 11 |
| System Commands, Editors and Utilities | 13 |
| System Commands, Editors and Utilities | 13 |
| Enhanced System Commands | 13 |
| CATALL Command | 13 |
| COMPOPT Command | 13 |
| LAST Command | 14 |
| LIST Command | 14 |
| Removed System Commands | 14 |
| SYSBUS | 14 |
| New Utilities | 14 |
| SYSOBJH Utility | 14 |
| Enhanced Editors and Utilities | 14 |
| Data Area Editor | 15 |
| SYSBPM Utility | 15 |
| SYSMAIN Utility | 15 |
| SYSPARM Utility | 15 |
| SYSTP Utility | 15 |

| | |
|---|----|
| SYSUNLD Utility | 15 |
| Invoking of Utilities | 16 |
| Database Interfaces | 17 |
| Database Interfaces | 17 |
| Natural for Adabas | 17 |
| Natural for DB2 | 17 |
| Natural for DL/I | 17 |
| Natural for VSAM | 18 |
| XML Support | 19 |
| XML Support | 19 |
| Natural XML Toolkit | 19 |
| Mapping Natural Data Definitions to DTD | 19 |
| Serializing the Data to XML | 19 |
| Mapping DTD to Natural Data Definitions | 20 |
| Parsing the XML File and Assigning the Contents to Natural Data Structure | 20 |
| Miscellaneous Enhancements | 21 |
| Miscellaneous Enhancements | 21 |
| Support of New 64-Bit Architecture | 21 |
| Operating System and Teleprocessing Interfaces | 21 |
| Profile Parameters | 21 |
| New Profile Parameters | 21 |
| Enhanced Profile Parameters | 22 |
| Natural Performance Enhancements | 23 |
| Size of Alphanumeric and Binary Variables | 24 |
| Size of Data Elements | 24 |
| Object Handles | 24 |
| Source-Change Inventory | 24 |
| Non-Sliding Year Window (YSLW Parameter) | 24 |
| The Stack | 24 |
| User Exit Libraries SYSEXT and SYSEXTTP | 25 |
| Improved Abend and Error Handling | 25 |
| Buffer and Storage Management | 25 |
| Reduction of Object Code | 25 |
| Performance of Add-On Products | 26 |
| Entire Connection | 26 |
| Change/Enhancement Requests | 26 |
| Natural Security | 28 |
| Natural Security | 28 |
| Administrator Services | 28 |
| Default Profiles | 28 |
| Two-Phase Maintenance and Activation of Security Profiles | 28 |
| Library Profiles | 29 |
| Statement and Command Restrictions | 29 |
| Private Libraries | 29 |
| Linking Users to Libraries | 29 |
| Copying Libraries | 29 |
| Copying, Renaming and Deleting Libraries | 29 |
| Utility Profiles | 30 |
| Links | 30 |
| Functional Security | 30 |
| Batch Mode | 30 |
| Transferring Security Data | 30 |
| Interface Subprograms | 30 |
| Natural Advanced Facilities | 31 |
| Natural Advanced Facilities | 31 |
| Spool File | 31 |

- New Functions 31
- CICS-Specific Enhancements 31
- IMS/TM-Specific Enhancements 31
- Spool File Options 32
- Server Options 32
- Report Display 32
- Printer Definitions 32
- CALLNAT Interfaces 32

Natural Version 4.1 Planning Guide for Mainframes

- Document version of June 2002 -

This Planning Guide provides information regarding the planned enhancements for Natural Version 4.1 on mainframe computers, as well as information concerning migration from Version 3.1 to 4.1.

The information in this Planning Guide is primarily intended as a planning aid for those persons responsible for Natural administration and implementation. The reader is assumed to have detailed knowledge of Natural.

In addition to providing the enhancements described in this Planning Guide, Natural Version 4.1 will also consolidate all modifications/enhancements provided with the various System Maintenance (SM) releases of Version 3.1.

The following topics are covered in this document:

- | | |
|--|---|
| ● General Information | Information on the documentation, availability, compatibility, prerequisites of Natural Version 4.1, other Software AG products with Natural 4.1, and examples. |
| ● Programming Language | Information on new and enhanced statements and system variables, as well as other enhancements to the Natural programming language. |
| ● System Commands, Editors and Utilities | Information on new and enhanced Natural system commands, editors and utilities. |
| ● Database Interfaces | Information on the Natural interfaces to Adabas, DB2, DL/I and VSAM. |
| ● XML Support | Information on Natural's support of XML processing. |
| ● Miscellaneous Enhancements | Information on all other enhancements and new features. |
| ● Natural Security | Information on enhancements to Natural Security. |
| ● Natural Advanced Facilities | Information on enhancements to Natural Advanced Facilities. |

General Information

This section covers the following topics:

- Documentation
 - Availability
 - Compatibility
 - Prerequisites
 - Other Software AG Products with Natural 4.1
 - Examples
-

Documentation

All enhancements described in this Planning Guide will be fully documented in the Natural Version 4.1 documentation set.

Availability

Natural Version 4.1 is scheduled to be available in the first quarter of 2003.

Compatibility

Applications that were created with Natural Versions 2.2, 2.3 or 3.1 can be executed with Version 4.1 without any adjustments to the programs or any conversion or migration procedure being required.

Version 4.1 will be the last Natural version to support applications created with Version 2.2.

Natural Version 4.1 on mainframe computers will be syntax-compatible with Version 4.1 of Natural on OpenVMS, UNIX and Windows.

Intended Incompatibility

The following intended incompatibility will be introduced with Natural 4.1:

Translation of System Library Output

With the profile/session parameter `TS=ON` or the compilation option `TSENABL=ON` (set with the `COMPOPT` system command), output from Natural system libraries is translated using a translation table, which may be necessary for locations with non-standard lower-case usage.

With Natural 3.1, this translation is not performed for each field written into the Natural page buffer, but for the entire contents of the page buffer immediately before being sent to the screen. This may in some cases lead to incorrect results: When the contents of the page buffer comes from different programs with different `TS/TSENABL` settings - that is, some parts of the page are to be translated, others are not - the last applicable setting will be used, and the page will be either translated entirely or not at all. It may also occur that the last page is output after the end of the program when the `TS=ON/TSENABL=ON` setting is no longer available to Natural and this last page will then not be translated.

These translation errors will be corrected with Natural 4.1: The translation will be performed individually for each field at the time when it is written into the page buffer, according to the current `TS/TSENABL` setting that applies to each field. Consequently, the resulting output may not be the same as with Version 3.1.

Prerequisites

Natural Version 4.1 will require the following versions of the following operating/teleprocessing systems:

- BS2000/OSD Version 2 (or above),
- OS/390 Version 2.10 (or above),
- Z/OS Version 1.1 (or above),
- VM/ESA Version 2.4 (or above),
- Z/VM Version 4.1 (or above),
- VSE/ESA Version 2.5 (or above),
- CICS/TS Version 1.3 (or above) for OS/390,
- CICS/VSE Version 2.3 for VSE/ESA,
- CICS/TS Version 1.1.1 (or above) for VSE/ESA,
- IMS/TM Version 6.1 (or above),
- TIAM (any version available with BS2000/OSD Version 2),
- UTM Version 4.0 (or above).

