Changes and Enhancements

This chapter covers the following topics:

- International Support
- Natural System Files
- Editors
- Configuration Utility
- Other Utilities
- Parameters
- Statements
- System Functions
- System Variables
- System Commands
- Application Programming Interfaces
- Error Messages Corresponding to Adabas Response Codes
- Natural Web Interface
- XML Toolkit
- Natural Connection
- Miscellaneous

International Support

Natural for OpenVMS now includes the international support which was previously only provided with the international version of Natural 5.1.1.

Bidirectional languages and double-byte characters are now supported. For detailed information on how to activate these features, see the sections *Bidirectional Language Support* and *Double-Byte Character Support* in the *Unicode and Code Page Support* documentation.

The following has changed (as compared with the international version of Natural 5.1.1):

- In the map editor, the **Flip Map** command has been renamed to **Reverse Map**.
- The environment variable `RTL_BW_COMP` is no longer supported. Natural no longer supports logical resorting of the characters during input; the characters are always displayed in the keying sequence for left-to-right fields and in the reversed keying sequence for right-to-left-fields.

- The environment variable `NATLANG` is no longer supported. For enabling the international support, the setting of the profile parameter `CP` is now used.
- The locale files (previously installed in *etc/locale*) which were delivered with the international version of Natural 5.1.1 are no longer supported.
  - Month and day names are now retrieved from the Natural text/error message files.
  - Character property information such as "is the character upper case" is retrieved from the ICU library which is used for the Unicode support.
  - Using the AM/PM strings from the locale files is no longer supported.
  - Using the year edit masks `EC`, `EE`, `EV` and `EY` is no longer supported.
  - Characters which are entered in a right-to-left field are no longer automatically converted according to the keymap defined in the locale files. Natural now uses the characters which are delivered from the operating system, which is compatible to the Windows behavior. Changing the input language has to be achieved by executing the appropriate operating system functions.

Natural System Files

The structure of the Natural library directory file *FILEDIR.SAG* has changed. The new format of the file is now platform-independent and it can be exchanged between all Windows, UNIX and OpenVMS platforms. The old format can still be read by Natural for OpenVMS Version 6.3 and will be automatically converted into the new format whenever a modify access is made. Older versions of Natural cannot read the new *FILEDIR.SAG* format. For detailed information, see *Portable Natural System Files* in the *Operations* documentation.

Editors

Data Area Editor

2-digit level support: the level number has been increased to 99 digits.

In addition, if you use a new functionality in the data area editor (e.g. level > 9), the source will be saved in a new internal format. This means that these new types of data areas cannot be read with the previous version of Natural.

For details, see *Data Area Editor* in the *Editors* documentation.

Configuration Utility

User Interface

The user interface of the Configuration Utility (previously known as NATPARM utility) has been redesigned.

Database Management System Assignments

It is now possible to specify the database type ADA2. This database type supports alphanumeric objects (Adabas LA option) and large object database fields (LOBs), which are dynamically defined inside a view, and view sizes greater than 64KB. It can be used for Adabas as of version 6 on Open Systems and Adabas as of version 8 on mainframes. See *DBMS Assignments* in the *Configuration Utility* documentation.

HTTP Client Parameter Settings

The global configuration file now offers the possibility to specify the following parameters for HTTPS (HTTP with SSL): `SSLPRX`, `SSLPRXPT` and `NOSSLPRX`. See *HTTP Client Parameter Settings* in the *Configuration Utility* documentation.

System Files

The global configuration file now offers the possibility to specify alias names for the system files. This feature applies when working in a remote development environment (SPoD) with Natural for Windows. These alias names are then shown in Natural Studio’s library workspace. See *System Files* in the *Configuration Utility* documentation.

Batch Mode

The new profile parameters `CPOBJIN`, `CPSYNIN` and `CPPRINT` can be specified. See *Batch Mode* in the *Configuration Utility* documentation.

Compiler Options

The new profile parameters `KCHECK` and `TQMARK` are now used in the Configuration Utility instead of `KC` and `TQ`.

When you export a parameter file using the Configuration Utility, `KCHECK` and `TQMARK` are now written to the resulting text file (instead of `KC` and `TQ`). When you import a parameter file which has been exported with an older version of Natural, this file is still read in correctly.

