natural

Version 4.1.2 for Mainframes | Parameter Reference

SOFTWARE AG

This document applies to Natural Version 4.1.2 for Mainframes and to all subsequent releases.

Specifications contained herein are subject to change and these changes will be reported in subsequent release notes or new editions.

Table of Contents

Parameter Reference

- Profile Parameters
- Session Parameters

General information on using parameters can be found in the platform-specific Operations documentation, see Profile Parameter Usage

For information on Natural parameter macros, refer to Using Optional Macros in a Natural Parameter Module in the Natural Operations for Mainframes documentation.

Profile Parameters - Overview

Profile Parameters - Overview

The following topics are covered:

- Introduction to Profile Parameters
- Profile Parameters in Alphabetical Order (Mainframes)

Introduction to Profile Parameters

The following topics are covered:

- Profile Parameter Usage
- Profile Parameter Descriptions

Profile Parameter Usage

As the use of the Natural profile parameters is different on the platforms supported by Natural, general information about parameter usage is included in the corresponding platform-specific documentation.

Natural for Mainframes

See the following documents in the Natural Operations for Mainframes documentation:

Profile Parameter Usage - Overview

- Natural Parameter Hierarchy
- Assignment of Parameter Values
- Profile Parameters Grouped by Function
- Using a Natural Parameter Module

Natural for Windows

See the following documents in the Natural Operations for Windows documentation:

Profile Parameter Usage

- Creating Parameter Files
- Profile Parameters Grouped by Function
- Dynamic Assignment of Parameter Values
- Runtime Assignment of Parameter Values

Natural for UNIX

See the following documents in the Natural Operations for UNIX documentation:

Profile Parameters

- General Information
- Creating Parameter Files
- Profile Parameters - Overview
- Dynamic Assignment of Parameter Values
- Runtime Assignment of Parameter Values

Profile Parameter Descriptions

Unlike the Natural session parameters whose use is consistent on all platforms supported by Natural, many of the Natural profile parameters described in this document are not available on all platforms. The parameter descriptions indicate to which platforms they apply. For a list of Natural session parameters, see Session Parameters.

ACIVERS - Define ACI Version for Use with EntireX Broker ACI

ACIVERS - Define ACI Version for Use with EntireX Broker ACI

For static specification, this parameter is available as a keyword subparameter of the NTRPC macro. For dynamic specification, this parameter is available as a subparameter of the profile parameter RPC.

It specifies the API version to be used with the EntireX Broker ACI. The broker stub in use must support the ACI version defined here. Please, refer to the EntireX documentation for the supported API versions. The setting of ACIVERS enables special features of the EntireX Broker, depending on the API version you are using. For details, see Setting Up an EntireX Broker Access in the Natural Remote Procedure Call documentation.

ACIVERS can be specified on both the client and the server side.

Possible settings	1 - 6	Single-digit number. The higher the version, the more features are available.
Default setting	2	API Version 2 is used.
Dynamic specification	YES	
Specification within session	NO	

For additional information on Natural RPC, see the Natural Remote Procedure Call documentation.

ACTPOLICY - Defines Default Activation Policy for DCOM Classes

ACTPOLICY - Defines Default Activation Policy for DCOM Classes

This Natural profile parameter is for Windows platforms only.

It defines the default activation policy for DCOM classes. It is evaluated if a Natural class is registered as a DCOM class with the REGISTER command.

Possible settings	ES	External single activation is performed.
	EM	External multiple activation is performed.
	IM	Internal multiple activation is performed.
Default setting	EM	
Dynamic specification	YES	
Specification within session	NO	

ADAMODE - Adabas Interface Mode

This Natural profile parameter is for mainframes only.

It controls the number of Adabas user queue elements (UQE) per Natural session.

Possible settings	0	Start Natural with one Adabas session without using the Adabas X48 communication (Natural in Natural Version 2.2 mode). Only one UQE is initialized and the ID of the UQE is built by the ADALN*x* module. All database calls - either sent by the nucleus, an application program or a 3GL program - are considered as the same Adabas user. Running under SYSPLEX is not possible. **Note:** If a non-zero value is specified for ADAMODE, but an Adabas subcomponent is unable to perform an Adabas X48 communication, an error message is issued and ADAMODE is set to 0.
	1	Start Natural with one user and Adabas X48 communication. Only one UQE is initialized, all nucleus and application database calls are submitted for the same UQE, however, calls sent by 3GL programs are excluded. Running under SYSPLEX is possible.
	2	Start Natural with two Adabas sessions, both using the Adabas X48 communication (Natural Version 2.3 mode). Two UQEs are generated at Natural session startup, and nucleus and application calls are running separate from each other. Database calls sent by 3GL programs are excluded from Natural transactions. Running under SYSPLEX is possible.
	3	Start Natural with one UQE using Adabas X48 communication for nucleus calls and one UQE without Adabas X48 communication for application calls. Nucleus and application calls are running separate from each other. Database calls sent by 3GL programs are included into Natural application transactions. Running under SYSPLEX is not possible.
Default setting	2	
Dynamic specification	YES	
Specification within session	NO	

Background Information:

With **Natural Version 2.2**, every Natural session is related to one Adabas user queue element (UQE). All calls to Adabas - sent by either the nucleus (e.g. to load Natural objects for execution) or any user application - appear to Adabas as the same user. The user identification in the UQE is automatically built by the ADALNx module and - depending on the environment (CICS, BS2000/OSD, TSO, etc.) - contains the task number, the terminal ID or any other unique item. Although the uniqueness of nucleus/application works fine in most cases, problems may arise under the following conditions:

When the nucleus and an application are starting update transactions (UDATE, STORE, DELETE) at the same time, an ET/BT command sent either by the nucleus or the application will confirm or roll back all transactions, no matter to whom they originally belong.

When a timeout situation (NAT3009) occurs, Natural has to inform the session user, but the logical user (nucleus/application) for whom the timeout was sent is not known.

With **Natural Version 2.3 and above**, a new logic to communicate (X48) with Adabas was introduced. This enables you to set up different users (UQEs) on Adabas for one Natural session. This feature separates Natural nucleus calls from application calls, and all transactions of the nucleus and the applications are running in their own logical environment. A timeout situation (NAT3009) received for nucleus calls is automatically handled by Natural and does not lead to an error message displayed to the user.
To allow Natural to run in a Sysplex environment, the mechanism previously used to build a UQE cannot be used any longer, because task number, terminal ID or whatever used by ADALNx to build a UQE ceases to be unique, since these items may change in a SYSPLEX environment on every terminal I/O statement. The X48 communication with Adabas is a prerequisite for Natural to run in a SYSPLEX environment.

This has the minor drawback that every Natural session is related to two different UQEs on Adabas; this means, the number of Adabas UQEs is doubled compared with Natural Version 2.2. This may cause problems if the number of UQEs is limited (refer to the Adabas parameter NU).

ADANAME - Name of Adabas Link Routine

This Natural profile parameter is for mainframes only, but does not apply for UTM and Com-plete.

It specifies the name of the Adabas link routine to be used.

Possible settings	1 - 8 characters	Valid module or entry name.
Default setting	ADABAS	
Dynamic specification	YES	
Specification within session	NO	

If the Adabas link routine is linked to the Natural parameter module (NATPARM) and its entry name is the same as the one specified by ADANAME in the parameter module, the linked routine will be used. If not, the specified link routine will be loaded dynamically. Thus, it is no longer necessary to statically link the Adabas link module to the Natural nucleus.

It is possible to run the same Natural nucleus with different Adabas link modules.

ADAPRM - REVIEW/DB Support

This Natural profile parameter is for mainframes only.

It is used to pass Natural session data to REVIEW/DB within the seventh Adabas buffer.

Possible settings	ON	Natural session data is passed. Set ADAPRM to ON if REVIEW/DB is installed.
	OFF	No Natural session data is passed.
Default setting	OFF	
Dynamic specification	YES	
Specification within session	NO	

ADASBV - Adabas Security by Setting

This Natural profile parameter is for mainframes only.

It can be used to prevent invalid results for accesses to Adabas files that are protected by "security-by-setting". When a file that is protected by "security-by-setting" is accessed, invalid results may be returned in some cases where no format buffer is generated and passed to Adabas.

Possible settings	ON	Natural session data is passed. It is recommended that you set ADASBV=ON if you access "security-by-setting"-protected Adabas files. A format buffer is then always passed to Adabas for a database access (even if this is a 2-byte dummy buffer), thus avoiding invalid results.
	OFF	No Natural session data is passed.
Default setting	OFF	
Dynamic specification	YES	
Specification within session	NO	

ASIZE - Entire System Server Auxiliary Buffer

This Natural profile parameter is for mainframes only.

It only applies if Entire System Server is installed.

It determines the size of the Entire System Server auxiliary buffer.

Alternatively, you can use the equivalent Natural profile parameter DS or macro NTDS (see Using Optional Macros in a Natural Parameter Module in the Natural Operations for Mainframes documentation) to specify the ASIZE value.

Possible settings	0 - 64	Buffer size in KB. If Entire System Server is to be used, this parameter **must** be set; see the Entire System Server documentation. If ASIZE=0 is specified or if the requested space is not available, the Entire System Server is not activated.
Default setting	0	
Dynamic specification	YES	
Specification within session	NO	

ASPSIZE (Internal Use)

This Natural profile parameter is for mainframes only.

It is reserved for internal use by Natural. Do not change its setting.

ASYNNAM - Output System ID for Asynchronous Processing

This Natural profile parameter is for mainframes only. It only applies to Natural under UTM.

For asynchronous processing between two Natural applications that are running under the TP monitor UTM, this parameter specifies the address of the synchronous application which is used by the asynchronous application to send messages to the synchronous application.

Possible settings	1 - 8 characters	Valid transaction name.
	blank	No asynchronous processing takes place.
Default setting	blank	
Dynamic specification	YES	
Specification within session	NO	

For further information on asynchronous processing under UTM, see **Asynchronous Transaction Processing** in the Natural TP Monitor Interfaces documentation.

ATTN - Attention Key Interrupt Support

This Natural profile parameter is for mainframes only.

It controls the use of the attention key for IBM SNA terminals. Pressing the attention key can interrupt Natural processing with an appropriate error message (NAT1016). The availability of an attention key depends on the environment and on the terminal type.

This functionality is also available for Natural in Batch under VSE/ESA.

Possible settings	ON	The attention key causes Natural processing to be interrupted.
	OFF	The attention key is ignored.
Default setting	ON	
Dynamic specification	YES	
Specification within session	NO	

AUTO - Automatic Logon

This Natural profile parameter is for:

- Mainframes
- UNIX and Windows

It causes an automatic logon to a specific library at the start of the Natural session.

Possible settings	ON	An automatic logon is executed at the start of the Natural session.
	OFF	No automatic logon is performed.
Default setting	OFF	
Dynamic specification	YES	
Specification within session	NO	

The setting contained in the system variable *INIT-USER is used as the user ID for the logon.

Note:
If used with Natural Security, AUTO=ON disables logons with another user ID (see the Natural Security documentation for further information).

AUTOREGISTER - Automatic Registry Update

This Natural profile parameter is for Windows platforms only.

It determines whether Natural classes are to be registered as DCOM classes each time they are cataloged. Alternatively, Natural classes can be registered manually with the REGISTER command. This parameter is used by NaturalX.

Possible settings	ON	Registry will be updated automatically.
	OFF	Registry will not be updated automatically.
Default setting	OFF	
Dynamic specification	YES	
Specification within session	NO	

AUTORPC - Automatic Natural RPC Execution

For static specification, this parameter is available as a keyword subparameter of the NTRPC macro. For dynamic specification, this parameter is available as a subparameter of the profile parameter RPC.

This parameter determines whether or not Natural RPC will automatically try to execute a subprogram remotely (on the server side) which was not found locally (on the client side). For details see Stubs and Automatic RPC in the Natural Remote Procedure Call documentation.

AUTORPC is specified on the client side only.

Possible settings	ON	Natural RPC will automatically try to execute it remotely.
	OFF	Natural RPC will not automatically try to execute it remotely. If AUTORPC=OFF, you can execute CALLNATs remotely using stubs.
Default setting	OFF	
Dynamic specification	YES	
Specification within session	YES	At runtime, this value can be overwritten using the Parameter Maintenance function of the SYSRPC utility.

For additional information on Natural RPC, see the Natural Remote Procedure Call documentation.

BATCH - Batch Mode Simulation

This Natural profile parameter is for UNIX and Windows platforms only.

It sets the system variable *DEVICE to "BATCH" when Natural is started.

Possible settings	ON	When Natural is started with profile parameter BATCH set, error messages are not displayed, but written to a log file. The log file is named "natbatch.log" and is located in the Natural binary directory.
Default setting	OFF	Error messages are displayed but not written to a log file.
Dynamic specification	YES	This parameter can only be specified dynamically.
Specification within session	NO	

To run Natural in real batch mode, use the parameter BATCHMODE instead of BATCH.

BATCHMODE - Real Batch Mode

This Natural profile parameter is for UNIX and Windows platforms and applies to batch mode only.

It enables real batch mode and sets the system variable *DEVICE to "BATCH" when Natural is started.

Possible settings	ON	Natural will run in batch mode.
Default setting	OFF	Natural will run in interactive mode, or in batch mode simulation.
Dynamic specification	YES	The parameter can only be specified dynamically.
Specification within session	NO	

BMBLANK - Display Trailing Blanks

This Natural profile parameter is for UNIX and Windows platforms and applies to batch mode only.

It is used to control the display of trailing blanks in the batch output file CMPRINT.

This parameter applies only if the parameter BMSIM is set to MF. Trailing blanks are generated automatically if BMSIM is set to MF. BMBLANK has no effect if BMSIM is set to OS or VM.

Possible settings	ON	Trailing blanks are written to CMPRINT.
	OFF	No trailing blanks are written to CMPRINT.
Default setting	ON	
Dynamic specification	YES	
Specification within session	NO	

BMCONTROL - Display Control Characters

This Natural profile parameter is for UNIX and Windows platforms and applies to batch mode only.

It controls the output of control characters (such as form feed and line feed) in the batch output file CMPRINT.

Possible settings	ON	Control characters will be written to CMPRINT.
	OFF	No control characters will be written to CMPRINT.
Default setting	ON	
Dynamic specification	YES	
Specification within session	NO	

BMFRAME - Window Frame Characters

This Natural profile parameter is for UNIX and Windows platforms and applies to batch mode only.

With this parameter you can define window frame characters that will be written to the batch output file CMPRINT. Specify a sequence of 6 characters.

Possible settings	6 characters	Character Position:	Displays:
		1 2 3 4 5 6	Horizontal bar Vertical bar Upper-left corner Upper-right corner Lower-left corner Lower-right corner
Default setting	"-!++++"		
Dynamic specification	YES		
Specification within session	NO		

Example:

To define the following frame you have to specify BFRAME=123456.

```
3111111111114
2           2
2           2
5111111111116
```


BMSIM - Similar Batch Mode Output

This Natural profile parameter is for UNIX and Windows platforms and applies to batch mode only.

It is used for the general appearance description of the batch mode output file CMPRINT.

<table>
<tr><td rowspan="3">Possible settings</td><td>MF</td><td>Forces output similar to Natural for Mainframe Version 2.3. Each line in CMPRINT is filled with trailing blanks.
A control character appears at the beginning of each line of CMPRINT. The control character codes are similar to the IBM control character option ASA.
The following control character codes are used:

Control Code Interpretation
blank Normal output line without control characters
0 Insert one empty line
- Insert two empty lines
+ Print this line twice (bold printing)</td></tr>
<tr><td>VM</td><td>Forces output similar to Natural for OpenVMS Version 2.1.
The following control character codes are used:

Control Code Interpretation
\n Line feed before printing this line
\f Form feed before printing this line

The FIN command will be generated if it is not specified in the batch command file CMSYNIN.</td></tr>
<tr><td>OS</td><td>Forces output similar to Natural for UNIX or OpenVMS.
The following control character codes are used:

Control Code Interpretation
\n Line feed before printing this line
\f Form feed before printing this line</td></tr>
<tr><td>Default setting</td><td>OS</td><td></td></tr>
<tr><td>Dynamic specification</td><td>YES</td><td></td></tr>
<tr><td>Specification within session</td><td>NO</td><td></td></tr>
</table>

BMTIME - Display Process Time

This Natural profile parameter is for UNIX and Windows platforms and applies to batch mode only.

It is used to display the elapsed and used CPU time consumed by the Natural process. This output will be written to the end of the batch output file CMPRINT.

Possible settings	ON	The elapsed and used CPU time is written to the end of the batch output file.
	OFF	The output is not written to the batch output file.
Default setting	OFF	
Dynamic specification	YES	
Specification within session	NO	

The time format is as follows:

DDDxHH:*II*:*SS*.*UU*

where:

- *DDD* is the number of days (at maximum 999)
- *x* is " " (blank) if *DDD* is less or equal to 999, or "+" (plus sign) if *DDD* is greater than 999
- *HH* is the number of hours
- *II* is the number of minutes
- *SS* is the number of seconds
- *UU* is the number of hundredths of seconds

Example:

```
Used CPU time:   0 00:00:00.56
Elapsed time:    0 00:00:16.20
```


BMTITLE - Display Window Title

This Natural profile parameter is for UNIX and Windows platforms and applies to batch mode only.

It is used to control the displaying of window titles in the batch output file CMPRINT.

Possible settings	ON	A window title will be displayed in CMPRINT.
	OFF	No window title will be displayed in CMPRINT.
Default setting	ON	
Dynamic specification	YES	
Specification within session	NO	

BMVERSION - Display Natural Version

This Natural profile parameter is for UNIX and Windows platforms only.

It is used to control the display of the Natural version including the startup and termination time.

Possible settings	ON	The Natural version and startup time are written to the very first line of the batch output file CMPRINT, the termination time is written at the end of CMPRINT.
	OFF	The Natural version and startup time are not written to CMPRINT.
Default setting	OFF	
Dynamic specification	YES	
Specification within session	NO	

Example:

```
Natural V 6.1.1  Software AG 2003 started on 06-Oct-2003 at 08:14:36.34
 ...
Natural terminated on 06-Oct-2003 at 08:14:36.38
```


BPC64 - Buffer Pool Cache Storage Type

This Natural profile parameter is for mainframes under z/OS only (not for Com-plete).

It specifies the type of storage for the local Natural buffer pool. It corresponds to the C64 subparameter of the BPI profile parameter or the NTBPI macro.

Possible settings	ON	This indicates that virtual storage above the 2GB line is to be used for the buffer pool cache.
	OFF	This indicates that a data space is to be used for the buffer pool cache.
Default setting	OFF	
Dynamic specification	YES	This parameter can only be specified dynamically.
Specification within session	NO	

The BPC64 parameter only applies to the primary Natural buffer pool (TYPE=NAT, SEQ=0). In the case of a global buffer pool, it is ignored. If there is a primary buffer pool with SEQ=0 in NATPARM, only the C64 setting of this buffer pool is updated.

In multi-user environments (for example, under CICS), the BPC64 profile parameter only affects the very first Natural session which initializes the local buffer pool.

Internally, the BPC64 specification is converted into the equivalent BPI specification.

Example:

`BPC64=ON` is converted into: `BPI=(TYPE=NAT,SEQ=0,C64=ON)`

For general information on the Natural Buffer Pool, see Natural Buffer Pool (in the Natural Operations for Mainframes documentation).

BPCSIZE - Cache Size for Natural Buffer Pool

This Natural profile parameter is for mainframes under OS/390 and VSE/ESA only (not for Com-plete and not for IMS/TM).

It specifies the size of the buffer pool cache (in KB) for the Natural buffer pool. It corresponds to the CSIZE subparameter of the BPI profile parameter or the NTBPI macro.

Possible settings	0	Size of the buffer pool cache in KB. If BPCSIZE=0 is set, no buffer pool cache is used.
	4 to 2097148	(that is, 4 KB - 2 GB) for cache in data space, that is, with C64=OFF.
	100 to 58720256	(that is, 56 GB) for cache "above the bar", that is, with C64=ON.
	Note: The specified value is rounded to the next 4 KB boundary for a data space cache and to the next 1 MB boundary for a memory object cache. If the value specified exceeds the possible maximum, the possible maximum value will be taken instead.	
Default setting	0	By default, no buffer pool cache is used.
Dynamic specification	YES	This parameter can only be specified dynamically.
Specification within session	NO	

The BPCSIZE parameter applies to the primary Natural buffer pool (TYPE=NAT, SEQ=0) only. In the case of a global buffer pool, it is ignored. If there is a primary buffer pool with SEQ=0 in NATPARM, only the CSIZE setting of this buffer pool is updated.

In multi-user environments (for example, under CICS), the BPCSIZE parameter affects the very first Natural session only, which initializes the local buffer pool.

The type of storage to be used for the buffer pool cache is determined by profile parameter BPC64 or subparameter C64 of profile parameter BPI or macro NTBPI.

Internally, the BPCSIZE specification is converted into the equivalent BPI specification.

Example:

`BPCSIZE=4000` is converted into: `BPI=(TYPE=NAT,SEQ=0,CSIZE=4000)`

For more information see Buffer Pool Cache (in the Natural Operations for Mainframes documentation).

BPI - Buffer Pool Initialization

This Natural profile parameter is for mainframes only.

It is used to assign buffer pools to a Natural session. It corresponds to the NTBPI macro in the parameter module NATPARM.

There are several types of buffer pools for different purposes. It is possible to define backup buffer pools (see examples below). If a buffer pool is unavailable, Natural tries to setup a backup buffer pool of the same type with the next higher sequence number.

Possible settings	See Keyword Subparameters, below.	Possible subparameter keywords: TYPE | SEQ | NAME | SIZE | CSIZE | LIST | TXTSIZE | METHOD | C64
Default setting	TYPE=NAT,SEQ=0,NAME=' ',SIZE=256, CSIZE=0,TXTSIZE=4,METHOD=S, C64=OFF	
Dynamic specification	YES	The parameter BPI can only be specified dynamically. In NATPARM, use the macro NTBPI.
Specification within session	NO	

The subparameters SIZE, CSIZE, TXTSIZE METHOD and C64 do not apply to global buffer pools. These subparameters are honored for the very first session only which initializes a local buffer pool.

The following topics are covered below:

- BPI Parameter Syntax
- NTBPI Macro Syntax
- Keyword Subparameters
- Examples of NTBPI Macros
- Examples of BPI Parameter

BPI Parameter Syntax

The BPI parameter is specified as follows:

```
BPI=(TYPE=type,SEQ=n,NAME=name,SIZE=nnn,LIST=name,TXTSIZE=n,CSIZE=nn,METHOD=x,C64=xx)
```

- To dynamically deactivate a buffer-pool definition, use the special value OFF as follows:
  `BPI=(TYPE=type,SEQ=n,OFF)`
- If OFF is used, it must be specified after TYPE and SEQ. OFF is not allowed for the macro NTBPI.
- If you use the BPI parameter to overwrite an existing buffer pool definition in the parameter module, you must specify new settings in all those subparameters which are to be changed; if you do not, the old settings will still be used.
  If, for example, you want to change from a global to a local buffer pool, you must specify: `NAME=' '`.
- If you use the BPI parameter to dynamically add a new backup buffer pool definition, you must specify a sequence number (SEQ) for it.
  If you omit the SEQ specification, the definition of the primary buffer pool (SEQ=0) will be overwritten.

- The NAME, SIZE, LIST, TXTSIZE, CSIZE, METHOD and C64 specifications for the primary buffer pool (SEQ=0) can also be set dynamically with the profile parameters BPNAME, BPSIZE, BPLIST, BPTEXT, BPCSIZE, BPMETH and BPC64.

NTBPI Macro Syntax

The NTBPI macro is specified as follows:

```
      NTBPI TYPE=type,                                  *
            SEQ=n,                                      *
            NAME=name,                                  *
            CSIZE=nnn,                                  *
            SIZE=nnn,                                   *
            LIST=name,                                  *
            METHOD=x,                                   *
            C64=xx,                                     *
            TXTSIZE=n
```

Keyword Subparameters

TYPE | SEQ | NAME | SIZE | CSIZE | LIST | TXTSIZE | METHOD | C64

TYPE - Type of Buffer Pool

Determines the type of the buffer pool. Possible types are:

NAT	Natural buffer pool (this is the default). For general information on the Natural buffer pool, see Natural Buffer Pool (in the Natural Operations for Mainframes documentation). *
DLI	DL/I buffer pool. *
EDIT	Software AG Editor buffer pool. Alternatively, an editor auxiliary buffer pool can be defined per session, see also the profile parameter EDPSIZE.
SORT	Sort buffer pool. *
MON	Buffer pool for Monitor function of SYSTP utility.
SWAP	Buffer pool to hold the Natural CICS swap pool.

* **Note:**
Buffer pools of the types NAT, DL/I or SORT can be managed with the utility SYSBPM.

SEQ - Sequence Number of Buffer Pool

Determines the sequence number *n* of the buffer pool.

The buffer pool defined with the lowest sequence number is called primary buffer pool. For every buffer pool type, except TYPE=SWAP, you can define one primary buffer pool and one or more backup buffer pools; that is, alternative buffer pools (of the same type, but with a different sequence number) which will be used if the primary buffer pool is not available at session initialization or cannot be allocated.

Buffer pools of the same type are sorted in order of sequence numbers (should two pools of the same type have the same sequence number, they will be sorted in the order in which they are specified). If a requested buffer pool is not available, the buffer pool of the same type with the next higher sequence number will be used instead. If that one is not available either, the one with the next higher number will be used, etc.

Possible values	0 to 9
Default value	0

NAME - Name of Global Buffer Pool

Applies only to global buffer pools and pools of TYPE=SWAP only.

Determines the *name* of the global buffer pool. For a local buffer pool, the *name* is blank. For TYPE=SWAP, name is the swap pool name which is the key of the associated swap pool definitions in the Natural system file FNAT or FUSER, see parameter SWPINIT (in the Natural Operations for Mainframes documentation, section Natural Swap Pool Initialization Control).

Possible values	1 to 8 characters
Default value	' '

The NAME specification can be overridden dynamically with the profile parameter BPNAME (with TYPE=NAT only).

Under Com-plete:
Under this TP monitor system, a global buffer pool is not possible, but only a local editor buffer.

SIZE - Size of Buffer Pool

Applies to local buffer pools only. Determines the size *nnn* of the buffer pool.

Possible values	100 to 2097151 (KB)
Default value	256

The SIZE specification can be overridden dynamically with the profile parameter BPSIZE (with TYPE=NAT only).

CSIZE - Size of the Local Buffer Pool Cache

Applies to local buffer pools of TYPE=NAT and TYPE=SWAP under OS/390 and VSE/ESA only (not for Com-plete).

It determines the size of the buffer pool cache in KB.

Possible values	0, 100 to 2097148 (that is, 2 GB - 4 KB) for cache in data space, that is, with C64=OFF.
	0, 100 to 58720256 (that is, 56 GB) for cache "above the bar", that is, with C64=ON.
	Note: The specified value is rounded to the next 4 KB boundary for a data space cache and to the next 1 MB boundary for a memory object cache. If the value specified exceeds the possible maximum value, the possible maximum value will be taken instead.
Default value	0 (that is, no buffer pool cache is used).

For more information see Buffer Pool Cache.