Other Software AG Products with Natural 4.1

To use the following Software AG products in conjunction with Natural Version 4.1, the following product versions (or above) are required:

- Adabas 7.1,
- Com-plete 6.1,
- Entire Connection 4.2,
- EntireX 6.1,
- Predict 4.2.

Examples

The example library SYSEXV will provide examples of the new features of Natural Version 4.1.

Programming Language

The following programming enhancements are provided with Natural 4.1:

- New Statements
 - Enhanced Statements
 - Dynamic Variables
 - Optional Parameters
 - SPECIFIED Option Logical Condition
 - MASK Option in Logical Condition
 - New System Variables
-

New Statements

The following new Natural statements will be available with Natural Version 4.1:

- EXPAND
- REDUCE

EXPAND Statement

The new statement EXPAND will be used to increase the allocated memory size of a dynamic variable.

In the statement, you specify the name of the variable and its desired size. If that size is smaller than the size currently allocated for that dynamic variable, the EXPAND statement will have no effect.

For further information, see Dynamic Variables.

REDUCE Statement

The new statement REDUCE will be used to reduce the allocated memory size of a dynamic variable.

In the statement, you specify the name of the variable and its desired size.

The allocated memory of the dynamic variable which is beyond the given size is released immediately when the statement is executed.

If the currently used size (as contained in the new system variable *LENGTH) of the dynamic variable is greater than the given size, *LENGTH is set to the given size and the content of the variable is truncated (but not modified). If the specified size is larger than the currently allocated size of the dynamic variable, the REDUCE statement will be ignored.

For further information, see Dynamic Variables.

Enhanced Statements

The following Natural statements will be enhanced with Natural Version 4.1:

- CALL
- CALLNAT
- DEFINE DATA
- DEFINE WORK FILE

- ESCAPE
- FIND
- HISTOGRAM
- INPUT
- INTERFACE
- METHOD
- PERFORM
- PROPERTY
- READ
- SEND METHOD

CALL Statement

The CALL statement will provide the following enhancements:

- The limit of 32KB for the maximum length per parameter will be removed.
- A new option, INTERFACE4, will provide for enhanced parameter descriptions. Also, with this option the number of parameters to be passed to the invoked non-Natural program (currently 40) will no longer be restricted.

CALLNAT Statement

The CALLNAT statement will provide the following enhancements:

- **Notation "nX"** - see Optional Parameters.
- **Parameter Transfer with Dynamic Variables** - see Dynamic Variables.

DEFINE DATA Statement

The DEFINE DATA statement will provide two new options to be specified in the *parameter-data-definition* of a DEFINE DATA PARAMETER statement:

| | |
|-----------------|--|
| DYNAMIC | If you define a parameter as DYNAMIC, its length will be determined at runtime. For further information, see Dynamic Variables. |
| OPTIONAL | By default, parameter is defined without OPTIONAL, which means that a value <i>must</i> be passed from the invoking object to the parameter. If you define a parameter as OPTIONAL, a value can - but need not - be passed from the invoking object to this parameter. For further information, see Optional Parameters. |

DEFINE WORK FILE Statement

The new option TYPE STREAM will allow you to specify that a work file is to be used in stream mode (instead of record-oriented mode).

ESCAPE Statement

The ESCAPE statement will provide the following enhancements:

- ESCAPE TOP REPOSITION
- ESCAPE OBJECT

ESCAPE TOP REPOSITION

This new option will allow you to dynamically reposition within a READ statement loop that is being executed, and restart the READ loop with another start value.

When an ESCAPE TOP REPOSITION statement is executed, Natural will immediately continue processing at the top of the active READ loop, using the current value of the search variable as new start value.

At the same time, ESCAPE TOP REPOSITION will reset the system variable *COUNTER to "0".

ESCAPE TOP REPOSITION can be specified within a READ statement loop accessing an Adabas, DL/I or VSAM database. The READ statement concerned must contain the option WITH REPOSITION.

ESCAPE OBJECT

This new option will allow you to stop an inline subroutine and continue processing with the programming object which has invoked the object containing the inline subroutine.

When used within a subroutine, the existing option ESCAPE ROUTINE causes processing to continue with the statement following the PERFORM statement that has invoked the subroutine. In the case of an *inline* subroutine this would be within the same programming object. If nested subroutines are used, that is, if the PERFORM statement is itself contained within another inline subroutine, it would take a lot of coding to leave the active programming object entirely.

The new option ESCAPE OBJECT, however, will not only stop the processing of the inline subroutine, but also of the programming object containing the inline subroutine; processing will then continue with the object invoking that programming object. This will be particularly useful when multiple nested inline subroutines are used, as a single ESCAPE OBJECT statement will suffice to leave the programming object altogether.

ESCAPE OBJECT will only be relevant in *inline* subroutines. In external subroutines, subprograms and invoked programs, its would have the same effect as ESCAPE ROUTINE.

As with ESCAPE ROUTINE, the IMMEDIATE option to suppress loop-end processing will also be available with ESCAPE OBJECT.

FIND Statement

The FIND statement will provide the following enhancement:

Multi-Fetch

Traditionally, Natural retrieves database records one by one. However, Adabas's Multi-Fetch functionality makes it possible to retrieve more than one database record per database access. To make use of this functionality, the FIND statement will provide a new MULTI-FETCH option. With this option, you will be able to specify the number of records to be retrieved per database access when the statement is executed. The MULTI-FETCH option will be available for accesses to Adabas databases only. For database updates, the MULTI-FETCH option cannot be used.

MULTI-FETCH only affects the way in which the records are retrieved from the database. The program's record-processing logic will not be affected; that is, the number of FIND processing loops executed will be the same as without MULTI-FETCH, and the records will still be processed one by one.

HISTOGRAM Statement

The HISTOGRAM statement will provide the following enhancements:

- Dynamic Change of Reading Direction
- New Comparators
- Multi-Fetch
- ENDING AT Controlled by Database

Dynamic Change of Reading Direction

With Natural 3.1, the database field values to be retrieved by a HISTOGRAM statement can be read in ascending or descending sequence. This is determined by the keywords ASCENDING and DESCENDING in the SEQUENCE clause. Also, the VARIABLE option allows you to determine the reading direction at runtime. However, once the HISTOGRAM statement is executed, you cannot change the reading direction.

With Natural 4.1, the new keyword DYNAMIC will be provided for the SEQUENCE clause: It will allow you to change the reading direction from ascending to descending (or vice versa) within an active HISTOGRAM processing loop that is being executed, without having to restart the loop. After the keyword DYNAMIC, you will specify a variable to which the values "A" (for "ascending") or "D" (for "descending") can be assigned. The DYNAMIC option will be available for accesses to Adabas and DB2 databases.

New Comparators

In addition to the comparators EQUAL TO, STARTING FROM and ENDING AT, Natural 4.1 will provide the possibility to specify start/end values with the following options:

- LESS THAN
- GREATER THAN
- LESS EQUAL
- GREATER EQUAL

These new comparators will be available for accesses to Adabas, DB2, DL/I and VSAM databases.

Multi-Fetch

Traditionally, Natural retrieves database records one by one. However, Adabas's Multi-Fetch functionality makes it possible to retrieve more than one database record per database access. To make use of this functionality, the HISTOGRAM statement will provide a new MULTI-FETCH option. With this option, you will be able to specify the number of records to be retrieved per database access when the statement is executed. The MULTI-FETCH option will be available for accesses to Adabas databases only.