The new profile parameters `MASKCME` and `PCHECK` can be specified.

The new profile parameter `PSIGNF` can be specified.

See *Compiler Options* in the *Configuration Utility* documentation.

Error Handling

The profile parameter `NOAPPLERR` can now be specified.

The new profile parameter `RTINT` can be specified.

See *Error Handling* in the *Configuration Utility* documentation.

Device/Report Assignments

The **Close mode** option is now defined for the devices. In the previous version it was defined for the reports. See *Device Assignments* in the *Configuration Utility* documentation.

The printer device TRANSFER has been replaced by a new report device assignment. See *Report Assignments* in the *Configuration Utility* documentation. See also *Defining Printers* in the *Natural Connection* documentation.

Regional Settings

The new profile parameters SUTF8, SRETAIN, CPCVERR and CP can be specified.

Note:
The existing CP parameter (used with Natural Remote Procedure Call) has been renamed to CPRPC. See *Remote Procedure Call* in the *Configuration Utility* documentation.

The new profile parameter SUBCHAR can be specified.

See *Regional Settings* in the *Configuration Utility* documentation.

Work Files

The profile parameter TMPSORTUNIQ can now be specified.

A new work file type, CSV (for comma-separated files), can now be specified.

It is now possible to open a work file in append mode and to delete a work file after closing it. Different attributes can be defined for these cases.

It is now possible to define a code page for the work file types ASCII, ASCII compressed, Unformatted and CSV.

For the work file types which write code page data (ASCII, ASCII compressed, Unformatted and CSV), it is now possible to write a byte order mark (BOM). The prerequisite is that the code page UTF-8 is defined for the work file.

See *Work File Assignments* in the *Configuration Utility* documentation.

Remote Procedure Call

The new profile parameter SRVWAIT can be specified. See *RPC (Server)* in the *Configuration Utility* documentation.

Web I/O Interface

"Web I/O Interface" is a new category which is available in the parameter group "Product Configuration". The new profile parameter WEBIO can be specified in this category. See *Web I/O Interface* in the *Configuration Utility* documentation.

Other Utilities

Utility Activation

As of this version, Natural invokes a Natural utility without performing a logon to the corresponding utility library in the FNAT system file. As a result, Natural preserves the global data area (GDA) and/or application-independent variables (AIV). The current user library and the settings are maintained. To reset the GDA and/or the AIVs, see the parameter FREEGDA. The following utilities can now be used in this manner:

- SYSERR
- SYSNCP
- Object Handler
- SYSRPC
- Natural Security

The behavior described above can have an impact on batch procedures from previous Natural versions that contain logon instructions to utility libraries. For more information, see the *Tools and Utilities* documentation.

FTOUCH

The `convert` option of the utility `FTOUCH` can now be used to migrate an old *FILEDIR.SAG* file into a new portable *FILEDIR.SAG* file, as supported with Natural Version 6.3 for OpenVMS and with Natural Version 6.2.1 for UNIX and Windows.

The following new options are provided: `encoding`, `endian` and `objname`.

The new option `-ignoreext` specifies that files with unkown extensions contained in a library are ignored.

You can now define that line numbers are not written to the source code in the file system.

In the Natural main menu, a minus (-) character after the library name indicates that in this library the line numbers are suppressed in the source code.

See *FTOUCH Utility* in the *Tools and Utilities* documentation.

INPL

The `INPL` utility now automatically recognizes the type of the load file. It is no longer necessary to specify the correct load file type. See *INPL Utility* in the *Tools and Utilities* documentation.

Object Handler

Setting up a profile for defining individual defaults and standard procedures is now possible. You can define an individual profile for your Object Handler utility by modifying the standard Object Handler user profile and specifying general or user-defined parameter settings, such as activating Advanced-user mode by default, or specifying default settings for work files, report files and Workplans. For more information, see *Object Handler - Profile Settings* in the *Tools and Utilities* documentation.

The Transfer format can now also be used for unloading or loading sources of Natural objects in Unicode/UTF-8 (Universal Transformation Format, 8-bit form). See *Transfer Format* in the *Object Handler* section of the *Tools and Utilities* documentation.