The CSIZE specification can be overridden dynamically with the profile parameter BPCSIZE (with TYPE=NAT only). To determine the type of storage for the buffer pool cache, subparameter C64 can be used.

LIST - Name of Preload List to be Used

Applies only to local buffer pools of TYPE=NAT.

Determines the *name* of the preload list to be used for this buffer pool.

Possible values	1 to 8 characters
Default value	The default is blank (that is, no preload list is to be used).

For general information on preload lists, see Preload List. Preload lists are maintained with the SYSBPM utility as described in the section Debugging and Monitoring.

The LIST specification can be overridden dynamically with the profile parameter BPLIST.

TXTSIZE - Size of Buffer Pool Text Segments

Applies to local buffer pools of the following types:

- TYPE=NAT
- TYPE=SORT
- TYPE=DLI

Determines the size *n* (in KB) of the buffer pool text segments.

Possible values	1, 2, 4, 8, 12, 16 (KB)
Default value	4.

In multi-user environments (for example, under CICS), the TXTSIZE specification only affects the very first Natural session which initializes the local buffer pool.

Under BS2000/OSD: The parameters SEQ, NAME and SIZE are ignored.

The TXTSIZE specification can be overridden dynamically with the profile parameter BPTEXT (with TYPE=NAT only).

METHOD - Search Algorithm for Allocating Space in Buffer Pool

Applies to local buffer pools of TYPE=NAT only.

Determines the algorithm for allocating storage in the buffer pool.

Possible values	S	This indicates that a selection process is to be used for allocating storage. The selection process consists of browsing the whole buffer pool directory and comparing different entries in order to find the most suitable entry. This method was formerly known as Algorithm 1+2.
	N	This indicates that the next available unused or free space is to be used. The search for the next available space is done from a pointer to directory entries which moves in a wrap around fashion. This method may be used in combination with a buffer pool cache.
Default value	S	

The METHOD specification can be overridden dynamically by profile parameter BPMETH.

C64 - Type of Buffer Pool Cache Storage

Applies to local buffer pools of TYPE=NAT under z/OS only (not for Com-plete).

Determines the type of storage to be used for the buffer pool cache.

Possible values	ON	This indicates that a memory object "above the bar" (that is, in 64-bit memory) is to be used for the buffer pool cache. Note that C64=ON is only honored if the prerequisites are met, namely: • z/ architecture hardware, • operating system z/OS Version 1.2 or higher. If the prerequisites are not met, the default value is taken.
	OFF	This indicates that a data space is to be used for the buffer pool cache.
Default value	OFF	

A buffer pool cache is used only if BPI subparameter CSIZE or profile parameter BPCSIZE is set to a non-zero value. The C64 specification can be overridden dynamically by profile parameter BPC64.

Examples of NTBPI Macros

```
NTBPI TYPE=NAT,SEQ=0,NAME=NATBP1
NTBPI TYPE=NAT,SEQ=1,NAME=NATBP2
NTBPI TYPE=NAT,SEQ=2,SIZE=1000,METHOD=N
```

These examples define multiple Natural buffer pools. If the global buffer pool NATBP1 is not available, the global buffer pool NATBP2 will be used instead. If the latter is not available either, a local buffer pool with a size of 1000 KB will be used.

Examples of BPI Parameter

```
BPI=(NAME=' ',SIZE=2000,METHOD=N)
```

The primary buffer pool is replaced by a local buffer pool of 2000 KB. This definition is equivalent to:

```
BPNAME=' ',BPSIZE=2000,BPMETH=N
```



```
BPI=(SEQ=0,NAME=LBP1),BPI=(SEQ=1,NAME=LBP2),BPI=(SEQ=2,SIZE=500)
```

First, Natural tries to allocate a global Natural buffer pool with the name "LBP1"; if this buffer pool is not found, it tries to allocate "LBP2"; if this is not found, it allocates a local buffer pool with a size of 500 KB.

```
BPI=(SEQ=0,TYPE=EDITOR,NAME=LBPE1),BPI=(SEQ=1,TYPE=EDITOR,SIZE=500)
```

First, Natural tries to locate a global editor buffer pool with the name "LBPE1"; if this is not found, it allocates a local editor buffer pool with a size of 500 KB.

```
BPI=(TYPE=SWAP,SIZE=500,NAME=SWAPPOOL,CSIZE=2000)
```

A Natural local swap pool named "SWAPPOOL" having a size of 500 KB and a cache size of 2000 KB is allocated.

BPID - Specify Buffer Pool ID

This Natural profile parameter is for UNIX and Windows platforms only.

It specifies the name (ID) of the Natural buffer pool.

Possible settings	1 - 8 characters	Name of the Natural buffer pool.
Default setting	NATBP	
Dynamic specification	YES	
Specification within session	NO	

Note:
Do not delete the default buffer pool "NATBP", as it is possible that Natural may not function properly anymore.

BPLIST - Name of Preload List for Natural Buffer Pool

This Natural profile parameter is for:

- Mainframes
- UNIX

With BPLIST you can specify the name of a preload list to be used for the Natural buffer pool.

Possible settings	1-8 characters, or blank	Name of a preload list to be used for the Natural buffer pool. If BPLIST=' ' (blank) is set, no preload list is used.	
Default setting	blank		
Dynamic specification	YES	On mainframes:	This parameter can only be specified dynamically.
	NO	On UNIX:	No dynamic specification
Specification within session	NO		

For general information, see Natural Buffer Pool.

The parameter corresponds to the LIST specification of the BPI profile parameter or the NTBPI macro.

It only applies to the primary Natural buffer pool (TYPE=NAT, SEQ=0). If there is a primary buffer pool with SEQ=0 in NATPARM, only the LIST setting of this buffer pool is updated. Internally, the BPLIST specification is converted into an equivalent BPI specification.

Example:
`BPLIST=LIST3` is converted into: `BPI=(TYPE=NAT,SEQ=0,LIST=LIST3)`

BPMETH - Buffer Pool Space Search Algorithm

This Natural profile parameter is for mainframes only.

It specifies the search algorithm that is to be used for allocating storage in the Natural buffer pool. It corresponds to the METHOD subparameter of the BPI profile parameter or the NTBPI macro.

Possible settings	S	This indicates that a selection process is to be used for allocating storage. The selection process consists of browsing the whole buffer pool directory and comparing different entries in order to find the most suitable entry. This method was formerly known as Algorithm 1+2.
	N	This indicates that the next available unused or free space is to be used. The search for the next available space is done from a pointer to directory entries which moves in a wrap around fashion. This method may be used in combination with a buffer pool cache.
Default setting	S	
Dynamic specification	YES	This parameter can only be specified dynamically.
Specification within session	NO	

The BPMETH parameter only applies to the primary Natural buffer pool (TYPE=NAT, SEQ=0). In the case of a global buffer pool, it is ignored. If there is a primary buffer pool with SEQ=0 in NATPARM, only the METHOD setting of this buffer pool is updated.

In multi-user environments (for example, under CICS), the BPMETH profile parameter only affects the very first Natural session which initializes the local buffer pool.

Internally, the BPMETH specification is converted into the equivalent BPI specification.

Example:

`BPMETH=S` is converted into: `BPI=(TYPE=NAT,SEQ=0,METHOD=S)`

For general information on the Natural buffer pool, see Natural Buffer Pool (in the Natural Operations for Mainframes documentation).

BPNAME - Name of Natural Global Buffer Pool

This Natural profile parameter is for:

- Mainframes
- UNIX

With this parameter, you can specify the name of the Natural global buffer pool.

Possible settings	1 - 8 characters, or blank	Name of the Natural global buffer pool. If BPNAME=' ' (blank) is set, a **local** Natural buffer pool is used.
Default setting	blank	
Dynamic specification	YES	This parameter can only be specified dynamically.
Specification within session	NO	

For general information, see Natural Global Buffer Pool (in the Natural Operations for Mainframes documentation).

This parameter can only be specified dynamically. It corresponds to the NAME specification of the BPI profile parameter or the NTBPI macro respectively.

The BPNAME profile parameter only applies to the primary Natural global buffer pool (TYPE=NAT, SEQ=0).If there is a primary buffer pool with SEQ=0 in NATPARM, only the NAME setting of this buffer pool is updated. Internally, the BPNAME specification is converted into an equivalent BPI specification.

Example:
`BPNAME=GBP1` is converted into: `BPI=(TYPE=NAT,SEQ=0,NAME=GBP1)`

BPNLE - Number of Directory Entries

This Natural profile parameter is for UNIX and Windows platforms only.

It determines the number of directory entries. This value is used by the NATBPSRV server during the start of the buffer pool. If BPNLE has reached the maximum value, this value is automatically increased.

Possible settings	10 - 9999	Number of directory entries
Default setting	10	
Dynamic specification	NO	
Specification within session	NO	

BPPROP - Global Buffer Pool Propagation

This Natural profile parameter is for mainframes (only applies under OS/390 and BS2000/OSD).

It controls the propagation of changes to an object in a buffer pool. If a modification occurs affecting a Natural object residing in one (global or local) buffer pool, this modification can be propagated to other global buffer pools - this will ensure the consistency of the buffer pools.

Possible settings	OFF	Changes are not propagated to any other global buffer pool. **Note for OS/390:** Any setting other than OFF requires that the Authorized-Services Manager is active.
	GLOBAL	Changes are propagated to all other global buffer pools. **In an OS/390 Sysplex environment:** The changes are only propagated within the current OS/390 image. (*)
	PLEX	Changes are propagated to all other global buffer pools of the same name within the entire OS/390 Sysplex. (*)
	GPLEX	Changes are propagated to all other global buffer pools within the entire OS/390 Sysplex. (*) **Note for BS2000/OSD:** The setting GPLEX has the same effect as GLOBAL.
Default setting	OFF	
Dynamic specification	YES	
Specification within session	NO	

*** Under OS/390:**
The propagation is always restricted to the Natural subsystem in which the change has occurred; that is, the scope of the propagation, as set with the BPPROP parameter, applies only within that subsystem, but not to other subsystems. For details, see Natural Subsystem (in the Natural Operations for Mainframes documentation).

For further information on the propagation, see Natural Global Buffer Pool (in the Natural Operations for Mainframes documentation).

BPSFI - Object Search First in Buffer Pool

This Natural profile parameter is for:

- Mainframes
- UNIX and Windows

With this parameter, you determine the sequence in which a requested object that is to be executed is searched for in the buffer pool and in the system file(s). You can choose between two search sequences:

Possible settings	ON	Search Sequence 1 is used (search buffer pool first for all libraries, then the system file(s)). Natural looks for the object in the following sequence until it is found: 1. in the buffer pool, first in the current library, then in one steplib after another, then in the two SYSTEM libraries; 2. in the system file(s), first in the current library, then in one steplib after another, then in the two SYSTEM libraries. For performance reasons, it is recommended that you set BPSFI=ON in production environments.
	OFF	Search Sequence 2 is used (alternating search in buffer pool and system file(s) for each library). Natural looks for the object in the following sequence until it is found: 1. in the current library, first in the buffer pool, then in the system file(s); 2. in one steplib after another, first in the buffer pool, then in the system file(s) for each steplib; 3. in the two SYSTEM libraries, first in the buffer pool, then in the system file(s) for each library. BPSFI=OFF is recommended in development environments to always get the most current object from your own current library.
Default setting	OFF	
Dynamic specification	YES	
Specification within session	NO	

BPSIZE - Size of Natural Local Buffer Pool

This Natural profile parameter is for:

- Mainframes(except Com-plete and BS2000/OSD, see notes below)
- UNIX and Windows

It specifies the size of the Natural local buffer pool. It corresponds to the SIZE specification of the BPI profile parameter or the NTBPI macro.

Possible settings	100 - 2097151	Size of the Natural local buffer pool in KB.
Default setting	256	
Dynamic specification	YES	This parameter can only be specified dynamically.
Specification within session	NO	

BPSIZE only applies to the primary Natural local buffer pool (TYPE=NAT, SEQ=0). For a global buffer pool, it is ignored. If there is a primary buffer pool with SEQ=0 in NATPARM, only the SIZE setting of this buffer pool is updated.

In multi-user environments (for example, under CICS), the BPSIZE parameter only affects the very first Natural session, which initializes the local buffer pool.

Internally, the BPSIZE specification is converted into an equivalent BPI specification.

Example:
`BPSIZE=1500` is converted into: `BPI=(TYPE=NAT,SEQ=0,SIZE=1500)`

For general information, see Natural Buffer Pool (in the Natural Operations for Mainframes documentation).

Notes:

- **Under Com-plete**, the size of a local buffer pool is set as described in the Natural Installation Guide for Mainframes.
- **Under BS2000/OSD**, the size of a local buffer pool is specified with the parameter SIZE of the ADDON macro.

BPTEXT - Size of Text Segments in Natural Buffer Pool

This Natural profile parameter is for mainframes only.

It specifies the size of the segments into which the text pool area of the Natural buffer pool is divided. It corresponds to the TXTSIZE specification of the BPI profile parameter or the NTBPI macro.

Possible settings	1, 2, 4	Size of segments in KB.
Default setting	4	
Dynamic specification	YES	This parameter can only be specified dynamically.
Specification within session	NO	

The BPTEXT parameter only applies to the primary Natural buffer pool (TYPE=NAT, SEQ=0). In the case of a global buffer pool, it is ignored. If there is a primary buffer pool with SEQ=0 in NATPARM, only the TXTSIZE setting of this buffer pool is updated.

In multi-user environments (for example, under CICS), the BPTEXT parameter only affects the very first Natural session, which initializes the local buffer pool.

Internally, the BPTEXT specification is converted into an equivalent BPI specification.

Example:
`BPTEXT=4` is converted into: `BPI=(TYPE=NAT,SEQ=0,TXTSIZE=4)`

For general information on the Natural Buffer pool, see Natural Buffer Pool (in the Natural Operations for Mainframes documentation).

BSIZE - Size of EntireX Broker Buffer

BSIZE - Size of EntireX Broker Buffer

This Natural profile parameter is for mainframes only.

It only applies if EntireX Broker is installed.

Alternatively, you can use the equivalent Natural profile parameter DS or macro NTDS (see Using Optional Macros in a Natural Parameter Module in the Natural Operations for Mainframes documentation) to specify the BSIZE value.

Currently, if EntireX Broker is used, EntireX Broker specifies the buffer size automatically.

Possible settings	0 - 64	Buffer size in KB.
Default setting	0	
Dynamic specification	YES	
Specification within session	NO	

CANCEL - Session Cancellation with Dump

This Natural profile parameter is for mainframes only.

With this parameter, you can specify a character string that will cause the Natural session to be terminated with a dump. This may be useful for debugging purposes.

Possible settings	1 to 8 characters	When you enter this character string in any input field within your Natural session (beginning in the first input field), the session will be terminated immediately and a dump will be produced.
Default setting	*CANCEL	
Dynamic specification	YES	
Specification within session	NO	

CC - Error Processing in Batch Mode

This parameter is available

- on mainframes as a Natural profile and session parameter,
- on UNIX and Windows as a Natural profile parameter.

It determines the action to be taken if an error is detected during the compilation/execution of a Natural program in batch mode. It only applies in batch mode.

It does not apply if user-written error-handling routines are used.

Within a Natural session, the profile parameter CC can be overridden by the session parameter CC.

Possible settings	ON	Natural flushes the input data stream for the batch input files CMSYNIN and CMOBJIN until a line containing "%%" in the first two positions is encountered or until an end-of-file condition is detected. If more data are available in the input stream, Natural resumes reading after the line containing "%%".	
	OFF	Natural attempts to process the next program (or command) in the input stream.	
Default setting	OFF		
Dynamic specification	YES		
Specification within session	YES	Applicable Statements:	SET GLOBALS
		Applicable Command:	GLOBALS

When a Natural session terminates, Return Code 4 is passed to the invoking program with Register 15 if an error is detected (regardless of the CC setting).

CCTAB - Printer Escape Sequence Definition

This Natural profile parameter for mainframes only.

It is used to set up a table of printer-control sequences, which is used for printing additional reports and hardcopies. It corresponds to the NTCCTAB macro in the Natural parameter module NATPARM.

- It is possible to either translate Natural field attributes into escape sequences or specify special characters to be translated into escape sequences.
- In addition, strings can be specified which are always sent as the first output record after an open operation or as the last output record before a close operation.
- This means that by using the right profile name, you can activate your printout either in portrait mode or in landscape. Then you can use all print features of this device by using simple attributes in Natural. This makes even bar-code printing or double-height printing possible.
- CCTAB defines tables which are used to recognize special characters in output fields and replace them with the defined control sequences. The parameter also defines the Natural attributes which are used to insert the defined control sequences.

Possible settings	See CCTAB Parameter Syntax below.	
Default setting	As specified within the macro NTCCTAB in NATCONFG.	
Dynamic specification	YES	This parameter can only be specified dynamically. In the Natural parameter module NATPARM, the macro NTCCTAB must be used instead.
Specification within session	NO	

The following topics are covered below:

- CCTAB Parameter Syntax
- NTCCTAB Macro Syntax
- String Syntax for CODE, CS, CCS or CSE
- Proportional Fonts
- Examples of NTCCTAB Macros
- Examples of CCTAB Parameter

CCTAB Parameter Syntax

For each profile, a separate CCTAB must be specified. The CCTAB parameter can be specified in three variants:

1st Variant

```
CCTAB=( name, OPN='xxxxx',CLS='yyyyy')
```

Where

name	is the name of the profile form, that is, the DEFINE PRINTER (*n*) OUTPUT '*nnnnn*' PROFILE '*name*', which is required and which has a maximum length of 8 bytes.
OPN='*xxxxx***'**	is optional and defines a data string (up to 250 bytes) which is sent to the printer with each open operation.
CLS='*yyyyy***'**	is optional and defines a data string (up to 250 bytes) which is sent to the printer before each close operation.

OPN and CLS can be specified in any sequence.

2nd Variant

```
CCTAB=(name,CODE='n',CS='xxxx')
```

Where

CODE='*n***'**	is a character which is recognized by Natural once it appears in the output string.
CS='*xxxx***'**	is the string to replace the CODE character.

The CS subparameter must follow the CODE subparameter.

3rd Variant

```
CCTAB=(name,ATR=nnnn,CSS='xxxx',CSE='yyyy')
```

Where

ATR='*nnnn***'**	is the Natural internal field attribute. The name is defined by the macro NAMATR.
CSS='*xxxx***'**	is the string (up to 20 bytes) which is inserted before the field. CSS is mandatory.
CSE='*yyyy***'**	is the string (up to 20 bytes) which is inserted after the field. CSE is mandatory.

The CSS and CSE subparameters must follow the ATR subparameter.

NTCCTAB Macro Syntax

The NTCCTAB macro can be specified in three variants:

1st Variant

```
NTCCTAB name,OPN='xxxxx',CLS='yyyyy'
```

For details, refer to CCTAB Parameter Syntax, 1st Variant.

2nd Variant

```
NTCCTAB name,CODE='n',CS='xxxx'
```

For details, refer to CCTAB Parameter Syntax, 2nd Variant.

3rd Variant

```
NTCCTAB name,ATR=nnnn,CSS='xxxx',CSE='yyyy'
```

For details, refer to CCTAB Parameter Syntax, 3rd Variant.

String Syntax for CODE, CS, CSS or CSE

You specify each character either as the one-byte character itself (enclosed in apostrophes) or as the hexadecimal representation of that character.

Proportional Fonts

If you use proportional fonts, be sure to return to a fixed-spacing font before using tables where you need correct positioning.

Examples of NTCCTAB Macros

```
NTCCTAB DBCST
NTCCTAB CODE=OE,CS=400E
NTCCTAB CODE=OF,CS=0F40<
NTCCTAB ATR=P5DBCS,CSS=OE,CSE=OF

NTCCTAB TEST,OPN='FIRST LINE MESSAGE',CLS='LAST LINE MESSAGE'
NTCCTAB CODE='<',CS=' B(SOB'
NTCCTAB CODE='>',CS='B(S3B '
NTCCTAB CODE='(',CS=' B(S1S'
NTCCTAB CODE=')',CS='B(SOS '
NTCCTAB ATR=P2UL,CSS=' B&&DD',CSE='B&&D§'
NTCCTAB ATR=P2UL,CSS=405FF1C25084C4,CSE=5FF1C250847C
NTCCTAB ATR=P2ITAL,CSS=' B(S1S',CSE='B(SOS'
NTCCTAB ATR=P1HIGH,CSS=' B(S3B',CSE='B(SOB'
NTCCTAB ATR=P2RVID,CSS=' B(S-3B',CSE='B(SOB'
```

Examples of CCTAB Parameter

```
CCTAB=(DBCST,CODE=OE,CS=400E,CODE=OF,CS=0F40,ATR=P5DBCS,CSS=OE,CSE=OF)

CCTAB=(TEST,OPN='FIRST LINE MESSAGE',CLS='LAST LINE MESSAGE')
```


CDYNAM - Dynamic Loading of Non-Natural Programs

This Natural profile parameter is for:

- Mainframes
- UNIX and Windows

This parameter determines how many non-Natural programs can be loaded dynamically by Natural during the execution of a single Natural program.

Possible settings	0 - 1024	Determines the maximum number of non-Natural programs which can be loaded per Natural program. If CDYNAM=0, no dynamic loading of non-Natural programs will be performed by Natural.
Default setting	5	
Dynamic specification	YES	
Specification within session	NO	

CF - Character for Terminal Commands

This Natural profile and session parameter is for all platforms.

It specifies the control character for Natural terminal commands; that is, the character which is to be used as the first character of any terminal command.

Within a Natural session, the profile parameter CF can be overridden by the session parameter CF.

Possible settings	Any special character	A terminal command must begin with the character specified. The character specified with the CF parameter • must not be the same as the one specified with the HI parameter (help character) or IA parameter (input assign character). • should not be the same as the one specified with the DC parameter (decimal character) or ID parameter (input delimiter character). • In the map editor, the control character for terminal commands is always "%" (so as to avoid conflicts with delimiter characters used in maps), no matter which character is defined with the CF parameter.	
	OFF	No control character for terminal commands is available. Terminal commands issued with SET CONTROL statements, however, are still accepted.	
Default setting	%	A terminal command must begin with the character "%".	
Dynamic specification	YES		
Specification within session	YES	Applicable Statements:	SET GLOBALS
		Applicable Command:	GLOBALS

CFWSIZE (Internal Use)

This Natural profile parameter is for mainframes only.

This parameter is reserved for internal use by Natural. Do not change its setting.

CLEAR - Processing of CLEAR Key in NEXT Mode

This Natural profile parameter is for:

- Mainframes
- UNIX and Windows

It causes Natural to execute a specific Natural terminal command whenever CLEAR is pressed during program execution in NEXT mode.

Possible settings	any character	The default action can be overridden by supplying a character which, when appended to the terminal-command control character (as specified with the CF parameter), forms a valid Natural terminal command.
		
		Example:
		`CF=%`
		`CLEAR=R`
		Natural executes the terminal command "%R" when CLEAR is pressed in NEXT mode.

Default setting	%	By default, when the CLEAR key is pressed, Natural responds as if the user had entered the terminal command "%%".
Dynamic specification	YES	
Specification within session	NO	

CM - Command Mode

This Natural profile parameter is for:

- Mainframes
- UNIX and Windows

It can be used to suppress Natural command mode (NEXT and MORE).

Possible settings	ON	NEXT and MORE are available for command input.
	OFF	The Natural session will be terminated whenever NEXT is encountered; the MORE line will be write-protected (no input possible).
Default setting	ON	
Dynamic specification	NO	For UNIX and Windows.
	YES	For Mainframes.
Specification within session	NO	

CMOBJIN - Batch Input File for Natural INPUT Data

This Natural profile parameter is for UNIX and Windows. It applies to batch mode only.

It is used for data intended to be read by Natural INPUT statements. These types of data can alternatively be placed in the CMSYNIN file immediately following the relevant RUN or EXECUTE command. The number of characters actually processed is restricted to 512 characters per line.

Possible settings	any string	**Example for UNIX:**
		`CMOBJIN = $HOME/tmp/batch.inp`
		
		Example for Windows:
		`CMOBJIN=C:\tmp\data.txt`

Default setting	none	
Dynamic specification	YES	
Specification within session	NO	

If the setting for the profile parameter CMSYNIN is equal to the setting of CMOBJIN, Natural reads input from CMSYNIN.

If an error occurs, Natural reacts in accordance with the setting of the profile/session parameter CC.

CMPO - Compilation Options

This Natural profile parameter is for mainframes only.

It can be used, at session start, to specify dynamically or to override the same options which you can specify statically with the NTCMPO macro in the parameter module or, during an active session, with the COMPOPT system command.

Possible settings	See COMPOPT system command	
Default setting	See COMPOPT system command	
Dynamic specification	YES	This parameter can only be specified dynamically. In the Natural parameter module NATPARM, the macro NTCMPO must be used instead.
Specification within session	NO	

The following topics are covered below:

- CMPO Parameter Syntax
- NTCMPO Macro Syntax
- Keyword Subparameters
- Example of CMPO Parameter
- Example of NTCMPO Macro

CMPO Parameter Syntax

The parameter syntax of CMPO is as follows:

```
CMPO=(keyword-subparameter=value,...)
```

NTCMPO Macro Syntax

The syntax of the NTCMPO macro in the Natural parameter module is as follows:

```
NTCMPO keyword-subparameter=value,...
```

Each keyword subparameter can take the value "ON" or "OFF" (GFID can also take the value "VID"). See keyword subparameter descriptions below.

Keyword Subparameters

The following keyword subparameters are available:

DBSHORT | GFID | FINDMUN | KCHECK | LOWSRCE | MASKCME | NMOVE22 | PCHECK | PSIGNF | TQMARK | TSENABL | V31COMP

In the macro, the keyword subparameters can be specified in any sequence. For a description, refer to the system command COMPOPT (in the Natural System Command Reference documentation).

Example of CMPO Parameter

```
CMPO=(KCHECK=ON,PCHECK=ON)
```

Example of NTCMPO Macro

```
NTCMPO KCHECK=ON,PCHECK=ON
```


CMPRINT - Batch Output File

This Natural profile parameter is for UNIX and Windows. It applies to batch mode only.

It is used for the output report resulting from DISPLAY, PRINT and WRITE statement in a Natural program. In addition, Natural commands from CMSYNIN and INPUT data from CMOBJIN are written to CMPRINT.

Possible settings	any string	**Example for UNIX:**
		`CMPRINT = $HOME/tmp/batch.out`
		
		Example for Windows:
		`CMPRINT=C:\tmp\out.txt`

Default setting	none	
Dynamic specification	YES	
Specification within session	NO	

CMPRTnn - Additional Report

This Natural profile parameter is for UNIX and Windows. It applies to batch mode only.

It is used for additional reports referenced by any Natural program executed during the session. "*nn*" is a two digit decimal number in the range from 01 to 31 corresponding to the report number used in a DISPLAY, PRINT and WRITE statement.

Possible settings	any string	**Example for UNIX:**
		`CMPRT07 = $HOME/tmp/report7.txt`
		
		Example for Windows:
		`CMPRT07=C:\tmp\report7.txt`

Default setting	none	
Dynamic specification	YES	
Specification within session	NO	

CMSYNIN - Batch Input File for Natural Commands and INPUT Data

CMSYNIN - Batch Input File for Natural Commands and INPUT Data

This Natural profile parameter is for UNIX and Windows. It applies to batch mode only.