MULTI-FETCH only affects the way in which the records are retrieved from the database. The program's record-processing logic will not be affected; that is, the number of HISTOGRAM processing loops executed will be the same as without MULTI-FETCH, and the records will still be processed one by one.

ENDING AT Controlled by Database

With Natural 3.1, if the ENDING AT clause is used to limit the range of values to be read, Natural internally reads one value beyond the specified ENDING AT value in order to determine the end of the range to be read. This has been necessary due to restrictions inherent in the underlying databases.

With Natural 4.1, these restrictions no longer apply, and the ENDING AT value can now be determined by the accessed databases themselves. This means that Natural will be able to read the values only until including the specified ENDING AT value, but not beyond.

As this may lead to different results and so as not confuse the "old" end-value mechanism with the "new" one, a new keyword, TO, will be provided for the specification of the database-controlled end value. The existing ENDING AT clause will not be affected and will continue to yield the same results as before.

The new keyword TO will be available for Adabas, DB2, DL/I and VSAM databases.

INPUT Statement

The INPUT statement will provide the following enhancement:

Selection Boxes

Natural 4.1 will provide the possibility to attach selection boxes to input fields. These selection boxes are similar to those used in graphical user interfaces and are a comfortable alternative to help routines attached to fields.

To assign a selection box to a field, the INPUT statement will provide the new field attribute SB. With SB, you specify the contents of the selection box, that is, the values, or the name of an array field that provides the values, to be displayed within the selection box. The size and position of the selection box will be determined automatically (using the same algorithm as for help windows).

For a field for which the field attribute SB is specified, a selection-box indicator "V" will be displayed next to the field. To invoke the selection box, the user positions the cursor on the "V" and presses the help key. The selection box will then be displayed as a window on the screen. If the list of values within the selection box is longer than the selection box itself, the user can scroll by placing the cursor on the "More/Top/Bottom" lines of the selection box and pressing ENTER. To select a value from the selection box, the user positions the cursor on the desired value and presses ENTER. The selected value will then be copied into the input field.

The field attribute SB will only be available for alphanumeric fields.

INTERFACE Statement

The new option EXTERNAL will allow you to declare a NaturalX interface definition to be external.

METHOD Statement

The new ID clause will allow you to specify a dispatch ID for a NaturalX interface definition.

PERFORM Statement

The PERFORM statement will provide the following enhancements:

- **Notation "nX"** - see Optional Parameters.
- **Parameter Transfer with Dynamic Variables** - see Dynamic Variables.

PROPERTY Statement

The new ID clause will allow you to specify a dispatch ID for a NaturalX interface definition.

READ Statement

The READ statement will provide the following enhancements:

- Dynamic Change of Reading Direction
- New Comparators
- Multi-Fetch
- ENDING AT Controlled by Database
- WITH REPOSITION for Non-VSAM Databases

Dynamic Change of Reading Direction

With Natural 3.1, the records to be retrieved by a READ statement can be read in ascending or descending sequence. This is determined by the keywords ASCENDING and DESCENDING in the *sequence/range-specification*. Also, the VARIABLE option allows you to determine the reading direction at runtime. However, once the READ statement is executed, you cannot change the reading direction.

With Natural 4.1, the new keyword DYNAMIC will be provided for the *sequence/range-specification*: It will allow you to change the reading direction from ascending to descending (or vice versa) within an active READ processing loop that is being executed, without having to restart the loop. After the keyword DYNAMIC, you will specify a variable to which the values "A" (for "ascending") or "D" (for "descending") can be assigned. The DYNAMIC option will be available for accesses to Adabas and DB2 databases.

New Comparators

In addition to the field/value comparators EQUAL TO, STARTING FROM and ENDING AT, Natural 4.1. will provide the possibility to specify start/end values with the following options:

- LESS THAN
- GREATER THAN
- LESS EQUAL
- GREATER EQUAL

These new comparators will be available for accesses to Adabas, DB2, DL/I and VSAM databases.

Multi-Fetch

Traditionally, Natural retrieves database records one by one. However, Adabas's Multi-Fetch functionality makes it possible to retrieve more than one database record per database access. To make use of this functionality, the READ statement will provide a new MULTI-FETCH option. With this option, you will be able to specify the number of records to be retrieved per database access when the statement is executed. The MULTI-FETCH option will be available for accesses to Adabas databases only. For database updates, the MULTI-FETCH option cannot be used.

MULTI-FETCH only affects the way in which the records are retrieved from the database. The program's record-processing logic will not be affected; that is, the number of READ processing loops executed will be the same as without MULTI-FETCH, and the records will still be processed one by one.

ENDING AT Controlled by Database

With Natural 3.1, if the ENDING AT clause is used to limit the range of values to be read, Natural internally reads one value beyond the specified ENDING AT value in order to determine the end of the range to be read. This has been necessary due to restrictions inherent in the underlying databases.

With Natural 4.1, these restrictions no longer apply, and the ENDING AT value can now be determined by the accessed databases themselves. This means that Natural will be able to read the values only until including the specified ENDING AT value, but not beyond.

As this may lead to different results and so as not confuse the "old" end-value mechanism with the "new" one, a new keyword, TO, will be provided for the specification of the database-controlled end value. The existing ENDING AT clause will not be affected and will continue to yield the same results as before.

The new keyword TO will be available for Adabas, DB2, DL/I and VSAM databases.

WITH REPOSITION for Non-VSAM Databases

Due to the introduction of the new ESCAPE statement option TOP REPOSITION, the WITH REPOSITION option of the READ statement will no longer be restricted to VSAM databases, but will also be available for Adabas and DL/I databases.

SEND METHOD Statement

The SEND METHOD statement will provide the following enhancement:

- **Notation "nX"** - see Optional Parameters.

Dynamic Variables

In addition to removing the size limitations for alphanumeric and binary variables (see Size of Alphanumeric and Binary Variables), Natural Version 4.1 will make it possible to allocate the length of such variables dynamically at runtime.

As the maximum size of large data structures (for example, pictures, sounds, videos) may not exactly be known at the time an application is developed, Natural provides for the definition of alphanumeric and binary variables with the attribute DYNAMIC. The value space of variables which are defined with this attribute will be extended dynamically at runtime when it becomes necessary (for example, during an assignment operation: #picture1 := #picture2). This means that large binary and alphanumeric data structures may be processed in Natural without having to define a length at development time.

The new Natural system variable *LENGTH will be provided to obtain the value space (number of bytes) currently used for a given dynamic variable at runtime.

For performance optimization and also to avoid problems with too much or too little allocated memory space, the new statements EXPAND and REDUCE will be introduced. If the space allocated for a dynamic variable is no longer needed, the REDUCE statement can be used to reduce the allocated space (to zero or any other desired size). If the upper limit of memory usage is known for a specific dynamic variable, the EXPAND statement can be used to set the used space for the dynamic variable to this specific size.

Dynamic variables can be used, for example, in CALLNAT or PERFORM statements.

Optional Parameters

Natural Version 4.1 will support the use of optional parameters in subprograms, external subroutines and dialogs.

An optional parameter is a field defined with the keyword OPTIONAL in the DEFINE DATA PARAMETER statement of an invoked object (subprogram, external subroutine or dialog). To such a field, a value can - but need not - be passed from an invoking object.