The Object Handler has been enhanced to

- support unloading of Predict sets,
- allow the specification of a library name for the Predict XRef save set option,
- allow direct output of report data for Object Handler functions executed in batch or with Application Programming Interface without storing the data in a file.

The new command `DISPLAY STATISTICS` allows to display statistical data of the objects processed when working in command mode or when a workplan of type PROCEDURE is processed.

The Object Handler now automatically recognizes the type of the load file for the load and scan functions. It is no longer necessary to specify the correct load file type. The user interface of the Object Handler has not been changed due to compatibility reasons, however, the **Portable work file** option will not be evaluated for the load and scan functions. See the description of `WORKFILETYPE` in *Keyword Explanation of option-setting*.

The `DELETEALLOWED` option can be used in an Object Handler direct command to specify that the delete instructions from work files created with `NATUNLD` (which is no longer installed by default) are processed.

SYSRPC

See *Natural Remote Procedure Call (RPC)* later in these *Release Notes*.

Parameters

ACIVERS

The EntireX ACI Version 9 is now supported by `ACIVERS`. Therefore, the maximum value has been increased from 8 to 9. See also the corresponding information in the section *Natural Remote Procedure Call (RPC)* later in these *Release Notes*.

HE

The session parameter `HE` and the corresponding field in the map editor (both used to specify the name of a helproutine to be assigned to a field) have been extended by *n*`X` notation. This notation allows you to omit parameters for which no values are to be passed.

LT

The range of possible values for the profile and session parameter `LT` has been changed from 0 - 99999999 to 0 - 2147483647. The default setting for `LT` is still 99999999.

`LT=0` shuts off the counting mechanism for loops. This means that there is no limit to the number times that a loop is run through when `LT=0`.

OPRB/ETEOP

The functional scope of the profile parameter `OPRB` has changed. Handling of implicit `END TRANSACTION` calls at the end of a program are no longer handled by `OPRB`, but now part of the functional scope of the profile parameter `ETEOP`. This offers greater flexibility. The following table shows the settings necessary to achieve the same result as with the former OPRB settings in earlier Natural versions:

Former OPRB Setting	New ETEOP/OPRB Settings
OPRB=OFF (ET is issued)	OPRB=OFF and ETEOP=ON
OPRB=NOOPEN (ET is not issued)	OPRB=OFF and ETEOP=OFF
OPRB=(ACC=...) (ET is not issued)	OPRB=(ACC=...) and ETEOP=OFF

Migration takes place automatically if a parameter file of an older Natural version is accessed. No user action is necessary.

The profile parameter OPRB can now be used to specify the encoding for W fields in an Adabas user session.

STEPLIB

The profile parameter LSTEP has been renamed to STEPLIB. The information which was previously provided for the STEPLIB parameter is now available under *Additional Steplib Assignments* in the *Configuration Utility* documentation.

The default value assigned to the profile parameter STEPLIB has been changed from "STEP" to "SYSTEM". Therefore, you have to take precautions if you create new parameter files and your applications use these newly created parameter files and depend on objects located in the library STEP (the former default value). The modification of the default value implies an alteration in the default library search order (steplib). Therefore, your applications might not run properly or might produce unexpected results.

When you are using old parameter files in which the value "STEP" is defined for the profile parameter STEPLIB, your existing applications which depend on objects located in library STEP will still run properly.

If your applications do not expect objects to be located in library STEP on FUSER or if library STEP on FUSER is empty, your applications will run properly after the default value change of STEPLIB. The performance is increased since the library search order decreased in size. You are encouraged to modify STEPLIB in your current installation to the proposed new default value in order to take advantage of the above mentioned performance increase.

USIZE

The default setting of the profile parameter USIZE has been been changed to 20 MB (previously 10 MB).

WORK

CLOSE MODE is a new option of the profile parameter WORK. It specifies the work file / printer close mode.

YSLW

It is now possible to specify a "year fixed window" with the profile parameter YSLW. This enhancement is provided for compatibility with Natural for Mainframes.

Statements

CALLNAT, FETCH, PERFORM

The performance has been enhanced for the statements CALLNAT, FETCH and PERFORM.

DECIDE ON

The SUBSTRING option can now be used as *operand1* and *operand2* in the DECIDE ON statement. It allows you to check only a certain part of an alphanumeric, Unicode or binary field.