CMSYNIN is used for the batch input file. It contains Natural commands and data to be read by INPUT statements during execution of Natural programs (optionally). The number of characters actually processed is restricted to 512 characters per line.

Possible settings	any string	**Example for UNIX:**
		`CMSYNIN = $HOME/tmp/batch.cmd`
		
		Example for Windows:
		`CMSYNIN=C:\tmp\cmd.txt`

Default setting	none	
Dynamic specification	YES	
Specification within session	NO	

CMWRKnn - Natural Work Files

This Natural profile parameter is for UNIX and Windows. It applies to batch mode only.

CMWRK*nn* is used for Natural work files referenced by any Natural program executed during the session."*nn*" is a two digit decimal number in the range from 01 to 32 corresponding to the number used in a READ WORK FILE or WRITE WORK FILE statement.

Possible settings	any string	**Example for UNIX:**
		`CMWRK05=$HOME/tmp/workfile5.sag`
		
		Example for Windows:
		`CMWRK05=C:\tmp\workfile5.sag`

Default setting	none	
Dynamic specification	YES	
Specification within session	NO	

CO - Compiler Output

This Natural profile parameter is for UNIX platforms only.

This parameter can be used to deactivate or reactivate the compiler operation-in-progress window wherein, by default, the code is displayed as it is processed by the compiler.

Possible settings	ON	Compiler operation-in-progress window on. This setting is useful in the event of a crash, because the code display will stop at the position affected.
	OFF	Compiler operation-in-progress window off.
Default setting	OFF	
Dynamic specification	YES	
Specification within session	NO	

COMPR - Set RPC Buffer Compression

For static specification, this parameter is available as a keyword subparameter of the NTRPC macro. For dynamic specification, this parameter is available as a subparameter of the profile parameter RPC.

This parameter can be used to set the RPC buffer compression. It is effective only, if the automatic Natural RPC execution is used (AUTORPC=ON) and the CALLNAT is executed without a stub. If a stub is used, the compression has already been set during stub generation. For details, see Using Compression in the Natural Remote Procedure Call documentation.

COMPR is specified on the client side only.

Possible settings	0	No compression will be performed.
	1	The send buffer contains modifiable fields and output fields and the format buffer. The reply buffer contains modifiable fields and input fields.
	2	Same as COMPR=1, additionally the reply buffer also contains the format buffer.
Default setting	1	
Dynamic specification	YES	
Specification within session	YES	At runtime, this value can be overwritten using the Parameter Maintenance function of the SYSRPC utility.

For further information see the Natural Remote Procedure Call documentation.

COMSERVERID - Determine DCOM Server ID

This Natural profile parameter is for Windows platforms only.

It can be used to determine the DCOM server name (used by NaturalX).

Possible settings	not specified, or 1-32 characters	DCOM server name.
Default setting	not specified	If COMSERVERID is not specified, the default server name "DEFAULT" is used.
Dynamic specification	YES	
Specification within session	NO	

CP - Define Code Page Name

This Natural profile parameter is for:

- Mainframes
- UNIX and Windows

It determines the name of the code page used by the Entire Conversion Service (ECS). Currently, it applies only to the Natural RPC facility when the transport protocol ACI (that is EntireX Broker) is used.

For more information about ECS and the EntireX Broker refer to the section about Software AG's Internationalization in the EntireX Broker documentation.

Possible settings	1 to 40 characters	Valid code page name of Entire Conversion Services.
Default setting	none	
Dynamic specification	YES	
Specification within session	NO	

CSIZE - Size of Con-nect Buffer Area

This Natural profile parameter is for mainframes only.

It determines the size of the Con-nect buffer area. It only applies if Con-nect is installed.

Alternatively, you can use the equivalent Natural profile parameter DS or macro NTDS (see Using Optional Macros in a Natural Parameter Module in the Natural Operations for Mainframes documentation) to specify the buffer size.

Possible settings	0 - 512	Buffer size in KB. If CSIZE=0 is specified or if the requested space is not available, Con-nect cannot be used.
Default setting	0	
Dynamic specification	YES	
Specification within session	NO	

See the Con-nect Installation documentation for further information.

CSTATIC - Programs Statically Linked to Natural

This Natural profile parameter is for mainframes only.

With this parameter, you can define a list of names of non-Natural programs which are to be linked together with the Natural parameter module (NATPARM).

Possible settings	list of program names	For each program name (1-8 characters) an external reference is generated for the linkage editor. If the external reference (entry name) is different from the program name, the entry name can be specified enclosed in brackets after the name as follows: *Program-name(Entry-name)*
Default setting	none	
Dynamic specification	NO	
Specification within session	NO	

Each non-Natural program specified and linked to Natural can be called from a Natural program using a CALL statement.

As the value of any parameter is limited to 256 bytes, the number of program names specified by the CSTATIC parameter is limited. Alternatively, the macro NTCSTAT may be used to define more statically linked programs, see examples below.

Modules which have been statically linked can be replaced dynamically by loading them during session initialization, see the profile parameter RCA. Modules which are linked neither statically nor dynamically are loaded dynamically when they are first invoked by a CALL statement.

If you want to link programs to a shared nucleus, you have to define them with the CSTATIC parameter in two parameter modules: one linked to the shared nucleus and the other linked to the environment-dependent nucleus.

For further information, see Statically Linked Non-Natural Programs (in the Natural Operations for Mainframes documentation).

The following topics are covered below:

- Example of CSTATIC Parameter
- Example of NTCSTAT Macro

Example of CSTATIC Parameter

```
CSTATIC=(PROG1,PROG7(ENTRY2),PROG12,PROG27($MAIN))
```

Example of NTCSTAT Macro

```
NTCSTAT PROG1,PROG7(ENTRY2),PROG12
NTCSTAT PROG27($MAIN)
```


CVMIN - Control Variable Modified at Input

This Natural profile parameter is for:

- Mainframes
- UNIX and Windows

It determines whether or not an attribute control variable is assigned the status "MODIFIED" when the setting of the field to which the attribute control variable is attached is overwritten by an **identical** setting. If an attribute control variable has been assigned the status "MODIFIED", the MODIFIED option evaluates this as TRUE. This applies regardless of whether the input was entered manually, read from the stack or supplied in batch mode.

Possible settings	ON	If a field setting is overwritten by the same setting, the corresponding control variable will be assigned the status "MODIFIED".
	OFF	If a field setting is overwritten by the same setting, the corresponding control variable will not be assigned the status"MODIFIED".
Default setting	OFF	
Dynamic specification	YES	
Specification within session	NO	

DATSIZE - Minimum Size of Buffer for Local Data

This Natural profile parameter is for mainframes only.

With this parameter you can set the minimum size of the local data buffer (DATSIZE).

Alternatively, you can use the equivalent Natural profile parameter DS or macro NTDS (see Using Optional Macros in a Natural Parameter Module in the Natural Operations for Mainframes documentation) to specify the DATSIZE value.

Possible settings	10 - 2097151	Minimum buffer size in KB.
Default setting	32	
Dynamic specification	YES	
Specification within session	NO	

The DATSIZE buffer is a "variable size" buffer. If more storage for local data areas is required during the session, the DATSIZE buffer is expanded dynamically. In a thread environment, the DATSIZE may be temporarily allocated outside the storage thread if it becomes too large. The size of the DATSIZE buffer is reduced back to the minimum size when the application does not need the space any longer.

Function of the DATSIZE Buffer

At execution time, the DATSIZE buffer holds the local data used by the Natural main program being executed and the local data of all subordinate objects (except FETCHed programs) invoked by this program.

When you use Natural in a development environment, the minimum DATSIZE required is the default setting (that is, 32 KB). A smaller DATSIZE is only possible when using Natural as a runtime-only environment without any Natural utilities being available.

Calculating the DATSIZE Requirement

The actual DATSIZE requirement can be calculated as follows (refer to the illustration below):

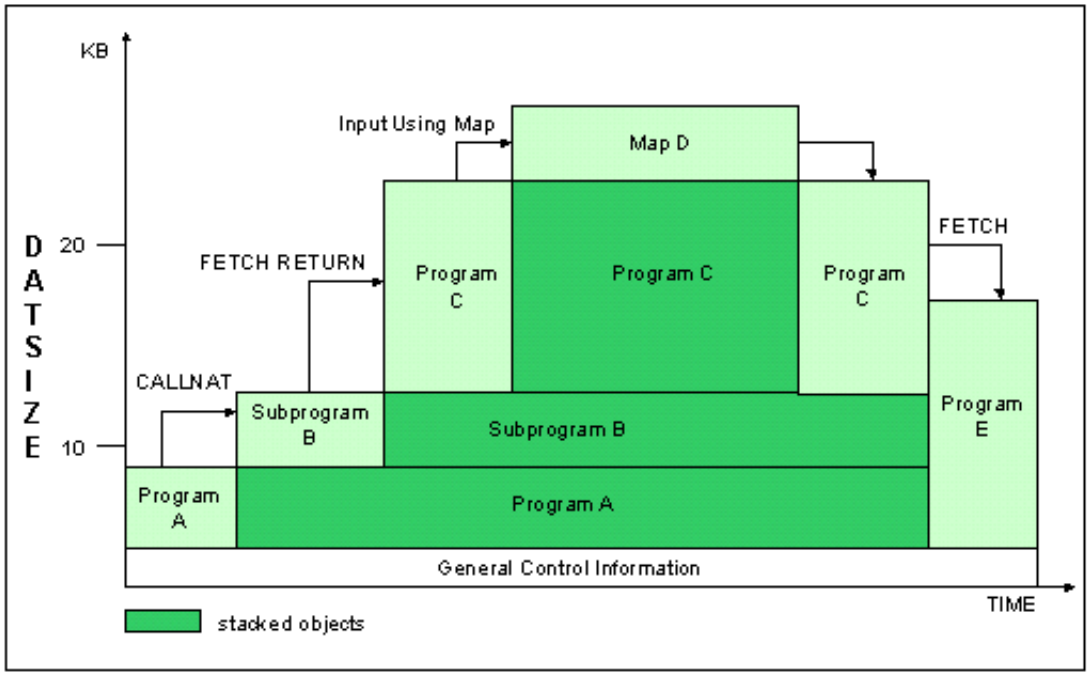
If another object is invoked by the main program, the local data of this object are also held in the DATSIZE buffer.

If other objects are invoked from the invoked object (with a CALLNAT, PERFORM, FETCH RETURN, INPUT USING MAP statement, a helproutine/helpmap being invoked), their local data are also held in the DATSIZE buffer; the local data of an invoked object is held in the DATSIZE buffer until control is returned from the invoked object to the invoking object.

If another main program is invoked with a FETCH statement, the local data of all previously invoked objects are deleted from the DATSIZE buffer and the local data of the FETCHed program are held in the DATSIZE.

In addition, an amount of approximately 128 bytes of general control information for execution are held in the DATSIZE buffer, plus approximately 128 bytes of control information for each object whose local data are being held in the DATSIZE buffer. This is illustrated in the figure below.

DATSIZE - Minimum Size of Buffer for Local Data



The **system command** LIST provides an option to display directory information about an object. This information includes the object's DATSIZE storage requirement (not including the control information).

DB - Database Types and Options

This Natural profile parameter is for mainframes only.

It can be used to define database types and options for all and for specific database IDs. It corresponds to the NTDB macro in the parameter module NATPARM.

Possible settings	*database-type*	*database-type* is besides ADABAS and its synonym ADAV7, for example, "DLI", "VSAM", "DB2", etc. (see the documentation for the corresponding Natural database management interface). This subparameter is mandatory.
	database-ID	*database-ID* must be in the range from 0 to 65535. Database ID "255" must not be specified, because it is reserved for internal use. You can specify a single database ID, a list of database IDs enclosed in parentheses, or an asterisk (*) to indicate all databases.
	options	*options*, see Possible Database Options below.
Default setting	ADABAS	Default database type.
Dynamic specification	YES	This parameter can only be specified dynamically. In the Natural parameter module NATPARM, the macro NTDB must be used instead.
Specification within session	NO	

The following topics are covered below:

- DB Parameter Syntax
- NTDB Macro Syntax
- Possible Database Options
- Examples of NTDB Macro
- Examples of DB Parameter

DB Parameter Syntax

The DB parameter is specified as follows:

1. Default Database Definition

The default database type and its default options is specified as follows. It applies to all database IDs not explicitly specified by the DB parameter or NTDB macro. If there are no options, the commas and the asterisk can be omitted.

```
DB=(database-type,*,options)
```

2. Single Database Definition

A single database ID is specified as follows:

```
DB=(database-type,database-ID,options)
```


3. Multiple Database Definition

Multiple database IDs of the same database type with the same options can be specified together, enclosed in parentheses:

```
DB=(database-type,(database-ID1,database-ID2,...),options)
```

NTDB Macro Syntax

The NTDB macro is specified as follows:

1. Default Database Definition

The default database type and its default options is specified as follows. It applies to all database IDs not explicitly specified by the DB parameter or NTDB macro. If there are no options, the commas and the asterisk can be omitted.

```
NTDB database-type,*,options
```

2. Single Database Definition

A single database ID is specified as follows:

```
NTDB database-type,database-ID,options
```

3. Multiple Database Definition

Multiple database IDs of the same database type with the same options can be specified together, enclosed in parentheses:

```
NTDB database-type,(database-ID1,database-ID2,...),options
```

Possible Database Options

The following options can be specified for both the DB parameter and the NTDB macro:

OPEN	This option applies to Adabas databases only, for which Adabas requires an open request to be issued. If OPEN is specified for such a database, an open request is always issued (even if the ETID is blank).
READ	The database is to be read-only.
ETP	The database is to be handled by Entire Transaction Propagator.
ENTIRE	The database is to be handled by Entire DB.
ADASTAR	The database is to be handled by Adastar.

The following options can be specified for the DB parameter only.

NOOPEN	Resets the OPEN option.
NOREAD	Resets the READ option.
NOETP	Resets the ETP option.
NOENTIRE	Resets the ENTIRE option.
NOADASTAR	Resets the ADASTAR option.
OFF	Removes any DB or NTDB definition for the specified databases.

Examples of NTDB Macro

`NTDB DLI,7`	This defines Database 7 as DL/I database.
`NTDB ADAV62,(10,15,57),ETP`	In the Natural parameter module NATPARM, this defines Databases 10, 15 and 57 as Adabas Version 6.2 databases which are to be handled by Entire Transaction Propagator.

Example of DB Parameter

`DB=(VSAM,(22,26,33))`	This defines Databases 22, 26 and 33 as VSAM databases.
`DB=(,*,READ)`	This sets all databases for which the default database definition applies to read-only.
`DB=(,(8,9),NOREAD)`	This removes the read-only option for Databases 8 and 9.
`DB=(,17,OFF)`	This resets the database definition of Database 17 to defaults.

DB2SIZE - Natural Buffer Area for DB2 or SQL/DS

This Natural profile parameter is for mainframes only. It only applies to Natural for DB2 and Natural for SQL/DS.

It sets the maximum size of the buffer area required by Natural for DB2 and Natural for SQL/DS.

Possible settings	0 - 64	Maximum size of the buffer area in KB. If the requested space is not available, the Natural for DB2 or Natural for SQL/DS interface cannot be used. Set DB2SIZE to 0 if Natural is **not** to be used for DB2 or SQL/DS. If Natural is to be used for DB2 or SQL/DS, DB2SIZE must be set to at least 40 KB.
Default setting	0	
Dynamic specification	YES	
Specification within session	NO	

DBCLOSE - Database Close at Session End

This Natural profile parameter is for mainframes only.

It determines whether or not Natural closes all databases that it has accessed during a session at the end of this session.

Possible settings	ON	Natural closes all databases.
	OFF	Natural does not close any database.
Default setting	OFF	
Dynamic specification	YES	
Specification within session	NO	

DBID - Default Database ID of Natural System Files

This Natural profile parameter is for mainframes only.

It identifies the default database in which the Natural system files (FNAT, FUSER, FDIC, FSEC, FSPOOL) are located.

Possible settings	0 - 254, 256 - 65535	Database ID. **Note:** Database ID 255 is reserved for internal use.
Default setting	0	
Dynamic specification	YES	If you specify the DBID parameter dynamically, the database ID for all system files is set to this setting. Therefore, you must specify the DBID parameter **before** any individual system file parameter (FNAT, FUSER, FDIC, FSEC, FSPOOL) if you want to specify any of these parameters, too.
Specification within session	NO	

Database IDs for individual system files can be specified with the parameters FNAT, FUSER, FDIC, FSEC and FSPOOL. The database ID specified with the DBID parameter applies to all Natural system files for which no individual database IDs are specified.

The type of database system is determined by the specification in the NTDB macro.

DBOPEN - Database Open without ETID

This Natural profile parameter is for mainframes only.

It controls the Open handling of Natural. DBOPEN overrules the setting ETID=blanks.

Possible settings	ON	A database open will be issued even if the ETID parameter is set to blanks.
	OFF	No database open will be issued if the ETID parameter is set to blanks. **Exception**: One open command will always be sent to the datebase specified as ETDB, even if ETID is set to blanks and DBOPEN is set to OFF.
Default setting	OFF	
Dynamic specification	YES	
Specification within session	NO	

DBROLL - Database Calls before Roll-Out

This Natural profile parameter is for mainframes and applies under CICS and Com-plete only.

It determines the number of database calls after which a roll-out of the Natural thread will be performed.

Possible settings	0 - 32767	Number of database calls after which a roll-out is performed.
Default setting	0	No roll-out is performed for database calls.
Dynamic specification	YES	
Specification within session	NO	

The Natural session is suspended during the roll-out.

DBSHORT - Interpretation of Database Short Names

This Natural profile parameter is for UNIX and Windows platforms only.

It can be used to determine whether database short names are to be shown or not.

Possible settings	ON	Short names are shown.
	OFF	Short names are not shown.
Default setting	ON	
Dynamic specification	YES	
Specification within session		

DBUPD - Database Updating

This Natural profile parameter is for:

- Mainframes
- UNIX and Windows

It indicates whether database updating is to be permitted during the Natural session.

Possible settings	ON	Database update is permitted.
	OFF	Database update is not permitted. A Natural statement which would cause a database update (STORE, UPDATE, DELETE) is not executed and an error message is generated.
Default setting	ON	
Dynamic specification	YES	
Specification within session	NO	

The DBUPD setting has no effect on the execution of Natural system commands.

DC - Character for Decimal Point Notation

This Natural profile and session parameter is for all platforms.

It determines the character to be used for decimal point notation.

Within a Natural session, the profile parameter DC can be overridden by the session parameter DC.

Possible settings	any character (except numeric characters)	You specify the DC parameter as DC='*c*' where *c* represents the character to be used as decimal point. The character specified with the DC parameter • must not be the same as the one specified with the IA profile/session parameter (input assign character), STACKD profile parameter (stack delimiter character) or ID profile/session parameter (input delimiter character), • should not be the same as the one specified with the CF profile/session parameter (control character for terminal commands) or HI profile parameter (help character).		
Default setting	. (period)			
Dynamic specification	YES			
Specification within session	YES	Applicable Statements:	SET GLOBALS	Parameter is evaluated at runtime.
		Applicable Command:	GLOBALS	Parameter may be specified dynamically with the GLOBALS system command.

DD - Day Differential

This Natural profile parameter is for:

- Mainframes
- UNIX and Windows

It adjusts the current machine date (as read by using the internal machine time) by adding/subtracting any number of days to/from it. This makes it possible to re-run an application that was to be run at a certain date but for some reason could not be run at that date.

The DD profile parameter is specified as follows:

```
DD=+nn
```

or

```
DD=-nn
```

where *nn* is the number of days.

Possible settings	**Mainframes**	**UNIX and Windows**	
	-32767 to +32767	-10953 to +10953	Machine date is adjusted. Specification of "+" is optional.
	0	0	No adjustment is made.
Default setting	0	0	
Dynamic specification	YES	YES	
Specification within session	NO	NO	

See also the profile parameters TD and YD.

DELETE - Deletion of Dynamically Loaded Programs

This Natural profile parameter is for mainframes only.

It determines whether dynamically loaded non-Natural programs are to be deleted at the completion of the Natural program in which they are used.

Possible settings	ON	Dynamically loaded non-Natural programs are deleted at the end of the Natural program in which they were loaded.
	OFF	Dynamically loaded non-Natural programs are not deleted at the end of the Natural program in which they were loaded; they are kept until command mode is entered again, and the deletion is performed immediately before Natural enters command mode processing.
Default setting	ON	
Dynamic specification	YES	
Specification within session	NO	

The following platform-specific requirements apply:

Platform:	**Comment:**
Under CICS	In a CICS environment, this parameter applies only if the non-Natural program is invoked via standard linkage conventions (SET CONTROL ’P=S’).
Under OS/390 Batch, TSO, VSE/ESA Batch and IMS/TM	This parameter does not apply in an IBM Language Environment (LE). All dynamic subprograms loaded during a Natural session are deleted upon LE environment termination, that is, during the termination of the Natural session. For more information about Natural in an LE environment, see Support of IBM Language Environment Subprograms (in the Natural Operations for Mainframes documentation).

DFOUT - Date Format for Output

This Natural profile and session parameter is for all platforms.

It determines the format in which the settings of date variables are displayed by INPUT, DISPLAY, PRINT and WRITE statements.

Within a Natural session, the profile parameter DFOUT can be overridden by the session parameter DFOUT.

Possible settings	S	Date variables are displayed with a 2-digit year component, and delimiters as determined by the profile parameter DTFORM. For example: *yy-mm-dd*.	
	I	Date variables are displayed with a full 4-digit year component and no delimiters. For example: *yyyymmdd*.	
Default setting	S		
Dynamic specification	YES		
Specification within session	YES	Applicable Statements:	SET GLOBALS
		Applicable Command:	GLOBALS

For details on session parameter specification at statement or element level and evaluation at compilation time or at runtime, refer to Session Parameter Specification/Evaluation Overview.

Notes:

- The **profile parameter** DFOUT is evaluated at runtime. It applies to date fields in INPUT, DISPLAY, PRINT and WRITE statements for which no explicit edit mask is specified and for which the **session parameter** DF is not set.
- The sequence of the day, month and year components in the date settings is determined by the DTFORM profile parameter.

See also Processing of Date Information in the Natural Programming Guide documentation.

DFS - Specify RPC Client's Default Server Address

For static specification, this parameter is available as a keyword subparameter of the NTRPC macro. For dynamic specification, this parameter is available as a subparameter of the profile parameter RPC.

DFS can be used to define an RPC default server address. It determines the server name, the server node, the logon indicator and the transport protocol. The default server address will be used only if no appropriate server is found in the service directory. For further information, see Specifying RPC Server Adresses in the Natural Remote Procedure Call documentation.

To define a default server address, you specify up to 4 subparameters.

DFS is specified on the client side only.

Possible settings	*server name*	(1 - 192 characters). There is no default, the value must be specified.
	server node	(1 - 192 characters). There is no default, the value must be specified.
	logon indicator	L The client initiates a Natural logon to the server with the library name of the current library on the client. ' ' (blank) No server logon will be executed. If nothing is specified, blank is the default. **Note for Windows platforms:** Instead of "L", check the selection box.
	transport protocol	The transport protocol to be used. ACI is the only possible value and the default.
Default setting	none	Subparameter defaults, see above.
Dynamic specification	YES	See below.
Specification within session	YES	At runtime, this value can be overwritten by a Natural application interface (USR2007N).

For dynamic specification, the syntax is as follows:

```
DFS=(server-name,server-node
name,logon-indicator,transport-protocol-name)
```

For the possible values for *server-name* and *server-node*, refer to SRVNAME and SRVNODE.

For further information, see the Natural Remote Procedure Call documentation.

DFSTACK - Date Format for Stack

This Natural profile and session parameter is for all platforms.

It determines the format in which the settings of date variables are placed on the stack via a STACK, RUN or FETCH statement.

Within a Natural session, the profile parameter DFSTACK can be overridden by the session parameter DFSTACK.

Possible settings	S	Date variables are placed on the stack with a 2-digit year component, and delimiters as determined by the profile parameter DTFORM. For example: *yy-mm-dd*.	
	C	Same as DFSTACK=S. In addition, if the century used when the setting is read from the stack is not the same as that of the original date setting, Natural will issue a runtime error.	
	I	Date variables are placed on the stack with a full 4-digit year component and no delimiters. For example: *yyyymmdd*.	
Default setting	S		
Dynamic specification	YES		
Specification within session	YES	Applicable Statements:	SET GLOBALS
		Applicable Command:	GLOBALS

For details on session parameter specification at statement or element level and evaluation at compilation time or at runtime, refer to Session Parameter Specification/Evaluation Overview.

The profile parameter DFSTACK does not apply to STACK, RUN or FETCH statements for which the session parameter DF is set.

See also Processing of Date Information in the Natural Programming Guide.

DFTITLE - Output Format of Date in Standard Report Title

This Natural profile ans session parameter is for all plattforms.

It determines the output format of the date in the default title line of a report page (as output with a DISPLAY, WRITE or PRINT statement).

Within a Natural session, the profile parameter DFTITLE can be overridden by the session parameter DFTITLE.

Possible settings	S	The date is output with a 2-digit year component and delimiters. For example: *yy-mm-dd*.	
	L	The date is output with a 4-digit year component and delimiters. For example: *yyyy-mm-dd*.	
	I	The date is output with a 4-digit year component and no delimiters. For example: *yyyymmdd*.	
Default setting	S		
Dynamic specification	YES		
Specification within session	YES	Applicable Statements:	SET GLOBALS
		Applicable Command:	GLOBALS

For details on session parameter specification at statement or element level and evaluation at compilation time or at runtime, refer to Session Parameter Specification/Evaluation Overview.

Notes:

- DFTITLE is evaluated at runtime and determines whether the date is displayed with a 2-digit or 4-digit year component with or without delimiters. It has no effect on a user-defined page title (as specified with a WRITE TITLE statement).
- The sequence of the day, month and year components and the delimiter characters used are determined by the profile parameter DTFORM.

See also Processing of Date Information and Date Format for Default Page Title - DFTITLE Parameter in the Natural Programming Guide.

DLISIZE - Size of Natural Buffer Area for DL/I

This Natural profile parameter is for mainframes only. It only applies to Natural for DL/I.

It determines the maximum size of the buffer area required by Natural for DL/I. If the requested space is not available, Natural for DL/I cannot be used.

Possible settings	0, or 26-512	Buffer size in KB. The size actually required depends on the specifications in the NDLPARM macro (see the Natural for DL/I documentation). If you use the default specifications in NDLPARM, DLISIZE=26 is sufficient.
Default setting	0	If you do not need DL/I support during a Natural session, you are recommended to invoke Natural with DLISIZE=0 to avoid overhead caused by handling of unused buffers.
Dynamic specification	YES	
Specification within session	NO	

If the size specified with the DLISIZE parameter is not sufficient, an appropriate error message at initialization of Natural for DL/I will tell you what size to specify.

If Natural for DL/I is installed, the corresponding Natural buffers are requested at the initialization of the Natural session.

DS - Define Size of Storage Buffer

This Natural profile parameter is for mainframes only.

It defines the default initial size of various Natural storage buffers.