In the invoking statement (CALLNAT, PERFORM or SEND METHOD), the notation *nX* is used to indicate optional parameters for which no values are passed. With *nX* you specify that the next *n* parameters are to be skipped; that is, for the next *n* parameters no values are passed to the invoked object.

For example:

| | |
|--|--|
| <p>Subprogram:</p> <pre> DEFINE DATA PARAMETER 1 #P1 (A10) 1 #P2 (A10) OPTIONAL 1 #P3 (A10) 1 #P4 (A10) OPTIONAL 1 #P5 (A10) OPTIONAL END-DEFINE ... </pre> | <p>Invoking Object:</p> <pre> CALLNAT 'MY-SUB' #A #B #C #D #E or CALLNAT 'MY-SUB' #A 1X #C 2X or CALLNAT 'MY-SUB' #A #B #C 1X #E </pre> |
|--|--|

To check in the invoked object whether or not an optional parameter has received a value from the invoking object, the new SPECIFIED option to be used in a logical condition will be available.

SPECIFIED Option in Logical Condition

With the new SPECIFIED option to be specified in a logical condition, you will be able to check whether or not an optional parameter in an invoked object (subprogram, external subroutine or dialog) has received a value from the invoking object.

If you process an optional parameter which has not received a value, this will cause a runtime error. To avoid such an error, you use the SPECIFIED option in the invoked object to check whether an optional parameter has received a value or not, and then only process it if it has.

For example:

```

IF #OPTFIELD1 SPECIFIED THEN ... ELSE ...
IF #OPTFIELD2 NOT SPECIFIED THEN ... ELSE ...
    
```

For a field not defined as OPTIONAL, the SPECIFIED condition will always be "TRUE".

MASK Option in Logical Condition

With Version 4.1, it will be possible to check positions of a field for a date in Julian format. This will be particularly useful when a MASK option is used in conjunction with a MOVE EDITED statement that uses a Julian date in its edit mask.

See also the COMPOPT system command for enhancements related to the MASK option.

New System Variables

The following new Natural system variables will be available with Natural Version 4.1:

| System Variable | Description |
|-------------------------|--|
| *CPU-TIME | Contains the CPU time used by the Natural process. |
| *DATV | Contains the current date in the format <i>dd-mon-yyyy</i> (where <i>mon</i> is the name of the month, abbreviated to 3 characters). |
| *DATVS | Contains the current date in the format <i>ddmonyyyy</i> (where <i>mon</i> is the name of the month, abbreviated to 3 characters). |
| *HOSTNAME | Contains the name of the machine on which Natural is running. |
| *LENGTH(<i>field</i>) | Contains the currently used length (in bytes) of a <i>field</i> defined as a dynamic variable. See also Dynamic Variables. |
| *NATVERS | Contains the Natural version number. |
| *PARM-USER | Contains the name of the parameter module currently in use. |
| *PATCH-LEVEL | Contains the Natural patch-level number. |
| *PID | Contains the current process ID. |

System Commands, Editors and Utilities

This section covers the following topics:

- Enhanced System Commands
 - Removed System Commands
 - New Utilities
 - Enhanced Editors and Utilities
 - Invoking of Utilities
-

Enhanced System Commands

The following Natural system commands will be enhanced for Natural Version 4.1:

- CATALL
- COMPOPT
- LAST
- LIST

CATALL Command

The CATALL command will provide the following enhancements:

- The error list provided by the CATALL command will be expanded to include additional information on the causes of CATALL processing errors.
- A user exit will be provided which will allow you to change the default settings of the main CATALL screen.

COMPOPT Command

The COMPOPT command will provide the following enhancements:

- With the MASK option in a logical condition, a valid year value must be in the range 0000 - 2699; with the MOVE EDITED statement, a valid year value must be in the range 1582 - 2699. Consequently, a year value found valid by a MASK option might lead to an error in a subsequent MOVE EDITED statement. To avoid this situation, the COMPOPT command will provide a new option, MASKCME, which will allow you determine whether the year range to be considered valid by the MASK option is to begin with 0000 or 1582.
- With Natural Version 2.3 (sic!), the internal handling of assignments between two format N variables of the *same* length was changed, so as to be consistent with the internal handling of assignments between format N variables of different lengths. One side effect of that change was that when these variables were redefined, this could in some cases lead to different results. With Version 4.1, the COMPOPT command will provide a new option, NMOVE22, which will allow you to determine whether the internal handling of assignments between format N variables of the same length is to be done inconsistently as in Version 2.2 or consistently as in all subsequent versions (the latter will be the default). The NMOVE22 option will replace the corresponding special-purpose zaps that were supplied for Versions 2.3 and 3.1.
- The option V22COMP (allow old Version 2.2 syntax), which was introduced with Natural Version 2.3, will continue to be available with Version 4.1.

LAST Command

"LAST *" displays a window showing the last 9 system commands that were issued. With Natural 4.1, this window will show the last 20 system commands.

LIST Command

The LIST command will provide the following enhancements:

- When the LIST command displays a list of objects, these object are in alphabetical order of their names. With Natural 4.1, it will also be possible to sort the list of objects by type, date, etc.
- In the list of objects, it will be possible to mark an object of type class with the command CR to register it, and with the command UR to unregister it.
- In the list of objects, it will be possible to mark an object of type class with a command to have its global unique ID (GUID) displayed.

Removed System Commands

SYSBUS

The Natural system command SYSBUS will no longer be available as of Natural Version 4.1. Instead, you use the system command BUS, which performs the same function.

New Utilities

The following new Natural utility will be provided with Natural Version 4.1:

SYSOBJH Utility

The new utility SYSOBJH (OBJect Handler) will process objects for the purpose of application distribution. This utility will combine the functionality currently provided by the utilities SYSTRANS and SYSUNLD. The utilities SYSTRANS and SYSUNLD will cease to be available with one of the next Natural releases.

SYSOBJH can be used to unload objects in the source environment to work files, and then load these objects from work files into the target environment. SYSOBJH can process Natural programming objects, resources, DDMs, error messages, Natural-related objects, Natural command processors, external objects, and Adabas FDTs.

Unloading and loading can be performed in internal format (as with the SYSUNLD utility) or in transfer format (as with the SYSTRANS utility).

Work files created with the utilities SYSTRANS and SYSUNLD can also be processed.

Work files created with SYSOBJH in transfer format can be processed by the utility SYSTRANS on all platforms.

Enhanced Editors and Utilities

The following enhancements will be provided with Natural 4.1:

- Data Area Editor
- SYSBPM
- SYSMAIN

- SYSPARM
- SYSTP
- SYSUNLD

Data Area Editor

The data area editor will provide the following enhancements:

- The number of possible field levels in a data area will be increased from 9 to 99.
- The maximum length of an alphanumeric variable in a data area will be increased from 253 bytes to 1 GB.
- The maximum length of a binary variable in a data area will be increased from 126 bytes to 1 GB.
- It will be possible to define object handles within a data area.

SYSBPM Utility

The SYSBPM utility will provide the following enhancements:

- It will be possible to display the contents of the buffer pool in batch mode.
- In batch mode, it will be possible to sort the objects in the buffer pool by various criteria (for example, use count).

SYSMAIN Utility

The SYSMAIN utility will provide the following enhancement:

- While the function Find Programming Objects is being executed, SYSMAIN displays a window indicating the name of the object being searched. With Natural Version 4.1, this window will also indicate whether the source form or cataloged form of an object is being searched.