DEFINE DATA

The following are enhancements to DEFINE DATA:

- arrays of dynamic variables are now allowed;
- extensible arrays or X-arrays are now allowed;
- the INIT and CONST clauses are new;
- DEFINE DATA now supports up to 99 levels.

When using the keywords BY VALUE, BY VALUE RESULT and OPTIONAL after a variable definition inside a DEFINE DATA LOCAL statement, you will receive an error. These keywords are only allowed in the parameter data area using DEFINE DATA PARAMETER. Nevertheless, you may use these keywords in parameter data areas in a USING clause of DEFINE DATA LOCAL.

```
DEFINE DATA LOCAL 1 #A(A1) BY VALUE /* NAT0635 END-DEFINE
END
```

DEFINE PRINTER

When using the OUTPUT option TRANSFER, all output data written to this report number are sent to an Entire Connection terminal. In other words, output data which would normally appear on the screen are written into an NCD file.

It is now possible to specify report number 0 as a valid report number with the DEFINE PRINTER statement. This enhancement is provided for compatibility with Natural for Mainframes.

DEFINE WORK FILE

It is now possible to specify attribute operands with the DEFINE WORK FILE statement. Using the different attribute operands, it is possible to open a work file in append mode and to delete a work file after closing it.

It is now possible to define the new work file type CSV. See also *CSV Format* in the *Operations* documentation.

For the work file types which write code page data (ASCII, ASCII compressed, Unformatted and CSV), it is now possible to specify the attribute operand BOM (writes a byte order mark) or NOBOM (does not write a byte order mark; default) with the DEFINE WORK FILE statement. The prerequisite is that the code page UTF-8 is defined for the work file.

DOWNLOAD PC FILE

For the statement DOWNLOAD PC FILE new options are available:

- **SYNC**
  With SYNC, you specify that the PC returns control to Natural after executing and terminating a COMMAND. SYNC can be used, for example, to ensure that the command SET PCFILE has been executed before a file transfer starts.

- **ASYNC**
  With ASYNC, you specify that the PC immediately returns control to Natural, regardless of whether the execution of a COMMAND has been terminated or not.

DISPLAY, INPUT, PRINT, REINPUT, STACK, WRITE

Dynamic variables may be used as an operand.

EXPAND, REDUCE, RESIZE

The following statements can now be used for the X-arrays:

- EXPAND
- REDUCE
- RESIZE

These statements are used to change the number of occurrences of an X-array.

EXAMINE, COMPRESS, SEPARATE

In addition to alphanumeric format, these statements now support binary format.

EXAMINE

The EXAMINE statement has two new clauses: DIRECTION and POSITION.

The EXAMINE statement can now be used to examine graphemes. The new clauses CHARPOSITION and CHARLENGTH are used for this purpose.

FIND

The FIND statement now allows a non-descriptor search.

FIND, HISTOGRAM and READ

Multi-fetch processing can be enabled for the statements FIND, READ and HISTOGRAM that do not involve database modification. For further details on the statements that can be used with the Multi-Fetch Clause, see *Multi-Fetch Clause* in the *Programming Guide*.

FOR

An arithmetic expression can now be used as *operand2*, *operand3* and *operand4* in the FOR statement.

MOVE

It is now possible to specify the following options with the MOVE statement:

- MOVE NORMALIZED

  Converts a Unicode string into the "Unicode Normalization Form C" (NFC). The resulting Unicode string does no longer contain combining sequences for characters which are available as pre-composed characters.

- MOVE ENCODED

  Translates a character string, encoded in one code page, into the equivalent character string of another code page.

READ

The READ statement has been enhanced by the new DYNAMIC clause. With DYNAMIC, you can specify the direction of loop processing dynamically.

The READ statement is enhanced by a WITH REPOSITION clause. The complete READ loop can be restarted by a ESCAPE TOP REPOSITION statement within the READ loop.

READ and HISTOGRAM

The READ and HISTOGRAM statements have been enhanced by a new TO clause. This is used for a different type of end value processing.

The READ and HISTOGRAM statements have been enhanced by new comparators for start value processing. They are:

- LT
- GT
- LE
- GE

REQUEST DOCUMENT

It is now possible to specify an encoding with the REQUEST DOCUMENT statement. The new ENCODED clause is used for this purpose.