In previous versions of Natural, single-value parameters (e.g. SSIZE) were used to define the sizes of the buffers. The DS profile parameter is a universal parameter to specify all buffer sizes. It corresponds to the NTDS macro in the Natural parameter module NATPARM.

See also Natural Storage Management in the Natural Operations for Mainframes documentation.

Possible settings	*(name,size)*	*Name* is the buffer name (1-8 characters), see Table of Buffer Sizes below. *Size* is the buffer size in kilobytes. For limit values, see Table of Buffer Sizes below.
Default setting	See table below.	
Dynamic specification	YES	Multiple pairs of buffer names/sizes can be specified. This parameter can only be specified dynamically. In the Natural parameter module NATPARM, the corresponding macro NTDS must be used instead.
Specification within session	NO	

You may continue using the old parameters or you may use the old parameters in parallel to the parameter DS. During the dynamic parameter evaluation, old buffer size parameters are converted internally into the new DS parameter format, for example, `SSIZE=55` is converted into `DS=(SSIZE,55)`.

The following topics are covered below:

- DS Parameter Syntax
- NTDS Macro Syntax
- Table of Buffer Sizes
- Examples

DS Parameter Syntax

The DS parameter is specified as follows:

```
DS=(name1,size1,name2,size2,...)
```

NTDS Macro Syntax

The NTDS macro is specified as follows:

```
NTDS name1,size1
NTDS name2,size2
...
```


Table of Buffer Sizes

Buffer Name	Description	Buffer Size	Default	Old
ASIZE	Entire System Server auxiliary buffer	0-64	0	YES
BSIZE	Size of EntireX Broker buffer	0-64	0	YES
CSIZE	Size of Con-nect buffer area	0-128	0	YES
DATSIZE	Size of buffer for local data	10-1024	32	YES
DSIZE	Size of debug buffer area	0, 2-64	2	YES
EDPSIZE	Size of the Software AG Editor auxiliary buffer pool	0, 32-32767	0	YES
ETPSIZE	Size of Entire Transaction Propagator buffer	0, 10-128	0	YES
EXCSIZE	Size of buffer for Natural Expert C interface	0-256	0	YES
EXRSIZE	Size of buffer for Natural Expert rule tables	0-256	0	YES
MONSIZE	Size of SYSTP monitor buffer	0-256	0	YES
MULFETCH	Size of Multi-fetch buffer	0-1024	64	NO
NAFSIZE	Size of buffer for Natural Advanced Facilities	0-64	0	YES
RDCSIZE	Size of buffer for the Natural Data Collector	0, 2-128	0	YES
RJESIZE	Initial size of NATRJE buffer	0-256	8	YES
RUNSIZE	Size of runtime buffer	10-64	16	YES
SSIZE	Size of Software AG Editor buffer	0, 40-128	64	YES
NSFSIZE	Size of SAF interface buffer **Note:** Replaces the RDCSIZE parameter used with NSF Versions 3.1.x.	0, 8-64	0	NO
TSIZE	Size of the buffer for Adabas Text Retrieval	0-128	0	YES
XSIZE	Size of buffer for user subsystem	0-64	0	YES
ZSIZE	Size of Entire DB buffer area	0-64	0	YES

Examples

Example of DS parameter:

```
DS=(ASIZE,33,TSIZE,60,EDPSIZE,500)
```

Equivalent in Natural parameter macro NATPARM:

```
NTDS ASIZE,33
NTDS TSIZE,60
NTDS EDPSIZE,500
```


DSC - Data-Stream Compression (for 3270-Type Terminals)

DSC - Data-Stream Compression (for 3270-Type Terminals)

This Natural profile parameter is for mainframes only. It only applies to 3270-type terminals.

With this parameter, you can switch off Natural's automatic optimization of the screen data stream for 3270-type terminals.

Possible settings	ON	Data-stream compression is used.
	OFF	Data-stream compression is not used.
Default setting	ON	
Dynamic specification	YES	
Specification within session	NO	

Natural's screen optimization causes screen data to be sent as compressed as possible. If this should conflict with any TP monitor's screen optimization or hardware limitation, you can use this parameter to switch off Natural's screen optimization; screen data will then be sent in non-compressed form.

This parameter has the same function as the terminal command %RO.

If you use the BX session parameter settings BX=L or BX=R, you should switch off Natural's screen optimization using DSC=OFF or %ROOFF..

DSIZE - Size of Debug Buffer Area

This Natural profile parameter is for mainframes only.

It specifies the size of the Natural debug buffer area.

Alternatively, you can use the equivalent Natural profile parameter DS or macro NTDS (see Using Optional Macros in a Natural Parameter Module in the Natural Operations for Mainframes documentation) to specify the buffer size.

Possible settings	0, or 2 - 64	Buffer area in KB. If DSIZE is set to 0 or if the requested space is not available, the DBLOG utility cannot be used.
Default setting	2	
Dynamic specification	YES	
Specification within session	NO	

This buffer area is used by the DBLOG function which is described in the Natural Utilities documentation, section Debugging and Monitoring.

DSLM - Size Limitation

This Natural profile parameter is for UNIX and Windows platforms only.

It limits index values, number of array occurrences and array sizes to the limits applicable for Natural Version 3.1 for Mainframes. The parameter is used for compatibility reasons only.

Note:
This parameter will be removed with the next Natural version.

Possible settings	ON	The above mentioned items will be limited.
	OFF	No limit is in effect.
Default setting	OFF	
Dynamic specification	YES	
Specification within session	NO	

DTFORM - Date Format

This Natural profile parameter is for all platforms.

It indicates the default format in which dates are to be provided automatically by Natural as part of the default title on Natural reports, as date constants and date input.

Possible settings	**Value**	**Area**	**Date Format**
	E	Europe	DD/MM/YYYY
	G	Germany	DD.MM.YYYY
	I	International	YYYY-MM-DD
	U	USA	MM/DD/YYYY
Default setting	I		
Dynamic specification	YES		
Specification within session	NO		

The first day of a week is assumed to be Monday - unless DTFORM=U is specified, in which case Sunday is used.

For date constants, the year component (YYYY) consists of all four digits. Only the last two digits of the year component are used for reports, date input, the Natural system function VAL, and when the date is moved to an alphanumeric field.

The output format of the date in a default report page title is also specified by the profile parameter DFTITLE.

See also Processing of Date Information and Default Edit Mask for Date - DTFORM Parameter in the Natural Programming Guide.

DU - Dump Generation

This Natural profile and session parameter is for all platforms.

This parameter determines whether a memory dump is to be generated in the case of an abnormal termination during the Natural session.

Within a Natural session, the profile parameter DU can be overridden by the session parameter DU.

Possible settings	ON	A memory dump is produced in the case of an abnormal termination (TP-monitor dump dataset or SYSUDUMP in OS/390 batch mode or TSO). Then the Natural session terminates with the error message NAT9974.	
	OFF	No memory dump is produced. In batch mode, subsequent action taken by Natural is determined by the setting of the CC profile parameter. In online mode, Natural responds with the error message NAT0953, NAT0954, NAT0955 or NAT0956. For further information on the abnormal termination, you can use the system command DUMP.	
	SNAP	**On mainframes:** This setting forces an immediate dump in the case of an abnormal termination during a Natural session. The Natural session continues as with DU=OFF after the dump has been taken.	
	FORCE	**On mainframes:** This setting forces an immediate dump in the case of an abnormal termination during a Natural session and terminates the Natural session immediately. This is useful for testing purposes in some environments.	
	ABEND	**On mainframes:** This works as with DU=ON, except that the session is terminated with the abend occurred - instead of the error message NAT9974. DU=ABEND is not available with the Natural session parameter DU.	
Default setting	OFF		
Dynamic specification	YES		
Specification within session	YES	Applicable Statements:	SET GLOBALS
		Applicable Command:	GLOBALS

For details on session parameter specification at statement or element level and evaluation at compilation time or at runtime, refer to Session Parameter Specification/Evaluation Overview.

Notes:

- Setting the DU profile parameter may impair the system performance considerably, due to I/O processing on the dump dataset.
- Be careful when you use this parameter, because all programs and subroutines currently active for the current user will be retained in the Natural buffer pool.
  DU=ON, SNAP or FORCE may cause buffer fragmentation which may result in a significant degradation in system performance.
- Under UTM, this parameter is ignored; under UTM, a dump is always produced in the case of an abnormal program termination.
- Profile parameter DUE can be used to get a storage dump for specific errors.

DUE - Dump for Specific Errors

This Natural profile parameter is for mainframes only.

This parameter allows you to specify Natural error numbers for which a storage dump shall be taken. This may be helpful to get a dump for the analysis of a specific error situation by Software AG personnel.

Possible settings	List of numbers 1-9999	One or more error numbers for which a dump shall be taken. If DUE is specified multiple times, all error numbers are saved in one table.
	OFF	Deletes the table and any error numbers specified previously are removed.
Default setting	OFF	
Dynamic specification	YES	
Specification within session	YES	Terminal command %DUE

If an error occurs which has been specified by DUE, a program check is forced. If the profile/session parameter DU=OFF is set, it will be changed to DU=ON. For further processing, the DU parameter setting is honored.

Examples:

```
DUE=1302
DUE=(6501,6502,6503,1500)
DUE=OFF
```


DYNPARM - Control Use of Dynamic Parameters

This Natural profile parameter is for:

- Mainframes
- UNIX

This Natural profile parameter can be used to restrict the use of dynamic profile parameters outside of PROFILE and SYS profile parameter strings. It corresponds to the NTDYNP macro in the parameter module NATPARM.

Possible settings	ON	All profile parameters can be specified dynamically.
	OFF	No profile parameters can be specified dynamically.
	`DYNPARM=(ON,`*`parameter-name`*`,...)`	Only those parameters whose *parameter-name* is specified, can be specified dynamically. Other parameters cause Error Message NAT7008 to be issued.
	`DYNPARM=(OFF,`*`parameter-name`*`,...)`	All profile parameters can be specified dynamically - except those whose *parameter-name* is specified. These parameters cause Error Message NAT7008 to be issued.
Default setting	ON	All profile parameters can be specified dynamically.
Dynamic specification	YES	Outside of PROFILE or SYS parameter strings, the DYNPARM parameter can be used only once and only if the NTDYNP macro is not specified in the Natural parameter module.
Specification within session	NO	

The parameter restrictions defined by DYNPARM (or the NTDYNP macro) do not apply within PROFILE or SYS profile parameter strings. If DYNPARM is used within PROFILE or SYS strings, it replaces any previous restrictions defined by DYNPARM or macro NTDYNP.

DYNPARM can be used only once within one string and should be placed at the end of it.

The following topics are covered below:

- DYNPARM Parameter Syntax
- NTDYNP Macro Syntax
- Examples

DYNPARM Parameter Syntax

The DYNPARM parameter is specified as follows:

```
DYNPARM=(ON,parameter-name1,parameter-name2,...)
```

or

```
DYNPARM=(OFF,parameter-name1,parameter-name2,...)
```

NTDYNP Macro Syntax

The NTDYNP macro is specified as follows:

```
NTDYNP ON,parameter-name1,parameter-name2,parameter-name3,...
NTDYNP parameter-name4,parameter-name5,...
...
```

or

```
NTDYNP OFF,parameter-name1,parameter-name2,parameter-name3,...
NTDYNP parameter-name4,parameter-name5,...
...
```

Examples

The example illustrates restricting of the dynamic parameters FNAT and FSEC. In the Natural parameter module NATPARM, the following parameter restriction should be defined:

```
NTPRM DBID=0,FNR=0
NTDYNP ON,PROFILE
```

Additionally, almost all parameter profiles could look like the following:

```
...,FNAT=(22,7,PASSW),FSEC=(22,9,PASSW),DYNPARM=(OFF,FNAT,FSEC)
```

If some special users are to be allowed to use all parameters including FNAT and FSEC, their parameter profiles could look like the following:

```
USER=(ADM1,ADM2),...,FNAT=(22,8),FUSER=(22,12),DYNPARM=(OFF,DUMMY)
```

This forces normal users to enter the PROFILE parameter as the first dynamic parameter. Subsequently, all parameters except FNAT and FSEC are allowed. Of course, the access to the parameter profile application SYSPARM must be restricted.

ECHO - Control Printing of Batch Input Data

This Natural profile parameter is for:

- Mainframes
- UNIX and Windows

It only applies in batch mode.

ECHO is used to enable or disable the printing of input data from the dataset CMSYNIN or CMOBJIN for INPUT statements provided to Natural during batch mode processing.

Possible settings	ON	Natural prints the input data provided during batch mode processing to the batch output file CMPRINT.
	OFF	Natural does **not** print input data provided during batch processing.
Default setting	ON	
Dynamic specification	YES	
Specification within session	NO	

It is also possible to suppress printing of a **single input line** by preceding it with a line containing the terminal command for record suppression %*.

ECPMOD - Entire Connection Protocol Mode

This Natural profile parameter is for UNIX platforms only.

It determines the protocol that is used when downloading workfiles.

Possible settings	ON	The TCP/IP protocol is used.
	OFF	The ZMODEM protocol is used.
Default setting	ON	
Dynamic specification	YES	
Specification within session	NO	

EDBP - Software AG Editor Buffer Pool Definitions

This Natural profile parameter is for mainframes only.

It controls the initialization and operation of the editor buffer pool and its work file. It corresponds to the NTEDBP macro in the Natural parameter module NATPARM.

Possible settings	See below	Various keywords subparameter are available.
Default setting	See below	
Dynamic specification	YES	This parameter can only be specified dynamically. In the Natural parameter module NATPARM, the macro NTEDBP must be used instead.
Specification within session	YES	Use the SYSEDT Editor Buffer Pool Services utility.

The editor buffer pool is defined for a session by profile parameter BPI with TYPE=EDIT or by profile parameter EDPSIZE (editor auxiliary buffer pool).

Shared Editor Buffer Pool

If the editor buffer pool is shared between multiple Natural sessions, all subparameters (except DDNAME, DSNAME and FMODE) are honored by the very first session only, which initializes the editor buffer pool work file during a buffer pool cold start. During a buffer pool warm start, the editor buffer pool subparameters (except DDNAME, DSNAME and FMODE) are read from the buffer pool work file.

With subparameter COLD=ON, a buffer pool cold start can be forced during the initialization of the editor buffer pool.

Editor Auxiliary Buffer Pool

If an editor auxiliary buffer pool is used (see profile parameter EDPSIZE), only the following subparameters apply:

FTOUT, LRECL, MAXLF

More Information

For more information on the editor buffer pool, refer to Editor Buffer Pool in the Natural Operations for Mainframes documentation.

For more information on buffer pool performance, refer to the SYSEDT Editor Buffer Pool Services utility documentation.

The following topics are covered below:

- EDBP Parameter Syntax
- NTEDBP Macro Syntax
- Keyword Subparameters

EDBP Parameter Syntax

The EDBP parameter is specified as follows:

```
EDBP=(keyword1=value1, keyword2=value2,...)
```

NTEDBP Macro Syntax

The NTEDBP macro is specified as follows:

```
NTEDBP keyword1=value1, keyword2=value2,...
```

Keyword Subparameters

The following keyword subparameters are available:

COLD | CTOUT | DDNAME | DSNAME | DTOUT | FMODE | FTOUT | IMSG | ITOUT | LRECL | LTOUT | MAXLF | PWORK | RECNUM | RWORK | UTOUT

COLD - Buffer Pool Cold Start

Determines whether a buffer pool cold start is performed.

Possible values	ON or OFF
Default value	OFF

A cold start means that the buffer pool work file is cleared and reinitialized during buffer pool initialization. Any editor recovery information and all buffer pool parameters stored in the work file are lost.

CTOUT - Timeout for Changed Buffer Pool Blocks

Determines the timeout value (in seconds) for changed buffer pool blocks.

Possible values	1-32767
Default value	120

A changed buffer pool block is written to the work file after the specified time interval has been exceeded, and no unchanged or free block is available.

DDNAME - Logical Work File Name of the JCL Definition

Determines the logical work file name of the JCL definition.

Possible values	1 to 8 bytes
Default value	CMEDIT

Notes:

- Under CICS: A corresponding file control table entry must be defined for the editor work file.
- Under Com-plete: The specified logical work file name is the name of the SD file.

DSNAME - Work File Dataset Name

Determines the work file dataset name for batch and TSO under OS/390 only.

Possible values	1-44 bytes
Default value	None

If no DD JCL statement is supplied and no ALLOC statement is issued (under TSO only) for the editor work file, then DSNAME will be allocated dynamically.

DTOUT - Logical File Timeout Check Value

Determines the logical file timeout check value (in seconds).

Possible values	1-32767
Default value	300

Logical files are checked for timeout each time the specified time interval has been exceeded.

FMODE - Work File Mode

Determines the work file mode (under VM/CMS and Com-plete/SMARTS).

Possible values	1-2 bytes
Default value	A1

With FMODE = SM, SMARTS work files are used. The SMARTS environment variable $NAT_WORK_ROOT must be set accordingly.

The file type is always DATA.

FTOUT - Timeout Value for Logical Files

Determines the timeout value (in seconds) for logical files.

Possible values	60-16777215
Default value	86400

A logical file is deleted after the specified time interval has been exceeded and no access has occurred.

IMSG - Buffer Pool Initialization and Termination Message

Determines whether a buffer pool initialization and termination message is issued on the operator console.

Possible values	ON or OFF
Default value	OFF

ITOUT - Buffer Pool Initialization Timeout Value

Determines the buffer pool initialization timeout value (in seconds) for multi-user buffer pools only.

Possible values	1-32767
Default value	300

The buffer pool is initialized by the first user by whom it is accessed. Other users have to wait until the first user finishes initialization. If the initialization is not finished after the specified time interval (for example, due to an abnormal termination of the first user), all other users receive an error message.

LRECL - Work File Record Length

Determines the buffer pool block size and work file record length.

This parameter is honored under BS2000/OSD, under Com-plete, under VM/CMS and for editor auxiliary buffer pools only.

For other environments, the work file record length is determined when the editor work file is created.

Possible values	800-16384
Default value	4096

Under BS2000/OSD: The record length must be a multiple of 2048 bytes.

LTOUT - Timeout Value for Locked Buffer Pool Blocks

Determines the timeout value (in seconds) for locked buffer pool blocks.

Possible values	1-32767
Default value	20

A buffer pool block that was locked during a read from the work file is freed after the specified time interval has been exceeded.

MAXLF - Maximum Number of Logical Files

Determines the maximum number of logical files.

Possible values	100-999999
Default value	1000

PWORK - Percentage of Work File Records Used as Work Records

Determines the percentage of work file records used as work records during an editor buffer pool cold start.

Possible values	0-100
Default value	50

The remaining records are used as recovery records.

RECNUM - Number of Work File Records

Determines the number of work file records (**under VM/CMS and Com-plete only**) during an editor buffer pool cold start.

Possible values	100 - 65535
Default value	200

This number determines the size of the work file.

Note:
For environments other than CMS, the number of work file records is determined when the editor work file is created.

RWORK - Percentage of Work Records Used for Regular Logical Files

Determines the percentage of work records that is used for regular logical files during an editor buffer pool cold start.

Possible values	51-100
Default value	90

The remaining records are used internally to release blocks from the buffer pool.

UTOUT - Timeout Value for Unchanged Buffer Pool Blocks

Determines the timeout value (in seconds) for unchanged buffer pool blocks.

Possible values	1-32767
Default value	20

An unchanged buffer pool block is written to the work file after the specified time interval has been exceeded and no free block is available.

EDITOR - Other Program Editor

This Natural profile parameter is for UNIX only.

It is modified to invoke a program editor other than the standard editor provided by Software AG.

Possible settings	any character string	Name of the editor you want to use. Specification, see below. **Note:** You can still use the former Natural program editor (as used with Natural versions prior to Version 2.2) by specifying NATCEDIT.
Default setting	NATEDIT	Standard editor provided by Software AG.
Dynamic specification	NO	
Specification within session	NO	

Note:
If an external editor is being used, the 4-digit line numbers at the beginning of each Natural source line have to be added manually, if an existing source is edited.

To use an external editor specify the path and editor name:

```
path-name editor-name command-to-specify-line %l command-to-specify-file %f
```

The string before "%l" is replaced by the ASCII representation of the line number. The string before "%f" is replaced by the file name as known by Natural.

Example for UNIX:

```
vi +%l %f
```

To use "vi" as a program editor, edit your parameter files as specified. If the existing parameter files are not updated, the specified editor (that is, "vi") cannot find the error line.

EDPSIZE - Size of Software AG Editor Auxiliary Buffer Pool

EDPSIZE - Size of Software AG Editor Auxiliary Buffer Pool

This Natural profile parameter is for mainframes only.

It determines the size of the Software AG Editor auxiliary buffer pool. It must be used when the Software AG Editor runs in a Sysplex environment. It allows the Software AG Editor to be run without a Software AG Editor (local or global) buffer pool.

Alternatively, you can use the equivalent Natural profile parameter DS or macro NTDS (see Using Optional Macros in a Natural Parameter Module in the Natural Operations for Mainframes documentation) to specify the buffer size.

Possible settings	0, or 48 - 2097151	Editor auxiliary buffer pool size in KB.
Default setting	0	No editor auxiliary buffer pool is used.
Dynamic specification	YES	
Specification within session	NO	

No Software AG Editor work file is required for the auxiliary buffer pool.

When the auxiliary buffer pool is used, the Software AG Editor’s recovery function is not available.

If EDPSIZE is not zero, an auxiliary buffer pool is allocated and used although a (local or global) Software AG Editor buffer pool is defined with the BPI profile parameter or the NTBPI macro.

For further information on the Software AG Editor, see Operating the Software AG Editor (in the Natural Operations for Mainframes documentation).

EDTBPSIZE - Software AG Editor Buffer Pool Size

This Natural profile parameter is for UNIX and Windows platforms only.

It sets the size of the Software AG Editor buffer pool (in KB).

Possible settings	0 - 4000	Size of the editor buffer pool in KB.
Default setting	400	
Dynamic specification	NO	
Specification within session	NO	

EDTLFILES - Number of Software AG Editor Logical Files

This Natural profile parameter is for UNIX and Windows platforms only.

It sets the maximum number of the Software AG Editor sessions a user can open at a time.

Possible settings	10 - 999	Maximum number of Software AG Editor sessions.
Default setting	100	
Dynamic specification	NO	
Specification within session	NO	

EDTRB - Program Editor Ring Buffer

This Natural profile parameter is for UNIX platforms only.

It can be used to determine whether the ring buffer of the program editor is to be used or not.

For further information, see Multiple Editor Session .

Possible settings	ON	Ring buffer is used.
	OFF	Ring buffer is not used.
Default setting	OFF	
Dynamic specification	NO	
Specification within session	NO	

EJ - Page Eject

This Natural profile and session parameter is for all platforms.

With this parameter, you specify whether a page eject is to be performed as a result of a logical page break, a break between program input and output, and the "normal end" message.

Within a Natural session, the profile parameter EJ can be overridden by the session parameter EJ. The EJ setting can in turn be overridden by an EJECT statement.

Possible settings	ON	A page eject is performed.		
	OFF	No page eject is performed. This setting may be used to save paper during test runs where page ejects are not needed.		
Default setting	ON			
Dynamic specification	YES			
Specification within session	YES	Applicable Statements:	SET GLOBALS	Parameter is evaluated at runtime.
		Applicable Command:	GLOBALS	Parameter may be specified dynamically with the GLOBALS system command.

This parameter only applies to the first report (Report 0). For additional reports, the **statement** EJECT with report specification (rep) has to be used.

Attention:

The **profile parameter** EJ has a slightly different meaning when specified for a Natural session under CICS in batch mode (for example, TTYPE=ASYL or TTYPE=BTCH); see Asynchronous Natural Processing under CICS in the Natural TP Monitor Interfaces documentation.

EMFM - Edit Mask Free Mode

This Natural profile parameter is for:

- Mainframes
- UNIX and Windows

With this parameter, you may activate/deactivate the Edit Mask Free mode at session startup. This mode allows you to omit literals during input into a field with a numeric edit mask.

Possible settings	ON	Edit Mask Free Mode is activated.
	OFF	Edit Mask Free Mode is deactivated.
Default setting	OFF	
Dynamic specification	YES	
Specification within session	NO	Within a running Natural session, you may override this setting with the terminal control command %FM+ or %FM-.

For additional information, see Numeric Edit Mask Free Mode in the INPUT statement description in the Natural Statements documentation.

ENDBT - BACKOUT TRANSACTION at Session End

This Natural profile parameter is for mainframes only.

It determines whether or not an implicit BACKOUT TRANSACTION statement is to be issued at the end of the Natural session.

Possible settings	ON	Natural will issue an implicit BACKOUT TRANSACTION statement at session end.
	OFF	Natural will not issue an implicit BACKOUT TRANSACTION statement at session end.
Default setting	ON	
Dynamic specification	YES	
Specification within session	NO	

ENDMSG - Display Session-End Message

This Natural profile parameter is for:

- Mainframes
- UNIX and Windows (in batch mode only)

By default, message NAT9995 is displayed at the end of the Natural session, indicating that the Natural session has been ended normally. With this parameter, you can suppress the display of this message.

Possible settings	ON	Message NAT9995 will be displayed at the end of the session.
	OFF	Message NAT9995 will not be displayed at the end of the session.
Default setting	ON	
Dynamic specification	YES	
Specification within session	NO	

If a session back-end program is defined with the profile parameter PROGRAM, the ENDMSG profile parameter has no effect; the message text will then be passed to the back-end program in the parameter area and will not be displayed by Natural.

ENDIAN - Endian Mode for Compiled Objects

This Natural profile and session parameter is for UNIX and Windows only.

It specifies the architecture for which the compiler should generate GP. See also Portable Natural Generated Programs (in the Natural Programming Guide).

Within a Natural session, the profile parameter settings can be overwritten by the session parameter ENDIAN.

Possible settings	DEFAULT	Endian mode is derived from the architecture currently used.	
	LITTLE	The compiler generates GP for Little Endian mode.	
	BIG	The compiler generates GP for Big Endian mode.	
Default setting	DEFAULT		
Dynamic specification	YES		
Specification within session	YES	Applicable Statements:	None
		Applicable Command:	GLOBALS

ESCAPE - Ignore Terminal Commands %% and %.

This Natural profile parameter is for:

- Mainframes
- UNIX and Windows

It can be used to disable the terminal commands "%%" and "%.".

Possible settings	ON	Enables the use of terminal commands "%%" and "%.".
	OFF	The terminal command "%%" and "%." will be ignored; that is, it will not be possible to leave the currently active Natural program or the Natural session respectively by entering "%%" or "%.".
Default setting	ON	
Dynamic specification	YES	
Specification within session	NO	

ESIZE - Size of User-Buffer Extension Area

This Natural profile parameter is for mainframes only.

It sets the size of the user-buffer extension area. This parameter determines the size of the Natural source area which is used by the Natural editors.

Possible settings	2 - 512	Size of buffer extension area in KB. **Note:** In a runtime environment (where the editors are not used), you can only set a value smaller than the default setting.
Default setting	28	
Dynamic specification	YES	
Specification within session	NO	

The user-buffer extension area contains:

- the source code of the Natural programming object to be compiled,
- the table of currently active PA/PF keys,
- other tables and work areas internally used by Natural.

In a production environment, Natural sources are not needed and the ESIZE value can therefore be reduced accordingly.