SYSPARM Utility

The SYSPARM utility will provide the following enhancement:

- With Natural 3.1, you can only maintain parameter profiles stored on and applying to the current FNAT system file. With Natural 4.1, it will also be possible to maintain parameter profiles of other FNAT system files.

SYSTP Utility

The SYSTP utility will provide the following enhancements:

- A new SYSTP function will allow you to cancel Natural user sessions depending on their last-activity date.
- In addition to getting a list of all Natural user sessions, it will be possible to display only the sessions related to one specific user ID or terminal ID.
- It will be possible to control the use of SYSTP functions via Natural Security (in the same way as has been possible for various other Natural utilities).

SYSUNLD Utility

The SYSUNLD utility will provide the following enhancement:

- If objects are to be unloaded/loaded via Entire Connection and Entire Connection has not been activated, SYSUNLD will automatically issue the terminal command "%+" to activate Entire Connection.
- A user exit will be provided which can be invoked as soon as NATLOAD processing has finished.

Invoking of Utilities

At present, it is possible to invoke a utility either by entering the utility name as a system command (for example SYSMAIN); or by issuing a LOGON command to the library containing the utility (for example, LOGON SYSMAIN), followed by the command MENU.

As of Natural Version 4.1, the only possible way of invoking a utility will be by using the utility name as a system command. This is to provide for a more consistent Natural Security protection of the utilities.

Database Interfaces

This section covers the following topics:

- Natural for Adabas
 - Natural for DB2
 - Natural for DL/I
 - Natural for VSAM
-

Natural for Adabas

Natural Version 4.1 will require at least Adabas Version 7.1. Consequently, Adastar will no longer be supported.

The following Adabas-related enhancements will be provided with Natural Version 4.1:

- Dynamic change of reading direction within an active READ or HISTOGRAM processing loop.
- Dynamic repositioning within a READ processing loop.
- New comparators LESS/GREATER THAN and LESS/GREATER EQUAL for READ and HISTOGRAM statements.
- Support of Multi-Fetch in FIND, HISTOGRAM and READ statements.
- ENDING AT value in HISTOGRAM and READ statements controlled by the database itself.
- Universal Encoding Support (UES) to support unicode code pages.
- Support of the Adabas Transaction Manager.

Natural for DB2

The following DB2-related enhancements will be provided with Natural Version 4.1:

- The SQL syntax enhancements provided by DB2 Versions 6 and 7 will be supported.
- At present, you have to use Natural format A for SQL data types DATE and TIME. With Version 4.1, it will be possible to use Natural variables of formats D and T; these will automatically be converted to SQL data types DATE and TIME.
- Large objects (alphanumeric and binary variables of up to 1 GB) will be supported.
- In addition to the two existing DB2 parameter styles, the new parameter style "DB2SQL" for the passing of data to stored procedures will be supported.
- It will be possible to write "user-defined functions" in Natural.
- Support of insensitive and sensitive static scrollable cursors in the Natural SQL statement SELECT.
- New comparators LESS/GREATER THAN and LESS/GREATER EQUAL for READ and HISTOGRAM statements.
- ENDING AT value in HISTOGRAM and READ statements controlled by the database itself.
- Dynamic change of reading direction within an active READ or HISTOGRAM processing loop.

Natural for DL/I

The following DL/I-related enhancements will be provided with Natural Version 4.1:

- Dynamic repositioning within a READ processing loop.
- New comparators LESS/GREATER THAN and LESS/GREATER EQUAL for READ and HISTOGRAM statements.
- ENDING AT value in HISTOGRAM and READ statements controlled by the database itself.

Natural for VSAM

The following VSAM-related enhancements will be provided with Natural Version 4.1:

- Support of "Transactional VSAM".
- Dynamic repositioning within a READ processing loop.
- New comparators LESS/GREATER THAN and LESS/GREATER EQUAL for READ and HISTOGRAM statements.
- ENDING AT value in HISTOGRAM and READ statements controlled by the database itself.

XML Support

This section covers the following topic:

- Natural XML Toolkit
-

Natural XML Toolkit

The Natural XML Toolkit is a set of tools for XML processing which will provide functionality required for the integration of XML processing into Natural. It will improve the integration of Natural application with XML, without requiring external software products like "msxml".

The XML Toolkit can be considered an intermediate step towards the integration of XML processing into Natural. The next step would be the full integration of XML functionality in the Natural programming language.

The XML Toolkit consists of a set of Natural programs, some of which will be made available in source-code form. The XML Toolkit programs may be integrated into your Natural applications, thus providing access to XML data or supplying data from Natural in XML format.

The XML Toolkit will provide the following functions:

- Mapping of Natural data definitions to DTDs, and vice versa.
DTDs (document type descriptions) are most commonly used to describe the structure of an XML document.
- XML Token -> Natural Data
After creating the Natural data structure, the XML document has to be parsed and saved into the data structure. An implementation for the delivered XML "tokenizer" will be generated. This tokenizer assigns the value of a data element to the corresponding data structure.
- Natural Data -> XML Document ("Serialization")
Serialization is the process of taking the data stored in the Natural data structures and creating an XML document according to the description in the DTD.
- Check data structure for:
 - alternative elements,
 - range of values for attributes,
 - occurrences and boundaries of elements.

Mapping Natural Data Definitions to DTD

The mapping of Natural data definitions to a DTD is the first step to bind Natural data structures to XML tags, and is required to implement a representation of Natural data as XML tags. The generated DTD will be used later during the serialization to an XML document.

Serializing the Data to XML

During the execution of a Natural program, the fields defined in the DEFINE DATA statement will be filled with real contents. During serialization, this contents will be written to a dynamic variable in XML format using the previously generated DTD as input. The XML Toolkit will also generate the program for the serialization of the data.

Mapping DTD to Natural Data Definitions

The mapping of a DTD to Natural data structures show the differences between the two. As the DTD does not specify how many records are to be included in the XML document, the XML Toolkit will assume a maximum number to be included. The application programmer may know the precise number and can adapt the data structure accordingly. A similar limitation exists with the length of the data: The DTD does not contain information about the length of the data in a record; therefore, the XML Toolkit will generate fields in the data structure with a length of 253 (the current maximum length).

Parsing the XML File and Assigning the Contents to Natural Data Structure

The XML Toolkit will generate Natural code based on the DTD. This code represents a subroutine which will be invoked from the Toolkit's "tokenizer" to assign the contents of the tags in the XML document to the corresponding Natural data structure.

Miscellaneous Enhancements

This section covers the following topics:

- Support of New 64-Bit Architecture
 - Operating System and Teleprocessing Interfaces
 - Profile Parameters
 - Natural Performance Enhancements
 - Size of Alphanumeric and Binary Variables
 - Size of Data Elements
 - Object Handles
 - Source Change Inventory
 - Non-Sliding Year Window (YSLW Parameter)
 - User Exit Libraries SYSEXT and SYSEXTP
 - The Stack
 - Improved Abend and Error Handling
 - Buffer and Storage Management
 - Reduction of Object Code
 - Change/Enhancements Requests
-

Support of New 64-Bit Architecture

Under the Z/OS operating system, Natural 4.1 will support IBM's new 64-bit architecture. This will apply to the Natural roll server (for cache), the Natural Authorized Services Manager (ASM)(for cache), and the buffer pool (e.g., for server Natural under OS/390), all of which at present use data spaces as local or global cache. With Version 4.1, you will be able to choose between data spaces and *memory objects* to be used as local or global cache for these parts of Natural.