HTTPS (HTTP with SSL) requests can now be handled by the REQUEST DOCUMENT statement if OpenSSL is installed.

SET GLOBALS

For the SET GLOBALS statement, the restriction that it can be applied in reporting mode only has been removed. This is to allow you to specify the CPCVERR session parameter (Code Page Conversion Error) for a Natural application that is to support Unicode also in structured mode. Also, the other session parameters available with this statement can now be set with SET GLOBALS in structured mode.

New Object Type Function in Statements and Expressions

Function calls can be used directly in statements or expressions, replacing operands. For further details, see *User-Defined Functions* in the *Programming Guide*.

Use of Large Constants

The limit of 253 bytes for alphanumeric and hexadecimal constants is no longer valid. The maximum size of one constant is 1073741824 bytes (1 GB).

SUBSTRING Option

The SUBSTRING option now supports binaries. For more information on this option, see *Relational Expression* in the *Programming Guide*.

System Functions

On OpenVMS, Windows and UNIX platforms, the behavior of the system functions has been changed in such a way so that they are compatible with the system functions on mainframe platforms. When an argument of a system function contains an array with a variable index, it is checked whether this argument is a scalar value or an array with several occurrences. A temporary variable is created as a scalar or array, according to the referenced variable in the argument which is used as the result of the whole system function.

Example:

```
DEFINE DATA LOCAL
  1 #BREAK (A1/1:11) INIT <’a’,’a’,’a’,’a’,’a’,’b’,’b’,’b’,’b’,’b’>
  1 #VALUE (P3/1:11) INIT <1,2,3,4,5,6,7,8,9,10>
  1 #I1    (P3)
END-DEFINE
*
F1. FOR #I1 = 1 TO 10
  PERFORM BREAK (F1.)
  AT BREAK OF #BREAK (#I1)
    /* Array with variable index which references only a scalar value!
    PRINT SUM(#VALUE (#I1)) SUM(#VALUE (#I1+1))
    /* Array with variable index which references an array with 2 occurrences!
    PRINT SUM(#VALUE (#I1:#I1+1))
    PRINT ’-’(10)
  END-BREAK
END-FOR
END
```

Output:

```
  15   20
  15   20
----------
  40   34
  40   34
----------
```

System Variables

*APPLIC-NAME

If Natural Security is installed, the Natural system variable `*APPLIC-NAME` contains the name of the library to which the user is logged on. If the user is logged on via a special link, the system variable now contains the link name instead of the library name.

*NATVERS

The format of `*NATVERS` has been changed to *rr.vv.ss* (where *rr*=release, *vv*=version, *ss*=system maintenance level). It is now compatible with the format used on the mainframe.

For example, the following program will now display 06.03.04 instead of 6.3.4:

```
DISPLAY *NATVERS
END
```

*SERVER-TYPE

The new value "WEBIO" of the `*SERVER-TYPE` variable indicates that Natural has been started as a Natural Web I/O Interface server.

System Commands

CATALL

The system command `CATALL` can now be used in batch mode. The direct command syntax has been enhanced accordingly.

The system command `CATALL` can now also be used to check, save or stow objects. By default, the source-code lines of sources that were saved or stowed are now automatically renumbered.

When Natural Security is active, it is now checked whether the selected action (catalog, check, save or stow) is allowed under Natural Security. When it is not allowed, an occur occurs.

COMPOPT

It is now possible to specify the following parameters with the `COMPOPT` command (in addition to the parameters `DBSHORT` and `GFID`):

- `KCHECK`
- `MASKCME`

- PCHECK
- PSIGNF
- THSEP
- TQMARK

LIST DIR

When you issue the command LIST DIR *object-name*, encoding information is now shown. It shows the code page with which the source has been saved. If a code page has not been specified, this field remains empty.

STRUCT

The result of the system command STRUCT is now the same as on the mainframe.

TECH

The system command TECH has been enhanced to display the code page and locale used in the current Natural session. (The same applies when the terminal command %<TECH is used instead.)