If this area is not large enough to contain the necessary information, Error Message NAT0886 is issued.

ESXDB - Database ID Used for Entire System Server DDMs

This Natural profile parameter is for Unix and Windows platforms only and applies to the Entire System Server Interface.

It specifies the database ID used for Entire System Server's DDMs.

Possible settings	0 or 1-254	Database ID. To activate this parameter, a database ID in the range of 1 to 254 must be specified. **Note:** Entire System Server's DDMs are cataloged with DBID=148. If you are using an Adabas database with this DBID, specify a different number for ESXDB. For information on how to do this, please refer to: • Setting up the Entire System Server Interface in the Natural Operations for Windows documentation, or • Setting up the Entire System Server Interface in the Natural for UNIX Installation documentation. With ESXDB=0, the Entire System Server Interface is not active.
Default setting	0	
Dynamic specification	NO	
Specification within session	NO	

ET - Execution of END/BACKOUT TRANSACTION Statements

This Natural profile parameter is for:

- Mainframes
- UNIX and Windows

It specifies for which databases END TRANSACTION and BACKOUT TRANSACTION statements are to be executed.

Possible settings	ON	END TRANSACTION and BACKOUT TRANSACTION statements are executed for all databases which have been referenced since the beginning of the Natural session or since the last execution of an END TRANSACTION and BACKOUT TRANSACTION statement.
	OFF	END TRANSACTION and BACKOUT TRANSACTION statements are executed only for the databases affected by the transaction (and - if applicable - for the database to which transaction data are written).
Default setting	OFF	
Dynamic specification	YES	
Specification within session	NO	

Note:
Any updates to a database which are not executed under the control of Natural (that is, by native invocation of the database link routines) do not affect the Natural transaction logic.

ETA - Error Transaction Program

This Natural profile parameter is for:

- Mainframes
- UNIX and Windows

It provides the name of the program which receives control if an error condition is detected during Natural program execution.

Possible settings	1 to 8 characters	Program name for error transaction.
	blank	With ETA=' ' (blank), no error transaction program is called.
Default setting	blank	
Dynamic specification	YES	
Specification within session	YES	

The setting of this parameter can be modified by a user program with the system variable *ERROR-TA.

Note: Error transaction programs must be in the library to which you are currently logged on.

When an error occurs, Natural executes a STACK TOP DATA statement and places the following information at the top of the stack:

- Error number (N4 if SG=OFF; N5 if SG=ON)
- Line number (N4)
- Status (A1)
- Program name (A8)
- Level (N2)

This information can be used as INPUT data by an error transaction. The status can be one of the following settings:

Setting	Explanation
C	Command processing error. (The line number will be "0".)
L	Logon processing error. (The line number will be "0".)
R	Error on Remote server (in conjunction with Natural RPC)
O	Object time error
S	Non-correctable Syntax error

If **Natural Security** is installed, the following rules apply:

- If an error occurs during the *first* logon to Natural, the program specified with the ETA parameter applies. The error transaction must be in SYSLIB at the time of the first logon.
- If an error occurs *after* the first logon, the program specified as error transaction in the security profile of the current library applies. If no error transaction is specified, the ETA parameter applies.

For additional information concerning the definition and use of error transaction programs, see Transactions (in the Natural Security documentation).

ETDB - Database for Transaction Data

This Natural profile parameter is for:

- Mainframes
- UNIX and Windows

It specifies the database in which transaction data, as supplied with an END TRANSACTION statement is to be stored.

Possible settings	1 - 65535, except 255	Database ID. Database ID 255 is reserved for logical system files for Software AG products, see profile parameter LFILE.)
	0	The transaction data are written to the database where the Natural Security system file (FSEC) is located. If FSEC is not specified, it is considered to be identical to the Natural system file FNAT (if Natural Security is not installed, the transaction data are written to the database where FNAT is located.
Default setting	0	
Dynamic specification	YES	
Specification within session	NO	

ETEOP - Issue END TRANSACTION at End of Program

This Natural profile parameter is for Mainframes, UNIX and Windows.

It determines whether or not an implicit END TRANSACTION statement is to be issued at the end of a Natural program (that is, before NEXT mode is reached).

Possible settings	ON	Natural will issue an implicit END TRANSACTION statement at the end of a Natural program.
	OFF	Natural will not issue any implicit END TRANSACTION statement at the end of a Natural program.
Default setting	OFF	
Dynamic specification	YES	
Specification within session	NO	

ETID - Adabas User Identification

This Natural profile parameter is for:

- Mainframes
- UNIX and Windows

It is used as an identifier for Adabas-related information; for example, for identification of data stored as a result of an END TRANSACTION statement.

Possible settings	1 to 8 characters	The setting is used as the user ID setting in an Adabas open call.
	OFF	Natural does not issue any Adabas open and close commands at the beginning of the Natural session. If, however, any ETID and/or OPRB specifications are present in Natural Security, these specifications are used in the subsequent open issued by Natural Security. This parameter setting is provided for use in conjunction with Natural Security to prevent Natural batch jobs that are sent at the same time from causing duplicate user ID settings in an Adabas open call during the initialization phase.
	' ' (blank)	If the ETID parameter is set to blanks, Natural does not issue any Adabas open and close commands; the OPRB parameter (if specified) and any ETID and OPRB specifications in Natural Security are ignored. In this case, you are recommended to set the Natural profile parameter DBCLOSE to ON to enforce a close command at session end. Otherwise, the user is not logged off from Adabas and the Adabas user queue element is not deleted. This may cause an overflow situation in the Adabas user queue.
Default setting	*INIT-USER	
Dynamic specification	YES	
Specification within session	NO	

If the ETID setting is **not** the same as the setting of the Natural system variable *INIT-USER, Natural issues an Adabas open with the specified ETID setting (and OPRB setting, if specified) at the beginning of the Natural session; this open remains in effect until the end of the Natural session; any ETID and OPRB specifications in Natural Security are ignored.

If the ETID setting is the same as the setting of *INIT-USER, or if the ETID parameter is not specified, Natural issues an Adabas open with the *INIT-USER setting as ETID (and the OPRB setting, if specified) at the beginning of the Natural session. If any Natural Security logon (initial logon or any subsequent logon) would change the currently valid ETID or OPRB setting (due to the library-/user-specific ETID and OPRB specifications in Natural Security), Natural Security issues a new open with the new ETID and OPRB settings. If the settings would remain the same after a logon, Natural Security does not issue a new open.

ETID and *INIT-USER can be modified by user exit NATUEX1 during session startup. See NATUEX1 - User Exit for Authorization Control (in the Natural Operations for Mainframes documentation).

ETIO - Issue END TRANSACTION upon Terminal I/O

This Natural profile parameter is for mainframes only.

It determines whether or not implicit END TRANSACTION statements are to be issued upon terminal I/Os.

Possible settings	ON	Natural will issue an implicit END TRANSACTION statement whenever a terminal I/O occurs. Whenever a transaction monitor commits the associated databases because of a terminal I/O, all related databases are also committed. This is useful for the synchronization of database transactions. **Note:** Natural utilities and Natural add-on products may not function correctly with ETIO=ON.
	OFF	Natural will issue no implicit END TRANSACTION statements upon terminal I/Os.
Default setting	OFF	
Dynamic specification	YES	
Specification within session	NO	

ETPDB - Databases Containing Entire Transaction Propagator Master Files

ETPDB - Databases Containing Entire Transaction Propagator Master Files

This Natural profile parameter is for UNIX and Windows platforms. It only applies if Software AG's Entire Transaction Propagator is installed.

ETPDB specifies the databases that contain Entire Transaction Propagator master files.

Possible settings	1-255	Database IDs; for details, see the Entire Transaction Propagator documentation. **Note for UNIX:** Mark all databases in the parameter's database list that contain master files. **Note for Windows:** The entries in the ETPDB field must be separated by blanks.
	0 or ' ' (blank)	Entire Transaction Propagator is not to be used.
Default setting	' ' (blank)	
Dynamic specification	NO	
Specification within session	NO	

ETPSIZE - Size of Entire Transaction Propagator Buffer

This Natural profile parameter is for:

- Mainframes
- UNIX and Windows

This parameter only applies if Entire Transaction Propagator is installed. It determines the size of the Entire Transaction Propagator buffer.

Alternatively, you can use the equivalent Natural profile parameter DS or macro NTDS (see Using Optional Macros in a Natural Parameter Module in the Natural Operations for Mainframes documentation) to specify the ETPSIZE value.

Possible settings	10 - 128	Size of the Entire Transaction Propagator buffer in KB. If Entire Transaction Propagator is to be used, an appropriate value has to be specified for this parameter; see the Entire Transaction Propagator documentation. If the requested space is not available, the Entire Transaction Propagator cannot be used.
	0	Entire Transaction Propagator is not to be used.
Default setting	0	
Dynamic specification	YES	
Specification within session	NO	

ETRACE - External Trace Function

This Natural profile parameter is for mainframes only.

It is used to activate/deactivate the (normal) external trace function or the Generalized Trace Facility (GTF) offered under OS/390 and TSO.

Do not use this parameter without prior consultation of Software AG Support.

The trace function is intended primarily for Software AG internal use for debugging purposes. It writes trace data to an external trace dataset depending on the TP environment in which Natural is running. In batch and TSO environments, a dataset (see also CMTRACE - Optional Report Output for Natural Tracing in the Natural Operations for Mainframes documentation) is required for the external trace.

Possible settings	ON	Activates the (normal) external trace function.
	OFF	Deactivates the (normal) external trace function.
	(ON,GTF) (OFF,GTF)	Activates/deactivates the Generalized Trace Facility (GTF). The trace records are written to the GTF.
	(ON,NOGTF) (OFF,NOGTF)	Activates/deactivates the (normal) external trace function.
	(,GTF)	Equivalent to ETRACE=GTF. Trace data is written to the GTF. ON or OFF is not altered.
Default setting	OFF	
Dynamic specification	YES	
Specification within session	YES	Within a Natural session, the **terminal command** %TRE can be used to activate/deactivate the external trace function, except GTF.

EXCSIZE - Size of Buffer for Natural Expert C Interface

EXCSIZE - Size of Buffer for Natural Expert C Interface

This Natural profile parameter is for mainframes only.

It determines the size of the buffer required by the C interface of Natural Expert. See the Natural Expert documentation.

Alternatively, you can use the equivalent Natural profile parameter DS or macro NTDS (see Using Optional Macros in a Natural Parameter Module in the Natural Operations for Mainframes documentation) to specify the EXCSIZE value.

Possible settings	1 - 256	Buffer size in KB.
	0	Natural Expert is not to be used.
Default setting	0	
Dynamic specification	YES	
Specification within session	NO	

EXRSIZE - Size of Buffer for Natural Expert Rule Tables

This Natural profile parameter is for mainframes only.

It determines the size of the buffer required by the rule tables of Natural Expert. See the Natural Expert documentation.

Alternatively, you can use the equivalent Natural profile parameter DS or macro NTDS (see Using Optional Macros in a Natural Parameter Module in the Natural Operations for Mainframes documentation) to specify the EXRSIZE value.

Possible settings	1 - 256	Buffer size in KB.
	0	Natural Expert is not to be used.
Default setting	0	
Dynamic specification	YES	
Specification within session	NO	

FAMSTD - Overwriting of Print and Work File Access Method Assignments

FAMSTD - Overwriting of Print and Work File Access Method Assignments

This Natural profile parameter is for mainframes only.

It controls the automatic overwriting of print and work file access method assignments during session initialization according to the dataset definition in the job control.

See also the AM subparameter of the macros NTPRINT and NTWORK.

Possible settings	ON	All print and work file data sets are automatically assigned to the batch access method AM=STD if the logical dataset name (defined by the DEST subparameter) is defined by job control (same behaviour as with Natural Version 2.2).
	OFF	Automatic print and work file assignment to AM=STD is done only if the file is not assigned to another access method, e.g. AM=NAF. If AM=OFF is specified, no automatic assignment is done. Specify AM=0 if you want to reset the access method type and to allow automatic assignment.
Default setting	OFF	
Dynamic specification	YES	
Specification within session	NO	

FC - Filler Character for INPUT Statement

This Natural profile parameter is for:

- Mainframes
- UNIX and Windows

It indicates the default filler character to be used for fields displayed by an INPUT statement.

Possible settings	any character	Default filler character. It is used to pre-fill fields non-protected input fields (field attribute specification AD=A) when fields are written to a terminal by an INPUT statement. For modifiable input fields (field attribute specification AD=M), it is used to fill the rest of the field.
Default setting	X’00’	For TTY or batch mode, the default setting is X’40’, i.e. blank in hexadecimal.
Dynamic specification	YES	
Specification within session	NO	

FCDP - Filler Character for Dynamically Protected Input Fields

This Natural profile and session parameter is for all platforms.

It allows you to suppress the display of filler characters for input fields that have been made write-protected dynamically (that is, to which the attribute AD=P has been assigned via an attribute control variable).

Depending on the setting of the FCDP parameter, dynamically protected input fields are displayed filled either with blanks or with the defined filler characters.

Within a Natural session, the profile parameter FCDP can be overridden by the session parameter FCDP.

Possible settings	ON	Dynamically protected input fields are displayed filled with filler characters. This may suggest to the users that they could enter something in the fields.	
	OFF	Dynamically protected input fields are displayed filled with blanks.	
Default setting	ON		
Dynamic specification	YES		
Specification within session	YES	Applicable Statements:	SET GLOBALS
		Applicable Command:	GLOBALS

For details on session parameter specification at statement or element level and evaluation at compilation time or at runtime, refer to Session Parameter Specification/Evaluation Overview.

Example:

```
DEFINE DATA LOCAL
1 #FIELD1 (A5)
1 #FIELD2 (A5)
1 #CVAR1  (C) INIT <(AD=P)>
1 #CVAR2  (C)
END-DEFINE
*
INPUT #FIELD1 (AD=Y'_' CV=#CVAR1)  /* field is protected
      #FIELD2 (AD=Y'_' CV=#CVAR2)  /* field is not protected
...
END
```

Execution of the above program will display the following:

FCDP=ON:

```
#FIELD1 _____ #FIELD2 _____
```

FCDP=OFF:

```
#FIELD1 #FIELD2 _____
```


FDDM - Natural System File for DDMs

This Natural profile parameter is for UNIX and Windows.

It defines five subparameters for the Natural system file for DDMs.

If this system file is defined, all DDMs are stored on the specified path. DDMs stored in libraries will no longer be accessible from Natural. This is similar to Natural for Mainframes, where all DDMs are stored in the system file FDIC.

If the FDDM system file is undefined (*database-ID* and *file-number* = 0), the DDMs are stored in the libraries as supplied before. The system file FDDM is displayed as an inactive environment.

For information on system files, refer to: .

Possible settings	*database-ID*	1 - 254	Database ID 255 is reserved for logical system files for Software AG products, see profile parameter LFILE.)
	file-number	1 - 255	File number.
	password	1 to 8 characters	The password is only required if the Natural user-program system file has been password-protected using the Adabas security feature.
	cipher key	8 characters	The cipher key is only required if the Natural user-program system file has been ciphered using the Adabas security feature.
	RO		"RO" indicates that the Natural user-program system file is "read-only" and is only specified if modifications on the file are to be disabled.
Default setting	None		
Dynamic specification	YES		
Specification within session	NO		

The syntax of this parameter is:

```
FDDM=(database-ID,file-number,password,cipher-key,RO)
```

Example:

```
FDDM=(22,5)
```


FDIC - Predict System File

This Natural profile parameter is for:

- Mainframes
- UNIX and Windows

This parameter defines five subparameters for the Predict system file which Predict uses to retrieve and/or store data. In a remote development environment, a Development Server File is used instead, see the Natural Single Point of Development documentation.

.

Possible settings	*database-ID*	1 - 65535, except 255	Database ID 255 is reserved for logical system files for Software AG products, see profile parameter LFILE.)
	file-number	1 - 65535	File number.
	password	1 - 8 characters	The password is only required if the Predict system file has been password-protected using the Adabas security feature. **Note for Natural with VSAM system files:** The password is used to specify the logical name (DD or DLBL) of the system file as defined to VSAM. Example: `FDIC=(10,5,SYSVSAM)` For further information, see Using Natural with VSAM System Files (in the Natural for VSAM documentation).
	cipher-key	1 - 8 characters	Cipher key for the Predict system file. It is only required if the Predict system file has been ciphered using the Adabas security feature.
	RO		"RO" indicates that the Predict system file is "read-only" and is only specified if modifications on the file are to be disabled.
Default setting	None		
Dynamic specification	YES		If you specify the FDIC parameter dynamically in conjunction with any of the parameters DBID, FNR, SYSPSW and SYSCIP, you must specify the FDIC parameter **after** any of these other parameters.
Specification within session	NO		

The syntax for this parameter is:

```
FDIC=(database-ID,file-number,password,cipher-key,RO)
```

Examples:

```
FDIC=(10,5,PASSW1,12345678)
FDIC=(1,200,,12345678)
FDIC=(1,5)
FDIC=(,5)
```


If any subparameter of the FDIC setting is not specified, the corresponding setting of the parameter DBID, FNR, SYSPSW or SYSCIP applies for the Predict system file.

FNAT - Natural System File for System Programs

This Natural profile parameter is for:

- Mainframes
- UNIX and Windows

This parameter indicates the database ID, file number, password and cipher key and read-only flag for the Natural system file for Natural system programs.

The Natural system file is the database file from which all Natural system programs are retrieved and upon which all system commands operate. Error texts and Natural help information are also contained in this system file.

Possible settings	*database-ID*	MF: 1 - 65535, except 255:MF]	Database ID 255 is reserved for logical system files for Software AG products, see profile parameter LFILE.)
	file-number	1 to 65535	File number.
	password	1 to 8 characters	The password is only required if the Natural system file has been password-protected using the Adabas security feature. **For Natural with VSAM system files:** The password is used to specify the logical name (DD or DLBL) of the system file as defined to VSAM. Example: `FNAT=(22,5,SYSVSAM)` For further information, see Using Natural with VSAM System Files (in the Natural for VSAM documentation).
	cipher key	8 characters	The cipher key is only required if the Natural system file has been ciphered using the Adabas security feature. The cipher key is reserved for future use; currently, it is ignored.
	RO		Read-only flag. RO indicates that the Natural system file is "read-only" and is only specified if modifications on the file are to be disabled.
Default setting	None		
Dynamic specification	YES		If you specify the FNAT parameter dynamically in conjunction with any of the parameters DBID, FNR, SYSPSW, SYSCIP or ROSY, you must specify the FNAT parameter **after** any of these other parameters.
Specification within session	NO		

The syntax of this parameter is:

```
FNAT=(database-ID,file-number,password,cipher-key,RO)
```


Examples:

```
FNAT=(,8)
FNAT=(22,5,PASSW2)
```

Notes:

- If any subparameter of the FNAT setting is not specified, the corresponding setting of the parameter DBID, FNR, SYSPSW, SYSCIP or ROSY applies for the Natural system file for system programs.
- If you reorganize an Adabas FNAT file or if you unload/load data from the FNAT file (e.g. using ADAULD/ADALOD), you must specify USERISN=YES for the ADALOD utility.

FNR - Default File Number of Natural System Files

This Natural profile parameter is for mainframes only.

It identifies the default number of the file in which the Natural system files (FNAT, FUSER, FDIC, FSEC, FSPOOL) are located.

Possible settings	1 - 65535	File number. It applies to all Natural system files for which no individual file numbers are specified.
Default setting	none	
Dynamic specification	YES	If you specify the FNR parameter dynamically in conjunction with any of the individual profile parameters which define the system files FNAT, FUSER, FDIC, FSEC and FSPOOL, you must specify the FNR parameter **before** any individual system file parameter.
Specification within session	YES	

File numbers for individual system files can be specified with the profile parameters FNAT, FUSER, FDIC, FSEC and FSPOOL.

Example 1:

```
FNR=5,DBID=10,FUSER=(,8)
```

This example assigns the user-program system file to File 8 on Database 10. All other system files are assigned to File 5 on Database 10.

Example 2:

```
FUSER=(,8),FNR=5,DBID=10
```

This example assigns all system files to File 5 on Database 10.

FREEGDA - Release GDA in Utility Mode

This Natural profile parameter controls whether current user global data area (GDA) and application-independent variables (AIV) are to be reset or not when a utility is invoked in utility mode (see Utility Activation in the Natural Utilities documentation), that is, by using the direct command that corresponds to the utility's name.

Possible settings	ON	The current user GDA and AIV variables are reset before a utility is started. This behavior corresponds to the previous situation when the utility was invoked using the system command LOGON <*library*>.
	OFF	The current user GDA and AIV variables are preserved when a utility is started. Note that this will increase the data size correspondingly and may lead to thread problems under certain operating systems.
Default setting	ON	
Dynamic specification	YES	
Specification within session	NO	

FS - Default Format/Length Setting for User-Defined Variables

This Natural profile and session parameter is for all platforms. It only applies to reporting mode; it has no effect in structured mode.

This parameter determines whether a default format/length setting is to be in effect for the definition of user-defined variables in reporting mode.

Within a Natural session, the profile parameter FS can be overridden by the session parameter FS.

Possible settings	ON	No default format/length is assigned by Natural for a newly introduced variable in reporting mode. The format/length of all user-defined variables must be explicitly specified.	
	OFF	A user-defined variable in a Natural program for which no format/ length is specified is assigned the default format/length "N7".	
Default setting	OFF		
Dynamic specification	YES		
Specification within session	YES	Applicable Statements:	SET GLOBALS
		Applicable Command:	GLOBALS

For details on session parameter specification at statement or element level and evaluation at compilation time or at runtime, refer to Session Parameter Specification/Evaluation Overview.

FSEC - Natural Security System File

This Natural profile parameter is for:

- Mainframes
- UNIX and Windows

It only applies if Natural Security is used.

This parameter defines five subparameters for the Natural Security system file which is used by Natural Security to retrieve/store its security information.

.

Possible settings	*database-ID*	1 - 65535, except 255	Database ID 255 is reserved for logical system files for Software AG products, see profile parameter LFILE.)
		0	DBID=0 sets FSEC inactive. This is mandatory for a non-security environment.
	file-number	For mainframes: 1 - 65535 For UNIX and Windows: 1 - 5000	File number for the Natural Security system file.
		0	FNR=0 sets FSEC inactive. This is mandatory for a non-security environment.
	password	1 to 8 characters	The password is only required if the Natural Security system file has been password-protected using the Adabas security feature.
	cipher-key	1 - 8 characters	The cipher key is only required if the Natural Security system file has been ciphered using the Adabas security feature.
	RO		Read-only flag. RO indicates that the Natural Security system file is "read-only" and is only specified if modifications on the file are to be disabled.
Default setting	None		
Dynamic specification	YES		If you specify the FSEC parameter dynamically in conjunction with any of the parameters DBID, FNR, SYSPSW, SYSCIP or ROSY, you must specify the FSEC parameter **after** any of these other parameters.
Specification within session	NO		

The syntax of this parameter is:

```
FSEC=(database-ID,file-number,password,cipher-key,RO)
```


Example:
```
FSEC=(10,8)
```

Note:
If any subparameter of the FSEC setting is not specified, the corresponding setting of the parameter DBID, FNR, SYSPSW, SYSCIP or ROSY applies for the Natural Security system file.

FSPOOL - Natural Advanced Facilities Spool File

This Natural profile parameter is for mainframes only.

It only applies to Natural Advanced Facilities. It defines five subparameters for the Natural Advanced Facilities spool file. The spool file is the database file that is used by Natural Advanced Facilities. This file must be different from the FNAT, FUSER, FDIC and FSEC system files.

Possible settings	*database-ID*	1 - 65535, except 255	Database ID 255 is reserved for logical system files for Software AG products, see profile parameter LFILE.) If any component of the FSPOOL setting is not specified, the corresponding setting of the parameter DBID, FNR, SYSPSW or SYSCIP applies for the spool file.
	file-number	1 - 65535	Database file number.
	password	1 - 8 characters	The password is only required if the spool file has been password-protected using the Adabas security feature. **Note for Natural with VSAM System Files:** The password is used to specify the logical name (DD or DLBL) of the system file as defined to VSAM. Example: `FSPOOL=(10,8,SYSVSAM)` For further information, see Using Natural with VSAM System Files (in the Natural for VSAM documentation).
	cipher-key	8 characters	The cipher key is only required if spool file has been ciphered using the Adabas security feature.
	RO		"RO" indicates that the Natural Advanced Facilities spool file is "read-only" and is only specified if modifications on the file are to be disabled. This would mean, for example, that no reports could be stored on the spool file.
Default setting	None		
Dynamic specification	YES		If you specify the FSPOOL parameter dynamically in conjunction with any of the parameters DBID, FNR, SYSPSW or SYSCIP, you must specify the FSPOOL parameter **after** any of these other parameters.
Specification within session	NO		

The syntax for this parameter is:

```
FSPOOL=(database-ID,file-number,password,cipher-key,RO)
```

Example:

FSPOOL - Natural Advanced Facilities Spool File

```
FSPOOL=(10,8)
```

FUSER - Natural System File for User Programs

This Natural profile parameter is for:

- Mainframes
- UNIX and Windows

It defines five subparameters for the Natural user-program system file. This system file is the database file from which all user-written Natural programs are retrieved.

.

Possible settings	*database-ID*	1 - 65535, except 255	Database ID 255 is reserved for logical system files for Software AG products, see profile parameter LFILE.)
	file-number	1 to 65535	File number.
	password	1 to 8 characters	The password is only required if the Natural user-program system file has been password-protected using the Adabas security feature. **Note for Natural with VSAM System Files:** The password is used to specify the logical name (DD or DLBL) of the system file as defined to VSAM. Example: `FUSER=(22,5,SYSVSAM)` For further information, see Using Natural with VSAM System Files (in the Natural for VSAM documentation).
	cipher key	8 characters	The cipher key is only required if the Natural user-program system file has been ciphered using the Adabas security feature.
	RO		"RO" indicates that the Natural user-program system file is "read-only" and is only specified if modifications on the file are to be disabled.
Default setting	None		
Dynamic specification	YES		If you specify the FUSER parameter dynamically in conjunction with any of the parameters DBID, FNR, SYSPSW, SYSCIP or ROSY, you must specify the FUSER parameter **after** any of these other parameters.
Specification within session	NO		

The syntax of this parameter is:

```
FUSER=(database-ID,file-number,password,cipher-key,RO)
```

Examples:


```
FUSER=(,8)
FUSER=(22,5,PASSW2)
```

Note:
If any subparameter of the FUSER setting is not specified, the corresponding setting of the parameter DBID, FNR, SYSPSW, SYSCIP or ROSY applies for the Natural user-program system file.

GFID - Global Format IDs

This Natural profile parameter is for UNIX and Windows platforms only.

It allows you to control Natural's internal generation of global format IDs so as to influence Adabas's performance concerning the re-usability of format buffer translations.

Possible settings	ON	Global format IDs are generated for all views.
	OFF	Global format IDs are not generated.
	VID	Global format IDs are generated only for views in local/global data areas, but not for views defined within programs.
Default setting	ON	
Dynamic specification	YES	
Specification within session		

For details on global format IDs, see the Adabas documentation.

HCAM - Hardcopy Access Method

This Natural profile parameter is for mainframes only.