Operating System and Teleprocessing Interfaces

The following enhancements to operating-system interfaces and teleprocessing-system interfaces will be provided with Natural 4.1:

- Natural 4.1 will support Complete Version 6 SMARTS/HTTP server environments.

Profile Parameters

- New Profile Parameters
- Enhanced Profile Parameters
- Retrieving Dynamic Profile Parameters

New Profile Parameters

The following new Natural profile parameters will be provided with Natural Version 4.1:

| Profile Parameter | Description |
|--------------------------|--|
| DS | Defines the default initial size of various Natural storage buffers (for example, ASIZE, DATSIZE). A corresponding new macro NTDS will also be available. |
| DUE | Allows improvedabend and error handling. |
| FREEGDA | Determines whether or not the currently used global data area is to be kept when a Natural utility is invoked, that is, whether the global data area is to be still available after the use of the utility or to be newly initialized. |
| OVSIZE | Defines the maximum total amount of variable storage which can be allocated outside the Natural thread. See Buffer and Storage Management below. |
| TMODEL | Defines the type of terminal model (screen size) being used. This is relevant for environments like IMS/TM, where the TP monitor is not capable of automatically supplying Natural with this information. |

Enhanced Profile Parameters

The following Natural profile parameters will be enhanced for Natural Version 4.1:

| Profile Parameter | Enhancement |
|-------------------|---|
| ADAMODE | New option ADAMODE=3. With ADMODE=2, two Adabas user-queue elements (UQEs) will be generated for the Natural session: one for the Natural system itself, and one for the user application's actual database access - plus a third UQE if a non-Natural program accessing Adabas is invoked during the session. ADAMODE=3 is identical to ADAMODE=2, except that for a non-Natural program no additional UQE will be generated, but the UQE of the user application's database access be used. |
| BPI | New option CSIZE=nnnn to set the size of the cache for the Natural local buffer pool. |
| CMPO | Same as for COMPOPT system command. |
| DU | New option DU=ABEND to produce a memory dump in the case of an abnormal Natural session end and terminate the session with an abend code (same as DU=ON, except that with DU=ON the session is terminated with an error message). This is particularly relevant under IMS/TM or for conditional JCL. |
| PRINT | New options: <ul style="list-style-type: none"> ● PADCHRI='x' and PADCHRO='y' to define the padding character to be used for input datasets and output datasets respectively on the print file. The default values will be PADCHRI=' ' (blank) and PADCHRO=' ' (blank). ● Access method AM=CICS to support print output under CICS. The reports can be written to temporary storage (TS) or transient data (TD). ● ROUTE=ON/OFF to control logical routing of print reports. |
| TD | New option TD=(time-zone). With this new option, Natural will compute the time differential not only at session start, but will also recompute it when date/time information is first requested within the Natural session after the specified point in time (SWT) has passed. SWT denotes the UTC (Coordinated Universal Time) point of time when daylight saving time is switched on or off. This will allow you to switch to/from daylight saving time without having to restart your Natural session. For performance reasons, the recomputation will take place once every 24 hours at the most; that is, once computed, the time differential will only be recomputed when date/time information is requested again after the specified point in time has passed again. A new macro, NTTZ, will be provided in the NATCONFIG configuration module, in which you can define the characteristics of different <i>time-zones</i> . |
| WORK | New option: PADCHRI='x' and PADCHRO='y' to define the padding character to be used for input datasets and output datasets respectively on the work file. The default values will be: PADCHRI=' ' (blank) and PADCHRO=X'00'. |
| WPSIZE | In addition to the storage block size, it will be possible to specify the maximum total size below and above 16 MB. This will make it possible to limit the non-thread storage area for a session. See Buffer and Storage Management below. |
| YSLW | New option to specify a certain year for the definition of a non-sliding year window. |

Natural Performance Enhancements

- In addition to the enhanced search algorithm of the new buffer pool manager that was made available with the Natural Turbo Plug-in, further enhancements of Natural runtime algorithms will provide faster access to objects already loaded in the Natural buffer pool. This will reduce the CPU consumption of object execution.
- The compression rate of the compression algorithm will be increased; this will reduce the compressed size of a Natural thread, which, in turn, will reduce the time needed for a roll-in/out event required particularly by a terminal I/O.

- The runtime algorithms for assignments, arithmetic operations and comparisons will be enhanced.
- The Adabas Multi-Fetch option will be made available for database statements (see Programming Language).

Size of Alphanumeric and Binary Variables

With Natural 4.1, the maximum possible size of an alphanumeric variable (format A) will be increased from 253 bytes to 1 GB. The maximum possible size of a binary variable (format B) will be increased from 126 bytes to 1 GB.

Size of Data Elements

With Natural 4.1, the maximum possible size of a single data element (array or indexed group) will be increased from 32 KB to 1 GB.

Object Handles

With Natural 4.1, it will be possible to define object handles within a global data area or as application-independent variables (AIVs).

Source-Change Inventory

In addition to the ZAP-inventory modules (e.g. NATAZAP) which exist for Natural and each of its subproducts and list all ZAPs applied to the product, Natural 4.1 will provide source-change inventory modules in which all source changes applied to a product will be logged. This will allow support personnel to ascertain the precise state of your product installation, which will facilitate error diagnosis.

Non-Sliding Year Window (YSLW Parameter)

With Natural 3.1, the YSLW profile parameter allows you to set a so-called "year sliding window" of 100 years to relate a 2-digit year value to a specific century. The value specified with the YSLW parameter determines how many years in the past - that is, before the current year - that 100-year range is to begin. In other words, the window range is always in relation to the current year.

With Natural 4.1, it will also be possible to set a "non-sliding" year window: You can set the year in which the 100-year range is to begin. For example, if you specify YSLW=1980, the window will be from the year 1980 to the year 2079, regardless of what year the current year is.

The Stack

A new Natural subprogram will be available, which will allow you to:

- ascertain the number of entries in the stack,
- ascertain the type of a stack entry (command, data),
- read a specific entry from the stack.

Thus it will be possible to retrieve information from the stack without having to use an INPUT statement; that is, without the other functions of an INPUT statement being performed, as these may not be desired if merely a stack entry is to be read.

User Exit Libraries SYSEXT and SYSEXTP

The SYSEXT library will provide a new user exit to process data areas, and transform them from Natural's internal representation format to an external user-readable format and vice versa.

The example library SYSEXTP will be integrated in the example library SYSEXT; that is, as of Version 4.1, the user exits that are now contained in SYSEXTP will be contained in SYSEXT.

Improved Abend and Error Handling

With the new profile parameter DUE, it will be possible to generate a snap dump whenever a specific Natural error occurs. This will facilitate error diagnosis, as it will not require the application of so-called "trap ZAPs".