Application Programming Interfaces

The following application programming interfaces (APIs) in the library SYSEXT have been enhanced:

API	Description
USR2026N	Get TECH information. This API can now read the contents of the system variables *CODEPAGE and *LOCALE.
USR2011N	Get or set a work file name. This API has the following new parameters: ● **WORK-ATTRIBUTES (A1000)** Matches the ATTRIBUTES clause of the DEFINE WORK FILE statement. Currently possible values are APPEND/NOAPPEND, BOM/NOBOM and DELETE/KEEP. ● **WORK-CODEPAGE (A64)** Matches the work file property **Code page** which is allowed for the work file types ASCII, ASCII compressed, Unformatted and CSV. Both parameters are optional. They must always be specified together (i.e. you must either specify both parameters or none of them). When only one parameter is specified, it is ignored.

Error Messages Corresponding to Adabas Response Codes

The Natural error messages that correspond to Adabas response codes have been completely revised with Natural Version 6.3.

Natural displays subcodes or other information from Additions fields if they are provided by Adabas.

Natural Web Interface

The following changes/enhancements to the Natural Web Interface are provided with this version of Natural:

- Server Extensions
- New Library SYSWEB3
- Changes to SYSWEB

Server Extensions

Natural Web Interface server extensions are now available as Apache module for Apache 2.0.*x* or Apache 2.2.*x*. For further details, refer to the following sections in the *Web Technology* documentation

- *Natural Web Server Extensions* - *Introduction for SYSWEB3*
- *Supported HTTP Servers*
- *Natural Web Server Extensions for MOD*

Using the Web Interface on a remote machine without EntireX technology is now possible via the PAL interface as used for the SPoD server.

The server extension program initializes a communication directly to the PAL server instead of, for example, using RPC. Because all Web Interface programs are subroutines, and it is not possible to run a subroutine directly using PAL, a generic stub is necessary. This stub handles steplib setting, parameter transfer and `CALLNAT` of the subprogram.

For further information, see *Communication Using PAL Techniques* and *Functionality* in the *Web Technology* documentation.

New Library SYSWEB3

A new version of the Web Interface (SYSWEB3) is provided.

Multi-byte character sets (such as Japanese) can now be used with Natural DCOM. A prerequisite for this is that you use the new Natural library SYSWEB3. See *Natural Web Online Documentation SYSWEB3* in the *Web Technology* documentation.

For information on how to convert your old SYSWEB applications to SYSWEB3, see *Migrate Natural Web Interface SYSWEB to SYSWEB3*.

Usage of dynamic variables:

- The interface PDA W3PARM has been modified to use (A) dynamic and (B) dynamic variables instead of `(A250/1:v Array)`.
- A separation of transferred data is no longer necessary - this avoids errors during character conversion.
- Fewer calls are made to the EntireX Broker because of a larger communication area.
- A source change in existing programs is not necessary.
- A recatalog of existing programs with the new library `SYSWEB3` is mandatory and the new Web Interface server extensions have to be used.
- The old interface and the library `SYSWEB` together with the old Web Interface server extensions are still delivered.
- The size of the incoming data (containing variables) is no longer limited to approx. 30000 bytes.
- It is no longer required to modify W3PARM to change the size of the incoming data or outgoing data.
- A recatalog of library `SYSWEB3` is no longer necessary to change configuration settings.
- To change the maximum size of the incoming data and the outgoing data, only the new program W3LIMIT needs to be changed (it may be required to increase EntireX Broker or RPC sizes).
- Incoming data no longer needs to be of mime type "application/x-www-form-urlencoded" (the check is no longer performed by the Web Interface server extension, but at W3INIT instead).
- The request method `PUT` is supported (the check is no longer performed by the Web Interface server extension, but at W3INIT instead).
- Incoming or outgoing data is no longer limited to alphanumeric data format. Binary data can be transferred e.g. via multipart-form-data upload from a browser, including pictures or delivering pictures, for example from the Natural resource directory.
- Parameter settings at the URL (`QUERY_STRING`) in combination with the `POST` method are supported.
- The number of parameters delivered to the HTTP server via the `POST` method is no longer limited to 200 parameters. The number of parameters delivered via the `POST` method now depends on the data size.
- It is now possible to set a response number other than 200 for HTTP requests from a Natural program (for example, as HTTP `PUT` response). See *Overview of Response Numbers for HTTP/HTTPs Requests* in the *Statements* documentation for a list of possible response numbers.