It determines which access method is to be used for hardcopy output processing. HCAM=*xxx* is equivalent to the AM subparameter of the profile parameter PRINT for Print File 0, that is, `PRINT=((0),AM=xxx)`.

You can specify one of the following access-method names:

Possible settings	Value:	Access method:
	STD	Standard sequential file (batch, TSO, TIAM, VM/CMS OS simulation).
	COMP	Com-plete print file.
	CMS	VM/CMS disk and SFS files.
	CICS	CICS transient data or temporary storage.
	IMS	IMS/TM printer.
	NAF	Natural Advanced Facilities.
	USER	Third-party vendor print interface.
	SMARTS	SMARTS print file.
	ESS	Entire System Server.
	ANY	Hardcopy output processing will be handled by the first access method available (the search sequence for available access methods is the sequence in which the access methods are listed here).
	OFF	Hardcopy output processing will not be handled by any access method.
Default setting	ANY	
Dynamic specification	YES	
Specification within session	NO	

The hardcopy output destination is specified using the profile parameter HCDEST. More specifications for the hardcopy output file can be made using the PRINT profile parameter or the NTPRINT macro for Printer 0.

Note for BS2000/OSD Users:
HCAM=STD is a necessary setting for routing hardcopy output to standard print files.

HCDEST - Hardcopy Output Destination

This Natural profile parameter is for mainframes only.

It presets the hardcopy output destination for the terminal command %H (without the *destination* operand). HCDEST=*xxx* is equivalent to the DEST subparameter of the profile parameter PRINT for Print File 0, that is, `PRINT=((0),DEST=xxx)`.

Possible settings	1 to 8 characters	Valid hardcopy output destination.
	blank	
Default setting	blank	In some environments, a default destination may be supplied by the operating system or TP monitor. If HCAM=STD is assigned for hardcopy, the default hardcopy output destination is the dataset CMHCOPY.
Dynamic specification	YES	
Specification within session	YES	The hardcopy output destination can be overwritten during the session by specifying "%H*destination"*; see also the terminal command %H.

If you are running Natural under TSO or in batch mode, the dataset must be defined in the JCL or by dynamic allocation.

Under TSO, the hardcopy dataset specified by HCDEST is closed after %H at the next terminal I/O. The default CMHCOPY dataset is closed not at terminal I/O, but at session termination.

The hardcopy output access method can be specified by profile parameter HCAM or by the DEST subparameter of profile parameter PRINT for Printer 0. More specifications for the hardcopy output file can be made using the profile parameter PRINT or the macro NTPRINT for Printer 0.

HI - Help Character

This Natural profile parameter is for:

- Mainframes
- UNIX and Windows

It defines the character which is to be used to invoke a field-specific helproutine or a map helproutine (if defined for a given map).

Possible settings	any special character	The character which is to be used to invoke a field-specific helproutine or a map helproutine. The character specified with the profile parameter HI must not be the same as the one specified with the profile/session parameter CF (control character for mainframe terminal commands); it should not be the same as the one specified with the profile/session parameter DC (decimal character), profile/session parameter IA (input assign character) or profile/session parameter ID (input delimiter character).
	blank	Numeric fields which have a helproutine assigned are internally translated to alphanumeric format so as to make it possible for the user to enter a question mark into the field to invoke the helproutine. To prevent this internal translation (that is, if you wish to make sure that alphabetical characters cannot be entered into a numeric field) you can set the profile parameter HI to blank. However, when HI=' ' is set, a help key must be defined in the Natural application, using the SETKEY statement correspondingly; otherwise it is not possible to invoke a helproutine for any field.
Default setting	?	Question mark.
Dynamic specification	YES	
Specification within session	NO	

IA - Input Assign Character

This Natural profile and session parameter is for all platforms

It defines the character to be used as the assignment character for the input parameter processing in INPUT statements, either in keyword/delimiter mode or when processing data from the Natural stack.

Within a Natural session, the profile parameter IA can be overridden by the session parameter IA.

Possible settings	any special character	Assignment character for the input parameter processing in INPUT statements. The character specified with the IA parameter • must not be the same as the character specified with the profile/session parameters CF (control character for mainframe terminal commands), DC (decimal character), STACKD profile parameter (stack delimiter character) or ID (input delimiter character); • should not be the same as the one specified with the profile parameter HI (help character).	
Default setting	’=’		
Dynamic specification	YES		
Specification within session	YES	Applicable Statements:	SET GLOBALS
		Applicable Command:	GLOBALS

For details on session parameter specification at statement or element level and evaluation at compilation time or at runtime, refer to Session Parameter Specification/Evaluation Overview.

Example:

```
* Program ’SAMPLE’
DEFINE DATA LOCAL
1 #A (A1)
1 #B (A1)
END-DEFINE
INPUT #A #B
WRITE #A #B
END
```

1. Enter the command `GLOBALS IM=D`
   This sets the input mode to "Delimiter Mode".
2. Then enter the command `SAMPLE #A=Y,#B=X`
   The program produces the following output: `Y X`
3. Enter the command `GLOBALS IA=:`
   This sets the input assign character to "colon".
4. Then enter the command `SAMPLE #B:X,#A:Y`
   The program produces the following output: `Y X`

ID - Input Delimiter Character

This Natural profile and session parameter is for all platforms.

It defines the character to be used as a delimiter character for INPUT statements in keyword/delimiter mode.

Within a Natural session, the profile parameter ID can be overridden by the session parameter ID.

Possible settings	any special character	Input delimiter character. The character specified with this parameter • must not be the same as the one specified with the profile/session parameter DC (decimal character), STACKD (stack delimiter character) or IA (input assign character); • should not be the same as the one specified with the CF parameter (control character for mainframe terminal commands) or HI parameter (help character). The period (**.**) should not be used as input delimiter, because this might lead to situations in which a program termination period would be misinterpreted as input delimiter. An asterisk (*****) should not be used either.	
	blank	No input delimiter character is defined.	
Default setting	, (comma)	If the input delimiter character is to be a comma, it must be specified as ID=',' when using the dynamic parameter facility, because the character "," separates individual parameters.	
Dynamic specification	YES		
Specification within session	YES	Applicable Statements:	SET GLOBALS
		Applicable Command:	GLOBALS

For details on session parameter specification at statement or element level and evaluation at compilation time or at runtime, refer to Session Parameter Specification/Evaluation Overview.

IKEY - Processing of PA and PF Keys

This Natural profile parameter is for:

- Mainframes
- UNIX and Windows

It indicates the action to be taken when a video-terminal program-attention key (PA key) or program-function key (PF key) is used to enter data, and the key has not been defined to the Natural program with the SET KEY statement.

Possible settings	ON	The setting "ENTR" is placed in the Natural system variable *PF-KEY; that is, Natural reacts as if ENTER had been pressed.
	OFF	A REINPUT message is generated, prompting the user to press a valid key.
Default setting	OFF	
Dynamic specification	YES	
Specification within session	NO	

IM - Input Mode

This Natural profile and session parameter is for all platforms.

It determines the default mode for video-terminal input.

Within a Natural session, the setting of the profile parameter IM can be overridden by the session parameter IM.

Possible settings	F	Forms mode.	
	D	Delimiter mode.	
Default setting	F		
Dynamic specification	YES		
Specification within session	YES	Applicable Statements:	SET GLOBALS
		Applicable Command:	GLOBALS

For details on session parameter specification at statement or element level and evaluation at compilation time or at runtime, refer to Session Parameter Specification/Evaluation Overview.

Notes:

- The IM parameter setting may also be changed with the Natural terminal commands %D and %F.
- For information on delimiter mode and forms mode, see the INPUT statement.

IMSG - Session Initialization Error Messages

This Natural profile parameter is for mainframes only.

It is used to suppress the initialization error-messages screen. It can be useful to avoid undesired output, for example, for printer sessions.

As error diagnosis may become difficult, use this parameter with caution.

Possible settings	ON	The initialization error messages screen is displayed in the case of an error.
	OFF	The initialization error messages screen is not displayed.
Default setting	ON	
Dynamic specification	YES	
Specification within session	NO	

INIT-LIB - Library for Automatic Logon

This Natural profile parameter is for UNIX and Windows platforms only.

It specifies the name of the library to be used for an automatic logon (see the profile parameter AUTO) when Natural is started.

Possible settings	1-8 characters	Valid library name.
Default setting	none	
Dynamic specification	NO	
Specification within session	NO	

If Natural Security is installed, INIT-LIB is not evaluated; the library to be used for automatic logon is read from the FSEC system file (see the Natural Security documentation for further information).

INTENS - Printing of Intensified Fields

This Natural profile parameter is for mainframes only.

It indicates how many times an intensified field or the underline character is to be overprinted when it is printed on a print device.

Possible settings	1 - 10	Number of times an intensified field or the underline character is overprinted. The underline character is printed only if the parameter is set greater than 1. With INTENS=1, underlined fields are printed without underlining.
Default setting	3	
Dynamic specification	YES	
Specification within session	NO	

ISIZE - Size of Initialization Buffer

This Natural profile parameter is for mainframes only.

It determines the size of the Natural initialization buffer. This buffer is used to hold the parameters Natural is initialized with, as well as the work areas and tables used by Natural during the initialization.

Possible settings	8-32	Buffer size in KB.
Default setting	10	
Dynamic specification	YES	
Specification within session	NO	

Note:
The profile parameter ISIZE is ignored if it is specified in a parameter string activated by a SYS or PROFILE profile parameter or in an alternative parameter module (as specified with the PARM profile parameter).

ITERM - Session Termination in Case of Initialization Error

This Natural profile parameter is for mainframes only.

It determines whether or not the Natural session is to continue in the case of a session initialization error.

Possible settings	ON	If a session initialization error occurs, the session is terminated immediately after the initialization error messages.
	OFF	If session initialization errors occur, the following happens: **In online mode:** the initialization errors are displayed and you can choose to either continue or terminate the session. **In batch mode:** the session is continued with the initialization errors going unnoticed - possibly leading to errors or undesired results later in the session. The setting ITERM=OFF is not possible when an INPL command is placed on the Natural command stack at the beginning of the Natural session, that is, with STACK=INPL.
Default setting	OFF	
Dynamic specification	YES	
Specification within session	NO	

ITRACE - Internal Trace Function

This Natural profile parameter is for mainframes only.

Do not use this parameter without prior consultation of Software AG Support.

It is used to activate/deactivate the internal trace function. The internal trace function is intended primarily for Software AG internal use for debugging purposes.

Possible settings	ON	Trace data is passed to the SYSRDC utility.
	OFF	No trace data is passed to the SYSRDC utility.
Default setting	ON	
Dynamic specification	YES	
Specification within session	YES	Within a Natural session, the terminal command "%TRI" can be used to activate/ deactivate the internal trace function.

KAPRI - Kanji Printing

This Natural profile parameter is for mainframes only.

It is used to specify parameters for Kanji printing. It corresponds to the NTKAPRI macro in the parameter module NATPARM.

For further information, please refer to the Kanji print interface documentation (available from Software AG of Far East).

Possible settings	See below.	Refer to the Kanji print interface documentation.
	OFF	If you do not wish to use the parameter list of the NTKAPRI macro, you specify KAPRI=OFF.
Default setting	None	
Dynamic specification	YES	This parameter can only be specified dynamically. In the Natural parameter module NATPARM, the macro NTKAPRI must be used instead.
Specification within session	NO	

The following topics are covered below:

- KAPRI Parameter Syntax
- NTKAPRI Macro Syntax

KAPRI Parameter Syntax

The KAPRI parameter is specified as follows:

```
KAPRI= 'parameter1,parameter2,parameter3,...'
```

The parameters specified with the KAPRI parameter will be appended to the list of parameters specified with the NTKAPRI macro. With a subsequent KAPRI parameter, you can then specify a new parameter list.

NTKAPRI Macro Syntax

The NTKAPRI macro is specified as follows:

```
NTKAPRI 'parameter1,parameter2,parameter3,...'
```

If multiple NTKAPRI macros are specified, the parameter lists are concatenated to constitute a single list.

KC - Check for Statement Keywords

This Natural profile parameter is for UNIX and Windows platforms only.

Note for mainframes: Use the keyword subparameter KCHECK of the profile parameter CMPO.

The profile parameter KC checks Natural programs during compilation for Natural statement keywords which are invalidly used as variable names.

Possible settings	ON	The check for statement keywords is performed.
	OFF	No check for statement keywords is performed.
Default setting	OFF	
Dynamic specification	YES	
Specification within session	NO	

The document Keywords and Reserved Words (in the Natural Programming Guide) contains a list of all Natural keywords and reserved words, in which the statement keywords affected by the KC parameter are marked.

KEY - Setting Assignments to PA, PF and CLEAR Keys

This Natural profile parameter is for:

- Mainframes
- UNIX

It is used to assign settings to the CLEAR key, program attention keys (PA keys) and program function keys (PF keys) on video terminals.

Possible settings	any character string	Settings can be assigned to the keys PA1 to PA3, PF1 to PF24 and to the CLEAR key. The setting assigned to each key can be any character string. The character string must represent a Natural system command or a user command (user program). If the setting contains embedded blanks, it must be enclosed in apostrophes.
Default setting	none	
Dynamic specification	NO	
Specification within session	YES	

Assignments made with the profile parameter KEY are only valid from the Natural NEXT prompt (Mainframe).

The entire string specified with the profile parameter KEY must be enclosed in parentheses (except KEY=OFF). KEY=OFF un-assigns all keys.

Examples:

```
KEY=(PF4=OFF,PF1=HELP,PF3='EDIT MAP',PF2=USERPGM1,CLR=LOGOFF)

KEY=OFF
```


LC - Lower to Upper Case Translation

This Natural profile parameter is for:

- Mainframes
- UNIX and Windows

It controls lower-case to upper-case translation of input characters.

Note:
This parameter does not apply to Natural stack data which was placed on the Natural stack by the STACK statement.

Possible settings	ON	No translation of lower-case characters to upper case is performed.
	OFF	All lower-case characters, except input from the Natural stack which was place there by the STACK statement, is translated to upper case by Natural.
Default setting	OFF	
Dynamic specification	YES	
Specification within session	YES	To disable or enable lower-case to upper-case translation dynamically within the active Natural session, you should use the terminal commands %L or %U

Notes:

- Lower/upper-case translation can also be performed by a TP monitor before control is given to Natural. The corresponding TP-monitor parameters for lower/upper-case translation also have to be reviewed to ensure correct translation.
- A user-supplied translation table can be used to perform translation from lower case to upper case; see NTUTAB1 macro (contained in the UTAB1 profile parameter description).

LDB - Wait Time for Response of Local Adabas Database

This Natural profile parameter is for Windows platforms only.

It specifies the time limit Natural is to wait for a response of the local Adabas database.

Possible settings	1 - 999	Time limit in seconds. If it is exceeded, an appropriate error message is issued.
	0	No time limit will be in effect.
Default setting	30	
Dynamic specification	NO	
Specification within session	NO	

If Entire Net-Work is installed, the specified time limit also affects Entire Net-Work’s timeout processing.

LE - Reaction when Limit for Processing Loop Exceeded

This Natural profile and session parameter is for all platforms.

It controls the action to be taken if the limit for processing loop execution is exceeded. The limit may be either a global limit (specified with the profile/session parameter LT) or a limit specified for a given processing loop.

Within a Natural session, the profile parameter LE can be overridden by the session parameter LE.

Possible settings	ON	The Natural program will be terminated normally at the point where the limit was reached. Loop execution is aborted and an error message is issued at the end of the Natural program. LE=ON applies only to programs which are loaded from a library located in the system file FUSER, that is, library SYSTEM, or with a (library) name that does not start with the prefix "SYS".	
	OFF	Loop execution is aborted and processing continues without an error message. The Natural program will be terminated normally at the point where the limit was reached.	
Default setting	OFF		
Dynamic specification	YES		
Specification within session	YES	Applicable Statements:	SET GLOBALS
		Applicable Command:	GLOBALS

For details on session parameter specification at statement or element level and evaluation at compilation time or at runtime, refer to Session Parameter Specification/Evaluation Overview.

LFILE - Logical System File Definition

This Natural profile parameter is for:

- Mainframes
- UNIX and Windows

It specifies information concerning the physical database file to be associated with a logical system file for Software AG products.

It can be used for Software AG products which have their own system files (for example, Con-nect and Natural Elite) to specify where such a system file is to be located. Such products use Database ID 255 and a logical file number (FNR) in their data definition modules (DDMs). With the LFILE parameter or the macro NTLFILE, you specify which physical database ID (DBID) and file number (and, if applicable, password and cipher key) are associated with that logical file number.

Natural records the physical file information and uses it for any database calls to Database ID=255 and File number=*logical-ID*.

Possible settings	*logical-FNR*	1 - 251	Logical file number (LFL). This parameter is mandatory.
	physical-DBID	0 - 65535, except 255	Physical database ID (DBID). Database ID 255 is reserved for logical system files for Software AG products.
	physical-FNR	1 - 65535	Physical file number (FNR).
	password		Must be a setting of 1 to 8 characters. *
	cipher-key		Must be a setting of 8 characters. *
	options		Can be RO for read-only access.
Default setting	None		
Dynamic specification	YES		This parameter can only be specified dynamically. In the Natural parameter module NATPARM, the macro NTLFILE must be used instead. It replaces the old macro NTFILE for logical system file definition which is still available, but should not be used any longer.
Specification within session	NO		

* Password and cipher key are only required if the database file has been password-protected and/or ciphered using the Adabas security feature.

Note: LFILE can also be used to define a so-called scratch-pad file with logical file number "212"; see also the profile parameter ROSY and refer to Natural Scratch-Pad File in the Natural Operations for Mainframes documentation.

To define different logical files, the LFILE parameter or the macro NTLFILE must be specified several times.

The following topics are covered below:

LFILE Parameter Syntax

The LFILE parameter is specified as follows:

```
LFILE=(logical-FNR,physical-DBID,physical-FNR,password,cipher-key,RO)
```

NTLFILE Macro Syntax

In contrast to the former NTFILE macro which has keyword subparameters, the NTLFILE macro has positional subparameters (like the LFILE parameter) and is specified as follows:

```
NTLFILE logical-FNR,physical-DBID,physical-FNR,password,cipher-key,RO)
```

Old NTFILE Macro Syntax

For compatibility reasons, the old macro NTFILE is still supported. It is specified as follows:

```
NTFILE
ID=logical-FNR,DBID=physical-DBID,FNR=physical-FNR,PASSW=password,CIPH=cipher-key,OPT=RO)
```

Example of LFILE Parameter

```
LFILE=(180,73,10),LFILE=(251,40,9,TEST99)
```

Example of NTLFILE Macro

Equivalent specification in the Natural parameter module:

```
NTLFILE 180,73,10NTLFILE 251,40,9,TEST99
```


LFILMAX - Maximum Number of Dynamic Specifications of LFILE Parameter

LFILMAX - Maximum Number of Dynamic Specifications of LFILE Parameter

This Natural profile parameter is for UNIX and Windows platforms only.

It can be used to specify the maximum number of dynamic specifications of the profile parameter LFILE.

Possible settings	0 - 255	Maximum number of dynamic specifications of the profile parameter LFILE.
Default setting	8	
Dynamic specification	NO	
Specification within session	NO	

LIBNAM - Name of External Program Load Library

This Natural profile parameter is for mainframes and only applies under BS2000/OSD, OS/390 batch mode, and TSO.

It specifies the name of the load library from which programs are to be loaded dynamically when Natural is used under BS2000/OSD, OS/390 batch mode, or TSO.

Possible settings	character string	Any valid BS2000/OSD file name, or 8-byte DDNAME of load library
Default setting	none	
Dynamic specification	YES	
Specification within session	NO	

Under OS/390, a JCL statement with a DDNAME that equals the LIBNAM setting also needs to be specified. By default, programs are loaded from the job steplib.

LOG (Internal Use)

This Natural profile parameter is for mainframes only.

This parameter is reserved for internal use by Natural. Do not change its setting.

LOGONRQ - Logon for RPC Server Request Required

For static specification, this parameter is available as a keyword subparameter of the NTRPC macro. For dynamic specification, this parameter is available as a subparameter of the profile parameter RPC.

LOGONRQ determines whether or not logon data are required for an RPC server request.

This parameter is specified on the server side only.

Possible settings	ON	A logon is required; that is, the server only accepts requests from clients which include logon data in the RPC server request. For conversational requests, the logon data is only necessary when the conversation is opened.
	OFF	A logon is **not required**. Logon data is nevertheless processed.
Default setting	OFF	
Dynamic specification	YES	
Specification within session	NO	

For Natural clients the logon data can either be requested:

- by setting the LOGON option of the SYSRPC Service Directory Maintenance;
- using the logon indicator of the profile parameter DFS.

You are strongly recommended to set LOGONRQ=ON if the Natural RPC server runs under Natural Security. For further information, see Using Natural RPC with Natural Security in the Natural Remote Procedure Call documentation.

For additional information on Natural RPC, see the Natural Remote Procedure Call documentation.

LS - Line Size

This Natural profile and session parameter is for all platforms.

It specifies the maximum number of characters permitted per line for DISPLAY, INPUT and WRITE statements.

The following topics are covered below:

- Profile Parameter LS
- Session Parameter LS
- Specification wthin Statements

Profile Parameter LS

When used as a profile parameter, LS is honored in batch mode only and defines the physical line size. In online mode, the line size is always set to the physical screen width.

Possible settings	35 - 250	Maximum number of characters permitted per line.
	0	Use physical line size (mostly 132).
Default setting	0	
Dynamic Specification	YES	

Session Parameter LS

Possible settings	2 - 250	Maximum number of characters permitted per line.
Default setting	0	Physical line size.
Applicable Command:		GLOBALS
Applicable Statements:		FORMAT SET GLOBALS

Note:
At logon to a library, LS is reset to the physical line size.

Specification within Statements

When specified within a statement, the LS parameter is evaluated at compilation time.

Applicable Statements:	DISPLAY INPUT WRITE	Parameter may be specified at statement level.

LSTEP - Initial Setting for *STEPLIB System Variable

This Natural profile parameter is for UNIX and Windows platforms only.

It specifies the initial setting for the *STEPLIB system variable; see also System Variables.

Possible settings	1-8 characters	Any valid library name.
Default setting	STEP	
Dynamic specification	NO	
Specification within session	NO	

All existing steplibs are displayed. In addition, you can specify the name, DBID and FNR of the library where the Natural steplib is located. Up to eight steplibs can be defined.

LT - Limit for Processing Loops

This Natural profile and session parameter is for all platforms.

It limits the number of database records which can be read in processing loops within a Natural program. The action to be taken if the limit for processing loop execution is exceeded is specified with the profile/session parameter LE.

Within a Natural session, the profile parameter LT can be overridden by the session parameter LT.

Possible settings	0 - 2147483647 Within a session: 0 to *n* (*n* = value of profile parameter LT at session start)	Maximum number of records which can be read in any given processing loop. This limit applies to any processing loop initiated with a READ, FIND or HISTOGRAM statement. All records read (including rejected records from a WHERE clause) are counted against this limit. LT=0 indicates that the parameter is of an infinite length. If a limit for a single processing loop (as set with a LIMIT statement or a limit notation) is higher than the limit set with the LT parameter, the limit set with the LT parameter applies.	
Default setting	99999999		
Dynamic specification	YES		
Specification within session	YES	Applicable Statements:	SET GLOBALS
		Applicable Command:	GLOBALS

For details on session parameter specification at statement or element level and evaluation at compilation time or at runtime, refer to Session Parameter Specification/Evaluation Overview.

MADIO - Maximum DBMS Calls between Screen I/O Operations

This Natural profile parameter is for:

- Mainframes
- UNIX and Windows

It indicates the maximum number of DBMS calls permitted between two screen I/O operations (also in batch mode).

Possible settings	30 - 32767	Maximum number of DBMS calls.
	0	MADIO=0 indicates that no limit is to be in effect.
Default setting	512	
Dynamic specification	YES	
Specification within session	NO	

If the specified limit is exceeded, the Natural program is interrupted and the user is notified with Natural Error Message 1009.

MAINPR - Override Default Output Report Number

This Natural profile parameter is for:

- Mainframes
- UNIX and Windows

Possible settings	0 - 31	Valid printer number.
Default setting	0	
Dynamic specification	YES	
Specification within session	NO	

With this parameter, you can separate program output from Natural system output, which may be useful particularly in batch mode.

This applies to program output for Report 0, as produced by DISPLAY, PRINT, WRITE or INPUT statements (except INPUT statements which contain non-protected input fields (field attribute specification AD=A) or modifiable input fields (AD=M)).

If the MAINPR parameter is specified, program output for Report 0, which would normally be output on the printer assigned to Report 0, is output on the printer specified with MAINPR instead; while system output (NEXT prompt, DATA prompt, etc.) is always output on the primary output device (Report 0); the MAINPR setting must be a valid printer number (0 - 31).

A logical printer corresponding to the report number specified must be defined to Natural. A printer is defined with the profile parameter PRINT, with the macro NTPRINT or automatically by JCL (in batch mode or under TSO).

The MAINPR parameter does not apply to output from system programs in the Natural system library SYSLIB, which is always output on the primary output device (Report 0).

MAXBUFF - Maximum Buffer Size

For static specification, this parameter is available as a keyword subparameter of the NTRPC macro. For dynamic specification, this parameter is available as a subparameter of the profile parameter RPC.

MAXBUFF can be specified on both the client and the server side.

On the server side, it determines the size of the buffer provided by the server to receive the client request including data and to send back the result. During the execution of the remote CALLNAT, this buffer also contains the PDA passed to the CALLNAT. The buffer must be large enough to hold the largest of the following three data areas for all client requests:

- the request received by the client,
- the internal PDA passed to the CALLNAT,
- the result send back to the client.

On the client side, it determines the size of the buffer provided for the automatic execution of Natural RPC calls. This buffer is used to build the client request including data and to receive the result from the server. The buffer must be large enough to hold the largest of the following two data areas for all requests sent by the client:

- the request send to the server,
- the result received from the server.

For further information, see Stubs and Automatic RPC Execution in the Natural Remote Procedure Call documentation.

The size of the data exchanged between the client and server is provided by the stub generation function of the SYSRPC utility. To calculate the size for automatic RPC execution, you must also use the stub generation function and delete the generated stub afterwards.

Possible settings	0-2097147, but smaller than or equal to RPCSIZE-4	Maximum buffer size in KB. The maximum buffer size must be equal to or less than the value (minus 4) specified with the profile parameter RPCSIZE, for example: RPCSIZE=128 => MAXBUFF =< 124.
Default setting	0	No buffer is allocated.
Dynamic specification	YES	
Specification within session	NO	

In case of an EntireX Broker node, special considerations apply if you are using Entire Net-Work as a transport layer. With Entire Net-Work, the receive buffer length passed to the EntireX Broker stub is restricted by the startup parameter IUBL and must not exceed 32 KB. To be able to use PDAs that are larger than 32 KB the receive buffer length is unbundled from the MAXBUFF setting.

Depending on the setting of the Natural profile parameter ACIVERS, the receive buffer length is set as follows:

- ACIVERS=1: 32000
- ACIVERS=2: 30K
- ACIVERS>2: the value specified with MAXBUFF

With ACIVERS=1 and ACIVERS=2, you can therefore specify a value for MAXBUFF that is not accepted by the EntireX Broker stubs. This may be useful when using input and output fields. In this case, the size of the input data and the size of the output data, each counted separately, may be less than the limit, but the sum of both

sizes, which must fit into the buffer, exceeds the limit.