Buffer and Storage Management

- With Natural 4.1, the internal I/O buffers, which are currently fixed-size buffers with a maximum size of 32 KB, will be allocated as variable buffers. This will avoid situations which lead to errors NAT1114 "internal output buffer overflow" and NAT1150 "attribute buffer overflow".
- Normally, Natural automatically increases the sizes of all its variable buffers (DATSIZE etc.) as required. Natural 4.1 will allow you to restrict this automatic increase if you wish: For each variable buffer and work pool, you will be able to define a maximum size in the NATCONFIG configuration module; the buffer / work pool will then be increased only up to the specified maximum size, but not beyond.
- At present, the source area, the fast locate table, the PF-key table and the loop table are located in the DATSIZE buffer. With Natural 4.1, each of them will be located in a buffer of its own.
- With Natural 3.1, the Natural thread is located either entirely above or below the 16 MB line. With Natural 4.1, a "mixed mode" will be possible, that is, some parts of the thread can be above 16 MB, and at the same time other parts can be below it.
- For the handling of large alphanumeric and binary variables, Natural's thread compression will be enhanced.
- Under OS/390 in batch mode and TSO, it will be possible for local buffer pools to be shared by multiple sessions.
- The new profile parameter OVSIZE will be provided to limit the total variable buffer storage outside the thread.
- The profile parameter WPSIZE will be enhanced for physical non-thread storage areas to allow the definition of total maximum values below and above 16 MB.

Reduction of Object Code

The following measures will be taken to reduce the size of the object code generated for a Natural programming object:

- With Natural 3.1, entries are generated in the Expanded Variable Description Table (EVDT) for all defined fields.
With Natural 4.1, an entry in the EVDT will be generated only for those fields which are actually referenced.
- Whenever the compiler detects an assignment of an initial value, it will generate a RESET statement instead, which will require less code. For example, for the statement MOVE 0 TO #NUMERIC the code generated will be the same as for the statement RESET #NUMERIC.
- For constants, even if they are used several times in the same program, only one entry will be generated in the constant table (KST).

The reduced object-code size of programming objects will yield several benefits: less storage space will be needed for the system files, the I/O rate for reading requested objects from the system file will be reduced, the Natural buffer pool can be kept smaller, etc.

Performance of Add-On Products

With Natural 4.1, the communication between Natural and its add-on products will be improved. This will in turn improve the performance of the add-on products.

Entire Connection

The Entire Connection version released in parallel with Natural 4.1 will support the so-called "stream mode" for uploading/downloading data to/from work files.

Change/Enhancement Requests

The following is an overview of the change/enhancement requests which will be implemented in Natural Version 4.1:

| Request Number | Description |
|----------------|---|
| CE0512 | Natural Advanced Facilities: New scan function for reports. (See Report Display.) |
| CE3291 | Natural Security: Allow special link of administrator to unprotected library. (See Linking Users to Libraries.) |
| CE4787 | SYSBPM: The length of the field "Use Count" for the global buffer pool will be increased from 5 to 8 bytes to be able to reflect use counts exceeding 99999. |
| CE4866 | Natural Security: New option to copy a library profile with or without links. (See Copying Libraries.) |
| CE5064 | Natural Security: Allow definition of default profiles for files. (See Administrator Services.) |
| CE5084 | Natural Security: After occurrence of Adabas response code 9 within current Natural session, OPRB value as determined by Natural Security will be used for new database open processing. |
| CE5087 | Natural Security: Enable two-phase maintenance/activation of security profiles: one user makes changes to a security profile, but the changes will only take effect after another administrator has authorized them. (See Two-Phase Maintenance and Activation of Security Profiles). |
| CE5098 | Support use of Julian date format in MASK option. (See MASK Option in Logical Condition). |
| CE5107 | Natural Advanced Facilities: Asterisk notation for report selection criteria. (See Report Display.) |
| CE5108 | Natural Advanced Facilities: New selection options for deleting reports from the spool file. (See New Functions). |
| CE6270 | Natural Advanced Facilities: New report status WAIT to trigger printing of reports not yet printed. (See Report Display.) |
| CE7138 | SYSTP: Display Natural user sessions of a specific user/terminal. (See SYSTP Utility.) |
| CE7808 | Make Natural version number available as system variable. (See New System Variables.) |
| CE7892 | SYSBPM: Make the direct commands DISPLAY BUFFERPOOL, SELECT BUFFERPOOL and RESET BUFFERPOOL available as functions that can be selected from one of the SYSBPM menus. |

| | |
|--------|--|
| CE7959 | New ESCAPE statement option for leaving inline subroutine <i>and</i> programming object containing same. (See ESCAPE Statement.) |
| CE8586 | CATALL: Provide a user exit to change the default settings of the main CATALL screen. |
| CE8817 | NATLOAD: Provide a user exit to be invoked after processing of NATLOAD has finished. (See SYSUNLD Utility.) |
| CE8886 | Natural Security: The number of unsuccessful logon attempts will be passed as parameter to user exit LOGONEX1. (See Logon Procedure.) |
| CE8891 | COMPOPT: The option V22COMP (allow old Version 2.2 syntax) will continue to be available with Version 4.1. (See COMPOPT Command.) |
| CE8901 | Natural Security: Allow specification of a group ID in addition to the user ID when logging on. |
| CE8906 | Roll server: User exit NATRSUX1 will provide additional information on each roll file (number of slots defined and in use). |
| CE8907 | Support of IBM's LE (language environment) user-defined error handlers. |
| CE8944 | Suppress error NAT1074 (global buffer pool not found) if an alternative buffer pool is used instead. |
| CE9048 | Natural Security: New function to automatically update/delete the functional security of all command processor whose status is "modified" or "unresolved". (See Functional Security.) |
| CE9063 | Natural Security: New function to create and maintain default profiles for libraries. (See Administrator Services.) |
| CE9159 | SYSTP: Simultaneous cancellation of multiple Natural user sessions depending on their last-activity date. (See SYSTP Utility). |
| CE9266 | Retrieve information from the stack without using an INPUT statement (See The Stack). |
| CE9292 | Natural Security: New option to adjust the physical content of a library on the FUSER system file accordingly when a Natural Security library profile is renamed, copied or deleted. (See Copying, Renaming and Deleting Libraries). |
| CE9344 | Natural Security: Control the display of mailboxes in batch mode by: a) a general setting in Administrator Services, b) for every mailbox individually. (See Batch Mode.) |
| CE9451 | READ statement: Evaluation of ENDING AT values not done by Natural, but by the database system. (See READ Statement). |
| CE9452 | Enhanced possibilities for usage of notation "&n&" for dynamic insertion of values in copycodes. |
| CE9687 | Natural Security: A warning message "your password will expire in <i>nnn</i> days" will be issued to the user. (See Logon Procedure.) |

Natural Security

The following enhancements will be provided with Natural Security Version 4.1:

- Logon Procedure
 - Administrator Services
 - Two-Phase Maintenance and Activation of Security Profiles
 - Library Profiles
 - Utility Profiles
 - Links
 - Functional Security
 - Batch Mode
 - Transferring Security Data
 - Interface Subprograms
-

The following enhancements will be provided:

- It will be possible to define a warning message "your password will expire in *nnn* day" which will be issued to users at the initial logon. In addition, you will be able to set an activation date for this message.
- The number of unsuccessful logon attempts will be passed as a parameter to the logon-related user exit LOGONEX1. Thus, it will be possible, for example, to display corresponding information to the user *before* the maximum number of logon attempts is reached.

Administrator Services

Default Profiles

At present, default security profiles can only be defined for users. With Version 4.1, it will also be possible to define default profiles for all other types of Natural Security objects.

Two-Phase Maintenance and Activation of Security Profiles

At present, when a security profile is created or modified, its definitions are "activated" - that is, take effect - immediately (unless an activation date is set, of course); that is, the maintenance work of specifying definitions and the "activation" of these definitions are done simultaneously and lie in the hands of the same person.