W3* programs for binary data and mime-types:

- There is a new W3BINARY call to transfer back binary data.

- W3INFO and PDA W3PINFO have been extended to be used with dynamic in/out data.
- W3INIT has been extended with optional parameters to allow mime types other than "application/x-www-form-urlencoded" and HTTP `PUT` requests.
- There is a new subroutine W3READ-INPUT to get all transferred data. For example, alpha and binary data, header and parameter (name/value pair) delivered via the `POST` method and/or `QUERY_STRING`.
- The routine W3MIME-DATA together with the Natural text member MIMEDATA controls the transport type (binary or alpha) of the return page for the given content type (W3CONTENT-TYPE).
- The routine W3MIME-TYPE together with the Natural text member MIMETYPE defines the file extension for given content types.

NAT* programs have been extended for the new interface:

- Program NAT-DATA displays all incoming data, also including binary data in a hexadecimal-editor like view.
- Program NAT-DIR has been extended with a version parameter to check which library (SYSWEB or SYSWEB3) has been used during compile time.
- Program NAT-DIRR has been extended to show links also for binary data (for example, **.gif*) in order to display the data using program NAT-RES.
- Program NAT-RES has been extended to display binary data files (for example, **.gif*).
- Program NAT-RES can be used to upload data to the Natural resource directory (this new feature has to be activated manually in the program source).

Changes to SYSWEB

The information listed below applies to the old `SYSWEB` library.

Internal Usage of Dynamic Variables and New Natural Features

- Internal usage of dynamic variables instead of (A1/1:v arrays).
- Existing programs need not be recataloged.
- Use new Translate commands internally.
- The predefined maximum return page is now set to 100000 (can be changed more easily).
- To change the maximum size of the return page, only the PDA W3LIMITS needs to be changed. It is no longer necessary to change w3context, too. There is no need to change the definition of t he (A1/1:v) array and its redefinition) up to the maximum size that might be needed even if the majority of pages is smaller.
- Using dynamic variables decreases the data consumption of the application.

- Easier handling of environment variables greater then (A250). These variables need not be read in parts using redefined long alpha with (A1/1:v) array.
- Use of long constant values. A complete HTML page now can be defined within an init clause of the program, or an simple assign statement.

W3* Programs for Dynamic Variables and Dynamic Arrays of Dynamic Variables

- All interfaces of programs that had been defined as (A250) By Value have been changed to (A) Dynamic By Value. User programs need not be changed or recataloged to work with the changed interface.
- New W3*DYNAMIC calls have been introduced. The existing calls look for trailing blanks (because of static length strings) and remove trailing blanks. The new calls take the compile dynamic variable up to *length(), without additional checking.
- Use of constant values greater the (A253).

Use #HTTP_NEWLINE together with W3HTML

If more then one line should be produced and converted to HTML, it had been necessary to call W3HTMLLINE for every page. The changed Web Interface now can handle lines that had been compressed with ##HTTP_NEWLINE.

Old:

```
perform w3text "<pre>"
perform w3htmlline "if a > b then"
perform w3htmlline " write ’result is > then old’ "
perform w3htmlline "else"
perform w3htmlline " write ’result is < then old’ "
perform w3htmlline "end-if"
perform w3text "</pre>"
```

New:

```
perform w3text "<pre>"
COMPRESS "if a > b then" ##HTTP_NEWLINE
" write ’result is > then old’ " ##HTTP_NEWLINE
"else" ##HTTP_NEWLINE
" write ’result is < then old’ " ##HTTP_NEWLINE
"end-if" ##HTTP_NEWLINE
into W3DYN LEAVING NO
perform w3html w3dyn
perform w3text "</pre>"
```

HTML2NAT Generates Long Constant Values

To reduce calls to W3TEXT, new generated programs will use constant values longer then (A253), if possible.

Call XSLT Processor before W3END

After a complete XML document is generated, call an XSLT processor to convert your XML. See e3xslt in the library `SYSWEB`. Or generate a HTML template and add your converted xml to a specific place. See e3xslt3 in the library `SYSWEB`.