For further information, see the Natural Remote Procedure Call documentation.

MAXCL - Maximum Number of Program Calls

This Natural profile parameter is for:

- Mainframes
- UNIX and Windows

It determines the maximum number of program calls permitted between two screen I/O operations. If the specified limit is exceeded, the Natural program is interrupted and and the user is notified with an appropriate Natural error message (Mainframe: NAT1029).

Possible settings	10 - 32767	Maximum number of program calls.
	0	MAXCL=0 indicates that no limit is to be in effect.
Default setting	50	
Dynamic specification	YES	
Specification within session	NO	

MAXROLL - Number of CMROLL Calls Before Roll-Out

MAXROLL - Number of CMROLL Calls Before Roll-Out

This Natural profile parameter is for mainframes and applies under Com-plete and CICS only.

It determines the number of CMROLL calls after which a roll-out of the Natural thread is to be performed.

Possible settings	1 - 32767	Number of CMROLL calls.
	0	MAXROLL=0 indicates that no conditional CMROLL requests are issued.
Default setting	128	
Dynamic specification	YES	
Specification within session	NO	

The MAXROLL parameter can be used to control the frequency of conditional CMROLL requests. For example, MAXROLL=128 means that a conditional CMROLL request is issued after every 128th statement at compilation.

In certain cases, the Natural nucleus issues a conditional CMROLL request (wait time = 0), particularly at compilation after each statement. This is done to reset the CPU time window (under Com-plete) in order to avoid an automatic cancel due to the CPU time limit being exceeded; however, this has a negative impact on performance.

MAXUSER - Maximum Number of Users

This Natural profile parameter is for UNIX and Windows platforms only.

It determines the maximum number of users that can have simultaneous access to the buffer pool. This number determines the sizes of some internal tables stored inside the shared memory during startup.

Possible settings	1 - 5000	Maximum number of users.
Default setting	20	
Dynamic specification	NO	
Specification within session	NO	

MENU - Menu Mode

This Natural profile parameter is for mainframes only.

It is used to enable or disable Natural menu mode.

Possible settings	ON	Menu mode is enabled.
	OFF	Menu mode is disabled.
Default setting	ON	
Dynamic specification	YES	
Specification within session	YES	Within a Natural session, the MENU parameter can be overridden by the **system command** MAINMENU (described in the Natural System Command Reference documentation).

MFSET - Multi-Fetch Setting

This Natural profile parameter is for UNIX and Windows only.

It specifies whether multi-fetch is used to retrieve records from Adabas databases.

Possible settings	NEVER	Always use single-fetch.
	OFF	Use single-fetch as default. This can be overwritten on statement level.
	ON	Use multi-fetch as default. This can be overwritten on statement level.
Default setting	OFF	
Dynamic specification	YES	
Specification within session	NO	

ML - Position of Message Line

This parameter is available as a profile parameter on all platforms. As as session parameter, it is available for Windows only.

It determines the line to be used for the display of applications which do not set the message line position explicitly by using the SET CONTROL ’M’ statement. For information on the operand ’M’, see also Natural terminal command %M - Control of Message Line.

Within a Natural session, the profile parameter ML can be overridden by the session parameter ML.

Possible settings	B	Natural messages are displayed at the bottom of the screen.	
	T	Natural messages are displayed at the top of the screen.	
Default setting	T		
Dynamic specification	YES		
Specification within session	YES	Applicable Statements:	SET CONTROL M
		Applicable Command:	

For details on session parameter specification at statement or element level and evaluation at compilation time or at runtime, refer to Session Parameter Specification/Evaluation Overview.

MONSIZE - Size of SYSTP Monitor Buffer

This Natural profile parameter is for mainframes only.

It determines the size of the buffer used by the Monitor function of the SYSTP utility (described in the Natural Utilities documentation).

Alternatively, you can use the equivalent Natural profile parameter DS or macro NTDS (see Using Optional Macros in a Natural Parameter Module in the Natural Operations for Mainframes documentation) to specify the buffer size.

Possible settings	1-256	Buffer size in KB.
	0	If MONSIZE=0 or if the requested space is not available, the Monitor function of the SYSTP utility cannot be used.
Default setting	0	
Dynamic specification	YES	
Specification within session	NO	

MP - Maximum Number of Pages of a Report

This Natural profile and session parameter is for mainframes only.

It determines the maximum number of pages to be produced for a report.

Within a Natural session, the setting of profile parameter MP can be reduced, but not increased by the FORMAT statement. The value specified with the session parameter MP applies only to the specified report.

Possible settings	1 - 99999	The value specified is the number of physical pages and has no effect on the starting page number used. The program will be terminated with an error message if the MP value is exceeded.	
	0	No page limit is defined.	
Default setting	0		
Dynamic specification	YES		
Specification within session	NO	Applicable Statements:	DISPLAY FORMAT PRINT WRITE
		Applicable Command:	None

For details on session parameter specification at statement or element level and evaluation at compilation time or at runtime, refer to Session Parameter Specification/Evaluation Overview.

MSGSF - Display System Error Messages in Short/Full Format

This Natural profile parameter is for:

- Mainframes
- UNIX and Windows

It can be used to avoid truncation of Natural system error messages.

Possible settings	ON	System error messages will be displayed in full; that is, program name, line number and actual message text.
	OFF	System error messages will be displayed in short form; that is, only the actual message text will be displayed (but not the program name and line number).
Default setting	ON	
Dynamic specification	YES	
Specification within session	YES	Within a Natural session, the profile parameter MSGSF can be overridden by the Natural terminal command %MSGSF.

By default, a Natural system error message consists of the following:

- the name of the program,
- the number of the line that caused the error,
- the actual text of the message.

Depending on the size of the window in which the message is displayed, the text may be truncated. With this parameter, you can avoid such truncation.

MT - Maximum CPU Time

This Natural profile and session parameter is for mainframes and only applies to programs executed in batch mode or under TSO.

It determines the maximum amount of CPU time which can be used by a Natural program.

Within a Natural session, the profile parameter MT can be overridden by the session parameter MT.

Possible settings	1 - 9999999	Maximum amount of CPU time in seconds. If Natural Security is installed, the profile parameter MT can be overridden within Natural Security. With Natural Security, the maximum value for the profile parameter MT is 32767. To use a higher value as specified with the MT profile or session parameter, specify MT=0 within Natural Security.	
	0	MT=0 defines that no Natural CPU time limit is to be in effect.	
Default setting	60		
Dynamic specification	YES		
Specification within session	YES	Applicable Statements:	SET GLOBALS
		Applicable Command:	GLOBALS

For details on session parameter specification at statement or element level and evaluation at compilation time or at runtime, refer to Session Parameter Specification/Evaluation Overview.

Notes:

- The limit for programs operating in interactive mode is controlled by the TP monitor in use.
- The maximum value that can be used is determined by the operating system environment. Any setting in excess of the maximum is reduced to the maximum supported by the operating system.
- In system environments which do not support CPU time measurement, the limit is interpreted as elapsed time. The CPU time limit is ignored for systems without timer support.

NAFSIZE - Size of Buffer for Natural Advanced Facilities

This Natural profile parameter is for mainframes and only applies if Natural Advanced Facilities is installed.

It specifies the size of the work buffer used by Natural Advanced Facilities.

Alternatively, you can use the equivalent Natural profile parameter DS or macro NTDS (see Using Optional Macros in a Natural Parameter Module in the Natural Operations for Mainframes documentation) to specify NAFSIZE.

Possible settings	1 - 64	Buffer size in KB.
	0	NAFSIZE=0 disables Natural Advanced Facilities.
Default setting	0	
Dynamic specification	YES	
Specification within session	NO	

If Natural Advanced Facilities is to be used, a setting has to be specified for this parameter; see NATSPOOL Initialization (in the Natural Advanced Facilities documentation).

If the requested space is not available, Natural Advanced Facilities cannot be used.

NAFUPF - Natural Advanced Facilities User Profile

This Natural profile parameter is for mainframes and only applies if Natural Advanced Facilities is installed.

It is used to specify the user-profile name for Natural Advanced Facilities.

Possible settings	1 to 8 characters	Name of the user profile.
Default setting	none	
Dynamic specification	YES	
Specification within session	NO	

See NATSPOOL Initialization (in the Natural Advanced Facilities documentation).

NATLOG - Natural Log File

This Natural profile parameter is for UNIX and Windows only.

It is used to log messages that will not (or could not) be written to the standard output in interactive mode or to the output file CMPRINT in batch mode.

Possible settings	OFF	Disables the log mechanism.
	ERR	Logs error messages.
	INF	Logs information and success messages.
	WRN	Logs warning messages.
	ALL	Logs all types of messages.
Default setting	OFF	
Dynamic specification	YES	
Specification within session	NO	

The location of the NATLOG file is the TEMP directory of Natural (specified in the local configuration file NATURAL.INI). If this path is unknown, Natural creates the file in the current directory.

Natural tries to create the following file name, if the user-ID/et-ID information could be retrieved:

`NATURAL_<user-ID>_<et-ID>.LOG`

If user-ID and et-ID could not be retrieved, then the following file name is used: NATURAL.LOG

Example File Names:

A Natural batch process is running with USER-ID=SYSTEM and ETID=14, then the resulting file name is:

```
NATURAL_SYSTEM_14.LOG
```

If the USER-ID could not be retrieved (which is the case if an error occurs during the initialization phase of Natural), then the resulting file name is:
NATURAL.LOG

Platform-specific Examples

The following examples show the contents of a Natural log file. At top of the file, there is a header with some environment information, for example the Natural Version, the parameter file currently in use and so on. Two entries follow. The first one an entry which displays the I/O channels needed for real batch mode. The second entry shows an error message. Both messages are counted in the statistics summary.

- Example NATLOG Output Contents for UNIX
- Example NATLOG output contents for Windows

Example NATLOG Output Contents for UNIX:

```
# ######################################################
#          N a t u r a l   L o g   F i l e
# ######################################################
#
# Logging started at : 06-Oct-2003 08:08:38.023
#                    :
# Natural Version    : V 6.1.1 DEVELOP SAG 2003
# Server Type        : (none)
# Device             : BATCH  (real)
# Parameter File     : NATPARM
#                    :
# User ID            : NATURAL
# ET ID              : TEST
# Network User ID    : NATURAL
#                    :
# Host Name          : hpn2
# Machine Class      : UNIX
# Operating System   : HP_HPUX  B.10.20A
#                    :
# Process ID         : 17921
#                    :
# NATLOG Option      : ALL
#
# ######################################################
#
# ------------------------------------------------------------------------------
# 08:08:38.025  NATURAL     INFORMATIONAL  STATISTICS:  INF=1  WRN=0  ERR=0
# ------------------------------------------------------------------------------
setting of parameter CMSYNIN (command file)
$HOME/tmp/batch.cmd
setting of parameter CMOBJIN (input file)

setting of parameter CMPRINT (output file)
$HOME/tmp/batch.out
# ------------------------------------------------------------------------------
# 08:08:38.028  NATURAL     ERROR          STATISTICS:  INF=1  WRN=0  ERR=1
# ------------------------------------------------------------------------------
 NATURAL Startup Error:  42
 Batch mode driver error.
 Parameter CMOBJIN not set.
```


Example NATLOG Output Contents for Windows

```
# ######################################################
#          N a t u r a l   L o g   F i l e
# ######################################################
#
# Logging started at : 06-Oct-2003 08:10:12.044
#                    :
# Natural Version    : V 6.1.1 SAG 2003
# Server Type        : (none)
# Device             : BATCH  (real)
# Parameter File     : NATPARM
#                    :
# User ID            : NATURAL
# ET ID              : TEST
# Network User ID    : MYDOMAIN\NATURAL
#                    :
# Host Name          : PCNAT
# Machine Class      : PC
# Operating System   : WNT-X86 4.0 (1381)
#                    :
# Process ID         : 274
#                    :
# NATLOG Option      : ALL
#
# ######################################################
#
# ------------------------------------------------------------------------------
# 08:10:13.003  NATURAL     INFORMATIONAL  STATISTICS:  INF=1  WRN=0  ERR=0
# ------------------------------------------------------------------------------
setting of parameter CMSYNIN (command file)
D:\TEMP\syn37437.tmp
setting of parameter CMOBJIN (input file)

setting of parameter CMPRINT (output file)
D:\TEMP\out37437.tmp
# ------------------------------------------------------------------------------
# 08:10:15.020  NATURAL     ERROR          STATISTICS:  INF=1  WRN=0  ERR=1
# ------------------------------------------------------------------------------
 NATURAL Startup Error:  42
 Batch mode driver error.
 Parameter CMOBJIN not set.
```


NATVERS - Switching between Natural Environments

This Natural profile parameter is for UNIX and Windows platforms only.

It enables you to access various Natural environments, that is, various err, txt, bin, etc, samples and tmp subdirectories. The NATVERS value you enter is used by Natural to determine the SAG.INI section you want to use with your Natural session. By default, the Natural installation creates a section in the SAG.INI file that points to the installed Natural environment.

The Natural installation automatically identifies the Natural version and creates the section [NATURAL=6.1.1] in your SAG.INI file that points to the installed Natural environment. This might look as in the following example:

```
[NATURAL-6.1.1]
PROD_API=C:\Program Files\Software AG\6.1.1\Bin\natprd32.dll
NATGUI_BMP=C:\Program Files\Software AG\Natapps\Pic
NATDIR=C:\Program Files\Software AG
NATINI=C:\Program Files\Software AG\6.1.1\Etc
[NATURAL-6.1.1-END]
```

If you want to access envrionments other than the one that was installed automatically, create a new section for each environment as shown in the following example:

```
[NATURAL-Server6.1.1]
NATVERS=6.1.1
PROD_API=\\Server\Program Files\Software AG\6.1.1\Bin\natprd32.dll
NATGUI_BMP=\\Server\Program Files\Software AG\Natapps\Pic
NATDIR=\\Server\Program Files\Software AG
NATINI=\\Server\Program Files\Software AG\6.1.1\Etc
[NATURAL-Server6.1.1-END]
```

Possible settings	any character string	Determines the Natural version.
Default setting	6.1.1	**Note:** A DEFAULT-VERSION entry in SAG.INI is no longer necessary.
Dynamic specification	YES	This parameter can only be specified dynamically.
Specification within session	NO	

Replaced settings are still contained in the SAG.INI file and can be used by specifying them with NATVERS to switch to the corresponding Natural environments.

NC - Use of Natural System Commands

This Natural profile and session parameter is for all platforms.

It controls whether Natural system commands can be used during the Natural session or not.

Within a Natural session, the profile parameter NC can be overridden by the session parameter NC.

Possible settings	ON	System commands cannot be used - except FIN, LAST, LOGOFF, LOGON, MAINMENU, RENUMBER, RETURN, SETUP and TECH. If you have Natural Security installed, any system command restrictions you set with Natural Security are valid, regardless of the setting of the NC profile parameter. In a Natural Development Server environment on mainframe computers, the value OFF will be assumed for the Natural Development Server, even if NC=ON has been specified. If NC=ON has been specified on the client side, subsequent system commands issued on the client side will be rejected as described above.	
	OFF	All system commands can be used.	
Default setting	OFF		
Dynamic specification	YES		
Specification within session	YES	Applicable Statements:	SET GLOBALS
		Applicable Command:	

For details on session parameter specification at statement or element level and evaluation at compilation time or at runtime, refer to Session Parameter Specification/Evaluation Overview.

Note:
Natural terminal commands and user-created commands (object module names) are not affected by the NC parameter.

NCFVERS - NCF File Protocol Version

This Natural profile parameter is for UNIX and Windows only.

This parameter enables downward compatibility with Natural Versions lower than Version 6.1. It specifies the protocol version of the Entire Connection format file (.NCF) to be used. This Entire Connection format is generated when work files of type ENTIRE CONNECTION or DEFAULT work files with the file extension ".NCD" are written.

Possible settings	0	A format file of Entire Connection Protocol Version 0 is written. The format files created are compatible with those of Natural Versions lower than Version 6.1.
	2	A format file of Entire Connection Protocol Version 2 is written, which is created by Natural Version 6.1.
Default setting	2	
Dynamic specification	NO	
Specification within session	NO	

For information on the work file types ENTIRE CONNECTION and the Entire Connection format, refer to ENTIRECONNECTION in the DEFINE WORKFILE statement documentation or Work-File Types in the Natural for UNIX Operations documentation.

NENTRY - Left/Right Alignment of Numeric Field Entries

This Natural profile parameter is for UNIX and Windows only.

It determines where the entry in a numeric field is written: to the left or to the right.

Possible settings	LEFT	An entry in a numeric field is written to the left.
	RIGHT	An entry in a numeric field is written to the right.
Default setting	LEFT	
Dynamic specification	YES	
Specification within session	YES	

NISN (Internal Use)

This Natural profile parameter is for mainframes only.

This parameter is reserved for internal use by Natural. Do not change its setting.

NOAPPLERR - Suppress Message Number Prefix NAT

This Natural profile parameter is for UNIX and Windows platforms only.

It is used to suppress the message number prefix "NAT" with user-supplied error messages.

Possible settings	none	If you enter NOAPPLERR in the command line, the prefix NAT is not displayed in error messages.
Default setting	none	
Dynamic specification	YES	This parameter can only be specified dynamically.
Specification within session	NO	

NOPROX - Specify Local Domains for Direct Addressing

This parameter is for UNIX only.

This Natural profile parameter specifies the domain(s) which shall be addressed directly, that is, not via the proxy.

Possible settings	See example below.
Default setting	None
Dynamic specification	NO
Specification within session	NO

Example:

```
"*.software-ag.de; sagus.software-ag.com"
```


NTASKS - Number of Server Tasks to be Started

This parameter is for mainframes only.

For static specification, this parameter is available as a keyword subparameter of the NTRPC macro. For dynamic specification, this parameter is available as a subparameter of the profile parameter RPC.

NTASKS=*nn* specifies the number *nn* of server tasks to be started.

If the server has to handle a large number of client requests, you can use this subparameter to improve the throughput by starting multiple (identically named) copies of the same server process.

Possible values:	1 - 99	Number of copies of one server process.
Default value:	1	
Dynamic specification	YES	
Specification within session	NO	

For additional information on Natural RPC, see the Natural Remote Procedure Call documentation.

NUCNAME - Name of Shared Nucleus

This Natural profile parameter is for mainframes only.

With this parameter, you specify the name of the (environment-independent) Natural shared nucleus if it is to be loaded dynamically and not linked to the environment-dependent Natural nucleus.

See also Natural Shared Nucleus under OS/390 and VSE/ESA in the Natural Operations for Mainframes documentation.

Possible settings	1 to 8 characters	Valid load module name.
Default setting	none	
Dynamic specification	YES	By specifying this parameter dynamically, you are able to use different Natural shared nuclei (for example, for production and for testing) together with the same environment-dependent Natural nucleus without having to relink the nucleus.
Specification within session	NO	

The NUCNAME parameter is ignored if it is specified in a parameter string activated by a SYS or PROFILE profile parameter or in an alternative parameter module (as specified with the PARM profile parameter).

OBJIN - Use of CMOBJIN as Natural Input File

This Natural profile parameter is for mainframes and applies to batch mode only.

It indicates whether the CMOBJIN file (see Natural in Batch Mode in the Natural Operations for Mainframes documentation) is to be used for input data provided with the INPUT statement in batch mode.

Possible settings	Y	Data for a Natural INPUT statement are read from the CMOBJIN file.
	N	The CMOBJIN file is not used and any data for an INPUT statement are read from the CMSYNIN file.
	R	Natural determines which option has been selected for a particular session by the presence or absence of the CMOBJIN DD/FILE statement in the Natural execution JCL/JCS.
Default setting	R	
Dynamic specification	YES	
Specification within session	YES	

OPF - Overwriting of Protected Fields by Helproutines

OPF - Overwriting of Protected Fields by Helproutines

This Natural profile and session parameter is for all platforms.

With this parameter, you specify whether the content of a write-protected field (attribute AD=P) can be overwritten by a helproutine assigned to the field.

Within a Natural session, the profile parameter OPF can be overridden by the session parameter OPF.

Possible settings	ON	A helproutine assigned to a field can overwrite the field's content, even if the field is write-protected.	
	OFF	Helproutines cannot overwrite the contents of write-protected fields.	
Default setting	ON		
Dynamic specification	YES		
Specification within session	YES	Applicable Statements:	SET GLOBALS
		Applicable Command:	GLOBALS

For details on session parameter specification at statement or element level and evaluation at compilation time or at runtime, refer to Session Parameter Specification/Evaluation Overview.

Notes:

- The OPF profile parameter only applies to the field for which a helproutine is invoked; it does not affect parameters explicitly passed to the helproutine. This means that the OPF profile parameter takes no effect if the field for which help is invoked is also explicitly specified as a parameter to be passed to the helproutine.
- In addition, in reporting mode you can change the OPF setting using the statement SET GLOBALS.

OPRB - Database Open/Close Processing

This Natural profile parameter only applies to Adabas and VSAM databases.

It controls the use of database Open/Close commands during a Natural session.

The NTOPRB macro can be used as an alternative to the profile parameter OPRB in the NTPRM macro. The maximum length of an OPRB parameter specification is 256 bytes. If you require a longer specification, use the NTOPRB macro instead of the OPRB parameter.

If you wish to make OPRB specifications that are to apply to all databases, it is strongly recommended that you use the OPRB parameter in the NTPRM macro (and not an NTOPRB macro).

Possible settings	`OPRB=(`*`string`*`)`	With this syntax, you specify an Open request for *all* databases.
	`OPRB=(DBID=`*`nn1`*`,`*`string`*`,DBID=`*`nn2`*`,`*`string`*`, ...)`	With this syntax, you specify an Open request for specific individual databases. As defined in the macro NTDB, the specified DBID identifies the type of database.
	`OPRB=(`*`string`*`,DBID=`*`nn1`*`,`*`string`*`,DBID=`*`nn2`*`,`*`string`*`, ...)`	With this syntax, you specify an Open request for specific individual databases and also a default Open request - the initial *string* - which applies to all databases for which you do not specify an individual *string*.
Default setting	None	
Dynamic specification	NO	
Specification within session	NO	

Generally, the OPRB parameter uses one of the above syntaxes (the possible contents of the *strings* depend on the database system).

Instead of using the OPRB parameter, you can also use the macro NTOPRB in the Natural parameter module NATPARM.

The following topics are covered below:

- Dynamic OPRB with Natural Security
- OPRB for VSAM
- OPRB for Adabas
- NTOPRB Macro Syntax
- Examples of NTOPRB Macros

Dynamic OPRB with Natural Security

A dynamically specified OPRB parameter applies for all logons to libraries in whose security profiles no OPRB parameter is specified. For a logon to a library in whose security profile the OPRB parameter is specified, any dynamically specified OPRB parameter is ignored and the one from the security profile applies.

OPRB for VSAM

The *strings* which can be specified for VSAM databases are described under the OPRB Parameter for VSAM Databases in the Natural for VSAM documentation.

OPRB for Adabas

For Adabas databases, the OPRB parameter is required if either of the following conditions is true for the Natural session:

- An explicit list of Adabas files to be accessed/updated is to be provided. This is necessary, for example, if Adabas cluster updating or exclusive file control is to be requested.
- A single logical transaction is to span two or more Natural programs and, therefore, it is not desired to have Natural issue an END TRANSACTION and CLOSE command at the termination of any given Natural program.

If the OPRB parameter is omitted in the NATPARM module or OPRB=OFF is specified as a dynamic parameter, a Natural session commences with an Adabas Open command requesting UPD (access/update) to the Natural system file. Natural also issues RELEASE CID (Adabas RC) commands to release all ISN lists (ISN lists specified in a RETAIN clause of a Natural FIND statement are not released).

The Adabas record buffer to be used with the initial Adabas OP command can be explicitly provided. The format is similar to that used in an Adabas record buffer for the OP command with the exception that no blanks can be embedded, and the complete setting must be enclosed in parentheses (not apostrophes).

Example 1:

```
OPRB=(ACC=2,4,6,UPD=8.)
```

This specifies that Adabas Files 2,4 and 6 are to be made available for access only and that Adabas File 8 is to be made available for update (which also implies access).

Example 2:

```
OPRB=(EXU=1,2,3.)
```

This specifies that Adabas Files 1,2 and 3 are to be placed under exclusive control for this Natural session.

Combinations of the keywords ACC, UPD and EXU must follow the rules as defined in the relevant Adabas documentation. When these keywords are coded, Natural issues an OP command at the start of a Natural session and a CL at the end of the Natural session. At the end of a Natural program, only the required RC commands are issued to release held ISN lists.

In all of the above situations, the OP command, which is always issued at the start of a Natural session, contains in the Additions 1 field of the Adabas control block the User ID for the Natural session. In batch mode, this is the job name. In TP mode, this is the setting supplied at system initialization by the Natural interface module. In both cases, the setting used is available in the Natural system variable *INIT-USER.

NTOPRB Macro Syntax

The syntax of the NTOPRB macro is as follows:

```
NTOPRB dbid,'ACC=fnr,UPD=fnr'
```

For possible values, see the OPRB parameter; if you use Natural with VSAM, see also the Natural for VSAM documentation.

Examples of NTOPRB Macros

```
NTOPRB 12,'ACC=40,UPD=20'
```

```
NTOPRB 15,'EXU=1,','2,3'
```

If the OPRB string for one DBID is very long, it can be divided in two or more strings separated by commas (see the second example), as the Assembler allows single strings up to 256 bytes only.

OPT - Control of Natural Optimizer Compiler

This Natural profile parameter is for mainframes and only applies if the Natural Optimizer Compiler is to be used.

This parameter is used to activate/deactivate the Natural Optimizer Compiler and controls the various options related to it. It corresponds to the macro NTOPT in the Natural parameter module NATPARM.

Possible settings		See the Dynamic Profile Parameter OPT in the Natural Optimizer Compiler documentation.
Default setting	OFF	
Dynamic specification	YES	This parameter can only be specified dynamically. In the Natural parameter module NATPARM, the macro NTOPT must be used instead.
Specification within session	YES	

The following topics are covered below:

- OPT Parameter Syntax
- NTOPT Macro Syntax

OPT Parameter Syntax

The parameter syntax of OPT is, for example, as follows:

```
OPT=(INDX,OVFLW,ZD=OFF)
```

For more syntax examples, refer to Dynamic Profile Parameter OPT in the Natural Optimizer Compiler documentation.

NTOPT Macro Syntax

The syntax of the NTOPT macro is, for example, as follows:

```
NTOPT 'INDX,OVFLW,ZD=OFF'
```

For more syntax examples, refer to Macro NTOPT in the Natural Optimizer Compiler documentation.

OUTDEST - Output Destination for Asynchronous Processing

This Natural profile parameter is for mainframes and only applies to Natural under CICS, Com-plete and UTM.

It specifies the destination to which any Natural error message produced by an asynchronous application is to be sent.

Possible settings	1 to 8 characters	Destination to which a Natural error message is sent.
Default setting	Setting of profile parameter SENDER	
Dynamic specification	YES	
Specification within session	NO	

After an error message has been sent, Natural terminates the asynchronous session.