With Version 4.1, a new two-phase mechanism will be provided which will allow you to separate maintenance from activation: one user will be allowed to do the actual maintenance work, that is, enter or modify the data in a security profile; however, these specifications/changes will only take effect after another administrator has authorized them. Unlike countersignatures, this authorization will not be given prior to an individual maintenance task to be performed, but at any time after the maintenance task, and it can be given for multiple security profiles all in one go. This will allow you to distribute the workload involved in maintaining security profiles without undermining the protection and control mechanisms provided by Natural Security's administrator concept.

Library Profiles

Statement and Command Restrictions

The use of the new statements and system commands provided with Natural Version 4.1 can also be controlled with Natural Security.

Private Libraries

At present, access to a private library is restricted to the user for whom the private library is defined.

With Version 4.1, it will be possible to remove this restriction: Private libraries will then be treated like other "normal" library, and you will be able to control their use like that of other libraries.

This will be implemented as follows:

- A general option will be provided with which you can determine whether the old (Version 3) or new (Version 4.1) private library concept is to apply.
- If you select the old concept, the handling of private libraries will be unchanged, as it is now.
- If you select the new concept, you will be able make a type specification for each private library individually within its security profile:
 - Type A: The library can only be accessed by the user to which it is attached, that is, whose user ID is the same as the library ID (as with Version 3).
 - Type B: The library is treated like an unprotected library, that is, it may be accessed by any user.
 - Type C: The library is treated like a protected library, that is, it may only be accessed by the user whose ID is the same as the library ID and by users who are linked to it (or are in a group linked to it).
- Private libraries of type A will continue to be maintained in the the user maintenance section of Natural Security, whereas private libraries of types B and C will be maintained in the library maintenance section.
- An interface subprogram will be provided to facilitate the transition from the old to the new concept.

Linking Users to Libraries

At present, it is not possible to establish a link between a user and an unprotected library. The conditions of use of the library are determined by the library profile.

With Version 4.1, it will be possible to establish a special link between an administrator (or group of administrators) and an unprotected library. Thus it will be possible to define special conditions of use for administrators if this should be required for administration or maintenance tasks.

Copying Libraries

For the Copy Library function, a new option "with links" will be provided. This will allow you to copy not only the library profile, but also existing links associated with that library profile (similar to the "with links" option of the Copy User function).

Copying, Renaming and Deleting Libraries

At present, when you copy, delete or rename a library security profile on the FSEC system file, this has no effect on the library itself and its contents stored on the FUSER system file.

With Version 4.1, a new option will allow you to also adjust the FUSER system file accordingly when a library profile is copied, deleted or renamed: the contents of the library on the FUSER file would then also be copied to another library, deleted, or moved to another library.

Utility Profiles

With Version 4.1, it will be possible to control the use of the Natural utilities SYSTP and SYSDB2.

Links

At present, when you invoke a function for the maintenance of links, you will get a list of all objects to which the selected object can be linked, that is, those for which links already exist and those for which not.

With Version 4.1, an new option will be provided which allows you to display either a list of all linkable objects or a list of only those objects which are already linked. This new option will be available for all link maintenance functions.

Functional Security

If the status of a command processor is "modified" or "unresolved" (that is, modified or deleted with SYSNCP), you have to update or delete respectively the functional security defined in Natural Security for the command processor. At present, you have to make this adjustment for each command processor individually. With Version 4.1, an new option will allow you to automatically update/delete the functional security of all "modified" or "unresolved" command processors.

Batch Mode

At present, mailboxes are not displayed in batch mode. With Version 4.1, it will be possible to define for each mailbox whether it is to be output in batch mode or not. In addition, it will be possible to set in Administrator Services a general option which determines whether mailboxes are to be output in batch mode or not.

Transferring Security Data

New import and export functions will be provided for the transfer of security data between FSEC system files. These new functions will replace the existing programs SECULD and SECLOAD and will provide for enhanced and expanded transfer capabilities. The new functions will not only be available as menu-driven functions (like SECULD and SECLOAD), but also via direct commands to be applied to individual security profiles: you will, for example, be able to unload/load a single security profile by marking it with the corresponding command on a profile maintenance selection list.

SECULD and SECLOAD will cease to be supported with one of the subsequent releases.

Interface Subprograms

- New interface subprograms will be provided to perform mailbox maintenance functions from outside the Natural Security library SYSSEC.
- A new interface subprogram will allow you to retrieve the user name belonging to a specific user ID, and the user ID belonging to a specific user name.

Natural Advanced Facilities

The following enhancements will be provided with Natural Advanced Facilities Version 4.1:

- Spool File
 - New Functions
 - CICS-Specific Enhancements
 - IMS/TM-Specific Enhancements
 - Spool File Options
 - Server Options
 - Report Display
 - Printer Definitions
 - CALLNAT Interfaces
-

Spool File

With Version 4.1, a new spool file layout will be provided for improved data storage.

A conversion program for existing Version 2.3 spool files will also be provided.

New Functions

The following new functions will be available:

- The application SYSPPOOL will support switching to other Version 4.1 spool files (dynamic change of FSPOOL assignment).
- It will be possible to unload and load spool file objects and report data from/to work files (work file numbers 3 and 4) online.
- It will be possible to copy objects from one spool file to another.
- It will be possible to delete from the spool file only reports which are created before a specific date or which are older than a specific number of days.

CICS-Specific Enhancements

In addition to the online definitions (CICS Options), it will be possible to use the parameter module NAFPARM. Parameter settings of the NAFPARM module will then override parameter definitions on the spool file. If no parameter module is used, the online definitions on the spool file will apply. The same will be true for parameters not set in the parameter module.

IMS/TM-Specific Enhancements

In addition to the online definitions (IMS/TM Options), it will be possible to use the parameter module NAFPARM. Parameter settings of the NAFPARM module will then override parameter definitions on the spool file. If no parameter module is used, the online definitions on the spool file will apply; the same will be true for parameters not set in the parameter module.

It will be possible to use a spool server (IMS BMP) with the option WAIT FOR INPUT for different spool files, as the corresponding FSPOOL value will be passed dynamically.

Spool File Options

It will be possible to pre-define the sequence in which reports are displayed:

- ascending by job number;
- ascending by: a) destination/form, b) status, c) priority, d) date of creation;
- ascending by: a) user ID, b) destination/form, c) status, d) priority, e) data of creation;
- automatically depending on the specified search criteria.

Server Options

It will be possible to automatically start reports whose status is WAIT (see also Report Display below).

Report Display

- A new line command DY will allow you to delete reports depending on the date or the number of days.
- A new line command SC will allow you to search for texts within a report.
- The display of report details will be enhanced.
- Two new report status values will be provided:
 - WAIT = Report could not be started, because the printer was active for another destination/form;
 - WOER = Error during processing of report data.
- Asterisk notation will be provided for the report selection criteria.

Printer Definitions

Control of the first/last page advance will be enhanced, allowing you to prevent a page advance being replaced by a line advance.

At present, the user exit USPSE01 in the spool server applies to all printers. With Version 4.1, you will be able to use a different user exit for each printer.

CALLNAT Interfaces

The existing interface subprograms will be expanded to cover the new functions (deletion by date/days, unloading/loading of spool file objects/report data).