Template Processing of Documents Saved at Resource Directory

It is possible to load templates defined with an HTML editor from the Natural resource directory. Your specific character strings like $NWW$ can be searched and replaced with the actual value. See e3templ in the library `SYSWEB`.

XML Toolkit

All XML Toolkit-revelant programs, dialogs etc. are now delivered in the library `SYSXTK` (no longer in the library `SYSEXXT`).

A **File Selection** dialog has been implemented. File names, whenever possible, can be selected from a selection list.

XML schemas containing `<import>` or `<include>` elements can now be used for generation. For the schema "Location", only local saved files can be used.

`<any>` and `<anyAttribute>` elements can now be specified. The content of an `<any>` or `<anyAttribute>` element is saved unparsed as a dynamic variable.

Conditional sections inside a DTD are now supported.

The element type `ANY` is now supported (equal to the `<any>` XML schema construct). The content of an `ANY` element is saved unparsed as a dynamic variable.

For information on the current limitations, see *Considerations and Limitations* in the *XML Toolkit* section of the *Web Technology* documentation.

Natural Connection

Support of Long and Dynamic Variables

The new feature now enables you to upload or download long and dynamic variables in Natural Connection work files.

Work File Format and Attributes

Below are the restrictions that apply to the use of work file attributes:

Accessing PC work files is restricted to a fixed record length of 1073741823 bytes or 32767 bytes when using the statement `WRITE WORK FILE VARIABLE`. Depending on the Entire Connection version installed on the PC, additional restrictions may apply as described below.

Maximum File Transfer Record Length for Natural Connection

The maximum record length supported for file transfer is 1 GB - 1 byte = 1073741823 bytes. For more information, see *Maximum File Transfer Record Length for Natural Connection* in the *Natural Connection* documentation.

Miscellaneous

Single-Fetch as Default Fetch Mode for Adabas Databases

By default, Natural uses single-fetch to retrieve data from Adabas databases. This default processing mode can be set to single-fetch or multi-fetch using the profile parameter `MFSET`. The setting can be changed on statement level if required.

For further information, see *Accessing Data in an Adabas Database* in the *Programming Guide*. You can also consult the statements `READ`, `FIND` and `HISTOGRAM` for more information.

Adabas Access

Multiple-value fields and periodic groups with more than 191 occurrences are now supported as of Adabas Version 5. The size of one Adabas record is still limited to 64 KB.

Characters in a Mask

The JJJ character (Julian date) may now be used within a mask definition. This enhancement is provided for compatibility with Natural for Mainframes. See *Characters in a Mask* in the *Programming Guide*.

Programs in Sample Libraries `SYSEXPG` and `SYSEXSYN`

The library `SYSEXRM` has been renamed to `SYSEXSYN`.

The sample programs included in the Natural documentation that are located in the libraries `SYSEXPG` and `SYSEXSYN` have been completely revised, enhanced and renamed.

Application Programming Interfaces `USR*` in Library `SYSEXT`

The `USR*` programs delivered in the library `SYSEXT` now run in a special mode. As a result, the `USR*` programs need no longer set further steplibs to execute related Natural objects for processing. This reduces the impact on the Natural buffer pool search logic and improves performance significantly if application programming interfaces are used extensively within user-written applications.

Buffer Pool

The `SHUTDOWN` command of the `NATBPMON` utility has been enhanced to quickly shut down a buffer pool. It is now possible to specify the `FORCE` option with an optional parameter. If `NATBPMON SHUTDOWN FORCE` *`grace-period`* is executed with super user rights, the terminate signal `SIGTERM` is sent to all active Natural sessions. After the expiry of the seconds defined with the *grace-period* parameter, the IPC resources held by the buffer pool are removed from the system. This command only has effect if executed with super user rights; without these rights, `NATBPMON` will reject its execution. See *NATBPMON Commands* in the *Operations* documentation.

Printing

When you print an object, the database ID and file number are now included on the printout.

NCF Format Files

The content of the NCF format files which are created for the work file type Entire Connection has changed. Now the information fields that were empty before are also filled with the work file processing information. The content of the information field **File name** has changed; the file name and path are now included in the information field. Since Unicode is now supported for NCD files, a new information field is available in the NCF file, containing code page information.