Under UTM, this parameter is used to specify the ID of the terminal where output from an asynchronous application is to be displayed.

When and how error messages/output from an asynchronous application are output depends on the respective TP monitor.

OVSIZE - Storage Thread Overflow Size

This Natural profile parameter is for mainframes only.

It defines the maximum total amount of variable storage that may be allocated by one Natural session outside its storage thread.

Possible settings	0-2097151	Maximum total storage outside the thread in KB.
Default setting	2097151	That is, the storage outside the thread is limited by the region size only.
Dynamic specification	YES	
Specification within session	NO	

If the storage within the thread is exhausted during a Natural session, additional storage can be allocated outside of the thread. OVSIZE can be used to limit the total amount of variable storage. This does not affect physical storage (see profile parameter WPSIZE), which is allocated outside the thread always.

For non-thread environments (e.g. in batch mode or under TSO), this parameter is not honored.

PARM - Alternative Parameter Module

This Natural profile parameter is for:

- Mainframes
- UNIX and Windows

This parameter specifies an object module containing profile parameter definitions.

Possible settings	1-8 characters	Module name.
Default setting	none	
Dynamic specification	YES	
Specification within session	NO	

These definitions are coded using the various macros as described under Creating a New Natural Parameter Module (in the Natural Operations documentation for Mainframes). The macros are then assembled, resulting in an object module whose name is specified by the user.

When the PARM parameter is specified (either in the linked parameter module or as a dynamic parameter at Natural session start), the appropriate object module is loaded and the profile parameter definitions contained therein take effect. The parameter module is loaded dynamically from the steplib.

Under CICS, a PPT entry is required for this parameter module.

Under BS2000/OSD, OS/390 batch mode and TSO, the current steplib can be defined by profile parameter LIBNAM.

Any profile parameter definitions in effect before the PARM parameter is processed (for example, definitions contained in the linked parameter module or prior dynamic parameters), except the profile parameters ISIZE and NUCNAME, are overridden when the specified parameter module is loaded. Therefore, any dynamic parameters should be specified after the PARM specification.

The profile parameters ISIZE and NUCNAME are ignored if specified in an alternative parameter module.

To restrict the use of an alternative parameter module, you can use the macro NTUSER (described in the USER profile parameter description).

PC - Control of Personal-Computer Access Method

This Natural profile parameter is for:

- Mainframes
- UNIX

It only applies if Natural Connection is installed.

This parameter determines whether support of the personal-computer access method is to be provided using Natural Connection.

Possible settings	ON	Personal-computer support is enabled. The Natural statements READ PC FILE or WRITE PC FILE can be used (for uploading or downloading); see UPLOAD PC FILE and DOWNLOAD PC FILE. With PC=ON, the system variable *DEVICE will always contain the value "PC".
	OFF	No personal-computer support is to be provided.
	NAM	Field names are sent when data are uploaded/downloaded. This value is for mainframe environments only.
	NONAM	No field names are sent when data are uploaded/downloaded. This value is for mainframe environments only.
Default setting	(OFF,NAM)	
Dynamic specification	YES	
Specification within session	YES	The terminal commands %+ and %- can be used to control the PC support.

Multiple values are specified in a value list:

Example:

```
PC=(ON,NONAM)
```

See the Natural Connection documentation for further information.

The files used for the PC access method have to be defined with the macros NTPRINT and NTWORK or the profile parameters PRINT, WORK and HCAM.

PCNTRL - Print-Control Characters

This Natural profile parameter is for mainframes only.

It determines the line-advance characters for printing which are inserted in Column 0 of each print line.

Possible settings	Any character string	This parameter can be specified in character or hexadecimal format (see below).
Default setting	Siemens Environments	`X'4041424344454647484 94A4B4C4D4E4F'`
	IBM Environments	The default setting (according to ASA standard settings) is: `' 0-'` which means that a blank causes a line advance of 1 line, "0" of 2 lines and "-" of 3 lines.  In any **IBM** environment, **do not change the default setting** of this parameter.
Dynamic specification	YES	
Specification within session	NO	

PD - Limit of Pages for NATPAGE

This Natural profile and session parameter is for mainframe and UNIX platforms only.

It specifies the maximum number of pages (screens) which can be stored at the same time in the Natural system file (FUSER) with the NATPAGE screen-paging utility.

Within a Natural session, the profile parameter PD can be overridden by the session parameter PD.

Possible settings	0 - 255	Maximum number of pages (screens).	
Default setting	50		
Dynamic specification	YES		
Specification within session	YES	Applicable Statements:	SET GLOBALS
		Applicable Command:	GLOBALS

For details on session parameter specification at statement or element level and evaluation at compilation time or at runtime, refer to Session Parameter Specification/Evaluation Overview.

Notes:

- If the number of stored screens exceeds the setting of PD, wrap-around technique is used for the system file, which means that the oldest page is overwritten.
- For further information on the NATPAGE utility, see the terminal commands %E, %I, %O, %P and %S.

PERSIST - Flag a Parameter Module as Persistent

This Natural profile parameter is for Windows only.

This parameter is intended for easier handling of parameter settings during development. Its use should be restricted to parameter files which are used by a single person or for a specific task. Multiple concurrent accesses to the same persistent parameter file might cause unexpected interferences.

PERSIST can only be set in the Natural configuration utility.

Possible settings	ON	Setting the PERSIST parameter to ON marks the parameter file as persistent. This means that all settings which are changed during a Natural Studio session are saved to the currently used parameter file when the session is ended. The next time Natural Studio is started with that parameter file, it will resume with the same parameter settings. The saved settings include those which were made at session startup (dynamic parameters issued at the OS command prompt when Natural was invoked) and those which were changed during the session, e.g. with the Natural system commands GLOBALS, COMPOPT.
	OFF	The parameter module is not flagged as persistent.
Default setting	OFF	
Dynamic specification	NO	
Specification within session	NO	

PLOG - Logging of Dynamic Parameters

This Natural profile parameter is for mainframes and only applies in batch mode, under TSO and under CICS.

It enables you to print a list of all profile parameters that were specified dynamically at the start of the session. This may be useful to ascertain which dynamic profile parameters were actually used, particularly if profile parameters like PROFILE or SYS are specified, which in turn "contain" other profile parameters (for a PROFILE or SYS parameter, the entire string of profile parameters activated by it is listed).

Possible settings	ON	In batch mode:	At session start, a list of the dynamically specified profile parameters and their settings is written to the output dataset CMPLOG. (If CMPLOG is not available, the list is written to the standard output dataset CMPRINT.)
		In online mode under TSO:	At session start, a list of the dynamically specified profile parameters and their settings is written to the output dataset CMPLOG. (If CMPLOG is not available, the list is sent to the terminal.)
		In online mode under CICS:	At session start, a list of the dynamically specified profile parameters and their settings is sent to the terminal.
	OFF	No list of dynamic profile parameters is written.	
Default setting	OFF		
Dynamic specification	YES	When specified dynamically, the PLOG parameter applies to all subsequent dynamic profile parameters until the next PLOG specification. This allows you to exclude individual parameters from being printed, for example, if their settings contain passwords or other sensitive information that should not be printed. All dynamic parameters which are specified between a PLOG=OFF specification and a subsequent PLOG=ON specification are not printed.	
Specification within session	NO		

PLUGIN - Enable the Natural Plug-In Components

This Natural profile parameter is for mainframes only.

It is used to enable new or additional Natural components without having to link new components to your Natural nucleus or to apply specific fixes.

Possible settings	OFF	Deactivates all plug-in components, see note below.
Default setting	OFF	
Dynamic specification	YES	
Specification within session	NO	

Note:
Because all components of Natural Version 3.1 that could be enabled using the profile parameter PLUGIN are an integral part as of Natural Version 4.1, the only possible setting of this parameter is OFF.

PM - Print Mode

This Natural profile and session parameter is for all platforms, however, some of its settings are platform-dependent.

It indicates how fields are to be printed or displayed.

Within a Natural session, the profile parameter PM can be overridden by the session parameter PM.

Possible settings	C	**On Mainframes:** An alternative character set is to be used. It can be defined by the profile parameters TAB1 and TAB2. **On UNIX and Windows:** Not available.
	P	**On Mainframes:** The primary (standard) character set is to be used. **On UNIX and Windows:** Not available.
	I	Defines that field settings are to be displayed in inverse direction; that is, from right to left (for example, for use in Middle Eastern countries). This setting requires a specific localized version of Natural. **On Mainframes:** See also the module NATPM in the Natural source library for further information.
	R	**On all platforms:** Resets the PM=I setting to normal (left to right) display direction. PM=R can only be specified dynamically.
	CI, CR, PI, PR	**On Mainframes:** Combinations of the individual values are possible. **On UNIX and Windows:** Not available.
Default setting	PR	On Mainframes.
	R	On UNIX and Windows.
Dynamic specification	YES	
Specification within session	YES	

POS22 - Version 2.2 Algorithm for POS System Function

POS22 - Version 2.2 Algorithm for POS System Function

This Natural profile parameter is for mainframes only.

It can be used to assure that fields are marked / referenced correctly.

Possible settings	ON	The old Version 2.2 algorithm will be used and the correct field will be marked/referenced **Note**: Set POS22=ON only if you execute objects using the POS function which were compiled with Natural Optimizer Compiler Version 2.2. For any other objects, the change in the internal POS algorithm does *not* lead to different results!
	OFF	The algorithm introduced with Natural Version 2.3 is used.
Default setting	OFF	
Dynamic specification	YES	
Specification within session	NO	

As of Natural Version 2.3 for Mainframes, the internal algorithm for the computation of the system function POS (internal field identification) is different. As a result, if you execute with Version 2.3 or higher programming objects which use the POS function in conjunction with the MARK option of an INPUT or REINPUT statement or with the system variable *CURS-FIELD **and** which were compiled with Version 2.2 of the Natural Optimizer Compiler, the wrong field may be marked/referenced.

This parameter will be available only for a limited period of time to allow you a smooth transition from Version 2.2. It will be removed again with a subsequent release of Natural.

In a **Sysplex environment**, or if the Natural thread size makes buffer reallocations necessary, POS22=ON cannot always be guaranteed to yield the desired results. In these cases, the objects concerned should be recataloged in the new version of the Natural Optimizer Compiler to ensure correct POS result.

PRGPAR - Data to be Passed to Program Receiving Control at Termination

PRGPAR - Data to be Passed to Program Receiving Control at Termination

This Natural profile parameter is for UNIX and Windows platforms only.

It specifies data to be passed to the program that receives control when Natural terminates (see also profile parameter PROGRAM).

Possible settings	any valid character string	Data to be passed.
Default setting	blank	No data is passed.
Dynamic specification	NO	
Specification within session	NO	

PRINT - Print File Assignments

This Natural profile parameter is for mainframes only.

It allows you to define the print files to be used during the session. Within a session, up to 31 logical print files (numbered 1 to 31) and the hardcopy print file (Number 0) can be used.

PRINT corresponds to the NTPRINT macro in the parameter module NATPARM. To provide different print file definitions, PRINT or NTPRINT can be specified multiple times.

Possible settings	See Keyword Subparameters below.	
Default setting	See below.	
Dynamic specification	YES	The parameter PRINT can only be specified dynamically. In NATPARM, the macro NTPRINT must be used.
Specification within session	NO	

The software components for accessing print files in different environments are called access methods. For the duration of a Natural session, each logical print file can be assigned to one access method only. The access method for a print file is determined by the keyword subparameter AM (see below).

In OS/390 under TSO and in batch mode, print files need not be predefined in the JCL. Provided they are defined by subparameter AM=STD, they can be allocated dynamically during the session by a Natural program using the DEFINE statement or application interface USR2021 (in library SYSEXT).

This document covers the following topics:

- PRINT Parameter Syntax
- NTPRINT Macro Syntax
- Keyword Subparameters for All Environments
- Keyword Subparameters for AM=STD in All Environments
- Keyword Subparameters for AM=STD in OS/390 Environments
- Keyword Subparameters for AM=STD in VSE/ESA Environments
- Keyword for AM=STD in BS2000/OSD Environments
- Keyword Subparameters for AM=CICS
- Keyword Subparameters for AM=COMP (Com-plete)
- Keyword Subparameters for AM=SMARTS (Com-plete)
- Keyword for AM=IMS
- Keyword Subparameters for DEFINE PRINTER Statement

See also Print and Work File Handling with External Datasets in a Server Environment in the Natural Operations for Mainframes documentation.

PRINT Parameter Syntax

With the PRINT parameter, you first specify one or more logical print file numbers, and then several keyword subparameters, which define the characteristics for these print files:

```
PRINT=((print-file-numbers),keyword-subparameters,...)
```


print-file-numbers

The file numbers must be specified first and enclosed in parentheses. The numbers can be from 0 to 31. They can be specified in any sequence. Multiple numbers must be separated from one another by commas or blanks. To specify a range of numbers, you can use a hyphen (-).

keyword-subparameters

The various types of keyword parameters are described below.

For print files with different characteristics, you specify different PRINT parameters. If any previous definition (or default) for the same print file exists, only the values for the specified keyword subparameters are overwritten, all other values remain unchanged.

Examples:

```
PRINT=((2,12,18),AM=STD,DEST='PRINT**',OPEN=INITOBJ,CLOSE=CMD)
PRINT=((1,3,6-11,15),AM=NAF)
PRINT=((0),AM=STD,DEST=HARDCOPX)
```

NTPRINT Macro Syntax

With an NTPRINT macro, you first specify one or more logical print file numbers, and then several keyword subparameters which define the characteristics that are to apply to these print files:

```
NTPRINT (print-file-numbers),keyword-subparameters,...
```

print-file-numbers

The file numbers must be specified first and enclosed in parentheses. The numbers can be from 0 to 31. They can be specified in any sequence. Multiple numbers must be separated from one another by commas. To specify a range of numbers, you can use a hyphen (-).

keyword-subparameters

The various types of keyword subparameters are described below.

For print files with different characteristics, you specify different NTPRINT macros. If any previous definition (or default) for the same print file exists, only the values for the specified keyword subparameters are overwritten, all other values remain unchanged.

Examples:

```
NTPRINT (2,12,18),AM=STD,DEST='PRINT**',OPEN=INITOBJ,CLOSE=CMD
NTPRINT (1,3,6-11,15),AM=NAF

NTPRINT (0),AM=STD,DEST=HARDCOPX
```


Keyword Subparameters for All Environments

The following keyword subparameters are available: AM | DEST | OPEN | CLOSE | ROUTE

AM - Type of Access Method

AM=*xxx* specifies the type of access method to be used.

For an online session, all print files to be used have to be assigned to a specific access method.

For a batch session, any print files not assigned to a specific access method will be automatically detected and assigned by the standard batch access method (AM=STD), provided that they have been predefined in the JCL. See also profile parameter FAMSTD (overwriting of print and work file access method assignments).

Value	Meaning
STD	Standard sequential batch files (batch, TSO, TIAM, VM/CMS OS simulation).
CMS	CMS disk and SFS files.
COMP	Com-plete print files.
CICS	CICS transient data or temporary storage.
NAF	Natural Advanced Facilities.
IMS	IMS/TM destinations.
PC	Entire Connection.
USER	Third-party vendor print interface.
SMARTS	SMARTS print file.
ESS	Entire System Server.
OFF	Unassigned. No automatic assignments if FAMSTD=OFF is set.
0	Unassigned. Automatic assignments if FAMSTD=OFF is set. This is the default value.

Note:
PRINT=OFF is equivalent to: PRINT=((1-31)), AM=OFF).
It does not affect any of the other keyword subparameter specifications.
`PRINT=((0),AM=`*`xxx`*`)` or `NTPRINT (0),AM=`*`xxx`* determines the hardcopy print access method and is equivalent to the profile parameter HCAM=*xxx*.

DEST - External Dataset Name

DEST=*name* specifies the print destination (1 - 8 characters).

This corresponds to the OUTPUT value of the DEFINE PRINTER statement (and can be overwritten by a DEFINE PRINTER OUTPUT specification).

The meaning of this keyword subparameter depends on the access method.

Access Method	Meaning of DEST
AM=STD	DEST is the logical dataset name (DDNAME, LINK name, DTF name). If the destination is to be for multiple files, two asterisks (**) have to be specified for the file number. These will be replaced by the corresponding logical file number for each print file. A DEST value including two asterisks must be enclosed in apostrophes when it is used as a dynamic parameter. The default value is DEST='CMPRT**' for IBM and DEST='P**' for SIEMENS environments. Under VSE/ESA only 7-character names are supported.
AM=CICS	There is no default value for print files under CICS. Here, the DEST subparameter is mandatory, that is, CICS print files defined without a valid DEST specification are ignored. The Natural CICS interface also supports a variable (see TERMVAR parameter in the NCIPARM generation macro; &TID is the default) as part of the DEST value which, when being specified, is replaced by the actual CICS terminal ID. See also Natural Print and Work Files under CICS in the Natural TP Monitors documentation.
AM=CMS	For usage of DEST under CMS, refer to Natural under VM/CMS (in the Natural Operations for Mainframes documentation).
AM=IMS	Specifies the IMS/TM destination.

Note:
`PRINT=((0),DEST=`*`xxx`*`)` or `NTPRINT (0),DEST=`*`xxx`* determines the hardcopy print destination and is equivalent to the Natural profile parameter HCDEST=*xxx*.

OPEN - Time of File Opening

OPEN=*xxx* determines when the file is to be opened:

Value	The file is opened
INIT	for output at session initialization.
OBF	according to the default OPEN value for the different environments (batch, CICS, Com-plete, TSO).
OBJ	when the execution of the first object which accesses the file starts. This is the general default, except for AM=COMP and AM=IMS.
OBJ1	when the execution of the first object on Level 1 that accesses the file starts. Otherwise, it is opened when it is first accessed.
ACC	when it is first accessed by a statement. This is the default for AM=COMP and AM=IMS.
INITOBF	for output at session initialization. Any subsequent re-opening of the file sets the default OPEN value for the different environments (batch, CICS, Com-plete, TSO).
INITOBJ	for output at session initialization. Any subsequent re-opening of the file will be performed when the execution of the first object which accesses the file starts.
INITOBJ1	when the execution of the first object on Level 1 that accesses the file starts. Otherwise, it is opened when it is first accessed.
INITACC	for output at session initialization. Any subsequent re-opening of the file will be performed when it is first accessed by a statement.

CLOSE - Time of File Closure

CLOSE=*xxx* determines when the file is to be closed:

Value	The file is closed
OBJ	either when processing of the object in which it was first accessed is finished or when command mode, NEXT mode or MAINMENU is reached.
CMD	when command mode, NEXT mode or MAINMENU is reached. This is the default for AM=NAF, AM=COMP and AM=IMS.
FIN	at session end (this is the default for AM=STD). With CLOSE=FIN, a DEFINE PRINTER PRINTER statement causes an error if the printer was opened already. A CLOSE PRINTER statement for the printer is ignored.
USER	only if the file is open and one of the following conditions is true: • a CLOSE PRINTER statement is issued, • a DEFINE PRINTER statement is issued, • the session terminates.

ROUTE - Logical Print File Routing

Route=*xxx* determines whether logical print file routing is done according to the OUTPUT clause of the DEFINE PRINTER statement.

ON	Print file routing is done. This is the default value.
OFF	No print file routing is done.

Print file routing means that, if the name defined in the OUTPUT clause of a DEFINE PRINTER statement denotes a print file destination which is defined by a different logical printer, all print output is routed to this print file.

NTPRINT Keyword Subparameters for AM=STD in All Environments

The following keyword subparameters are available: RECFM | BLKSIZE | LRECL | TRUNC | PAD | PADCHRO | ASA

RECFM - Default Record Format of Dataset

RECFM=*xxxx d*etermines the default record format of the dataset.

The following formats are supported:

F	Fixed
V	Variable
U	Undefined
B	Blocked
S	Spanned
A	ASA
M	Machine control characters

The following values and also combinations of values are possible:

Possible value:	F, FA, FM, FB, FBA, FBM, V, VA, VM, VB, VBA, VBM, VBS, VBSA, VBSM, U, UA, UM.
Default value:	RECFM=VBA (variable blocked with ASA).

The RECFM specification only applies if no record format is predefined in the JCL or (OS/390 only) in the dataset DCB.

BLKSIZE - Default Block Size of Dataset

BLKSIZE=*nnnnn* determines the default block size (in bytes) of the dataset.

Possible values:	0, or 8 to 32767.
Default value:	1016.

The BLKSIZE specification only applies if no block size is predefined in the JCL or (OS/390 only) in the dataset DCB.

LRECL - Default Record Length of Dataset

LRECL=*nnn d*etermines the default record length (in bytes) of the dataset.

Possible values:	0, or 5 - 254.
Default value:	0

This subparameter is used particularly to check for truncation and padding.

For RECFM=V (B) the LRECL value includes a 4-byte record descriptor word.

If LRECL = 0 is defined, the following applies:

- With RECFM = V (B), LRECL defaults to the minimum of BLKSIZE-4 and 254.
- With RECFM = U, LRECL defaults to BLKSIZE.
- With RECFM = F (B), the maximum record length in the Natural program being executed is taken when the file is opened. If no record length from a program is available when the file is opened, for example with OPEN=INIT, a record length of 132 is taken (plus 1 for ASA or a machine control character and/or plus 4 for a record-descriptor word if the record format is variable).

The LRECL specification only applies if no record length is predefined in the JCL or (OS/390 only) in the dataset DCB.

TRUNC - Truncation of Output Records

TRUNC=*xxx* determines whether the output records are truncated:

ON	Output records that are longer than the record length (LRECL) of the dataset will be truncated. This is the default value.
OFF	Error NAT1512 will be issued if an output record is longer than the dataset record length.

PAD - Padding of Output Records

PAD=*xxx* determines whether the output records are padded or not (applies only to datasets of fixed record length):

ON	Output records that are shorter than the record length (LRECL) of the dataset will be padded with padding characters defined by keyword subparameter PADCHRO. This is the default value.
OFF	Error NAT1510 will be issued if an output record is shorter than the dataset record length.

PADCHRO - Padding Character of Output Records

This subparameter defines the character which is used for padding if PAD=ON is defined for the print file.

Possible values:	*'x'* *x'xx'*	(one character *x* within single quotes) (one hex character *xx*)
Default value:	' '	(blank or *x*'40')

ASA - Use of ASA Record Format

ASA=*xxx* determines whether the ASA record format is used.

ON	An ASA character is included in the output print records. Under OS/390, this enforces ASA record format, regardless of the RECFM setting in the DCB or the RECFM subparameter. This is the default value.
OFF	No ASA character is included in the output print records. Under VSE/ESA batch access method (AM=STD), a valid ASA character must be supplied in column one of the output record if the output file is a spool file, otherwise error NAT1530 will be issued.

Keyword Subparameters for AM=STD in OS/390 Environments

The following keyword subparameters are available: REREAD | FREE | BUFNO | DISP | VMAX

REREAD - Closing of Tape File Datasets

REREAD=*xxx* sets the REREAD option for the closing of the tape file:

ON	The REREAD option is set for the CLOSE SVC. This causes the volume to be repositioned to reprocess the dataset . This is the default value.
OFF	The REREAD option is not set for the CLOSE SVC.

FREE - Dataset De-allocation at File Closure

FREE=*xxx* determines whether the dataset is de-allocated when the file is closed:

ON	The FREE option is set for the CLOSE SVC, which means that the dataset is de-allocated when it is closed (and not at step termination).
OFF	The FREE option is not set for the CLOSE SVC . This is the default value.

BUFNO - Default Number of OS/390 I/O Buffers of Dataset

BUFNO=*nnn* defines the default number of OS/390 I/O buffers of the dataset.

Possible values	0 - 255.
Default value	0. In this case, OS/390 allocates five I/O buffers per default.

The number of I/O buffers can improve the performance of print file access dramatically. Note that the storage for I/O buffers is allocated below the 16 MB line.

The BUFNO specification applies only if the BUFNO parameter is not specified in the JCL for the dataset.

DISP - Open Print File for Modification

DISP=*xxx* determines whether the print file is opened for modification.

This corresponds to the JCL DD statement subparameter DISP=MOD.

MOD	New records are added at the end of the file.
NOMOD	The print file is rewritten from the start. This is the default value.

VMAX - Control LRECL for Variable Record Format

VMAX=*xxx* controls the LRECL setting for an output file with variable record format (RECFM=V).

ON	Providing a nonzero BLKSIZE value exists for the file, VMAX=ON sets LRECL=BLKSIZE-4 for variable record format, regardless of the LRECL setting in the DCB or the LRECL subparameter.
NAT	LRECL is set to the length +4 of the largest record in the application program if this value is less than LRECL in the DCB for the dataset.
OFF	LRECL from the DCB for the dataset or the LRECL subparameter is used. This is the default value.

PRINT Keyword Subparameters for AM=STD in VSE/ESA Environments

The following keyword subparameters are available: SYSNR | LABEL | REWIND

SYSNR - Logical VSE SYS Number

SYSNR=*nn* determines the logical VSE SYS number.

Possible values:	1 - 99.
Default value:	By default, the SYS number is print file number plus 40 for print files 1 - 31; for print file 0, that is the hardcopy printer, the default is SYSLST. **Example:** The VSE default SYS number for print file 11 is 11 + 40 => SYS051.

LABEL - Tape Label Processing

LABEL=*xxx* determines the tape label processing:

ON	The tape is in standard label format . This is the default value.
OFF	The tape is unlabeled with front tape mark.
NOTM	The tape is unlabeled without front tape mark.

REWIND - Action at File Closure

REWIND=*xxx* determines the action to be taken when a tape file is closed:

ON	The tape is rewound when the file is closed . This is the default value.
OFF	The tape is not rewound when the file is closed.
UNLOAD	The tape is unloaded when the file is closed.

PRINT Keyword Subparameters for AM=STD in BS2000/OSD Environments

The following keyword subparameter is available: DISP

DISP - File Open Mode

DISP=*xxx* determines the open mode of the file:

EXT	The open mode is set to EXTEND.
NOEXT	The open mode is set to the default value OUTPUT. This is the default value.

PRINT Keyword Subparameters for AM=CICS

The following keyword subparameters are available: TYPE | DISP

TYPE - Type of CICS Storage Medium

TYPE=*xxxx* specifies the type of CICS storage medium to be used:

MAIN	Temporary main storage.
AUX	Temporary auxiliary storage.
TD	Transient data.

The default value used depends on the DEST parameter setting. If the DEST subparameter value matches a valid CICS transient data queue, the TYPE subparameter defaults to TD, otherwise MAIN will be taken as the default value.

DISP - CICS Temporary Storage Queue Disposition

DISP=(*xxx*,*xxx*) specifies the CICS temporary storage queue disposition.

Possible value pairs are:

(NEW,KEEP)	The storage queue is deleted when the file is opened. This is the default value.
(NEW,DELETE)	The storage queue is deleted when the file is opened and when it is closed.
(OLD,DELETE)	The storage queue is deleted when the file is closed.
(OLD,KEEP)	The storage queue is not deleted.

Note:
The DISP specification does not apply to CICS extra-partition transient data queues.

PRINT Keyword Subparameters for AM=COMP (Com-plete)

The following keyword subparameter is available: DRIVER

DRIVER - Name of Com-plete Print Driver

DRIVER=*name* specifies the *name* of the Com-plete print driver to be used.

PRINT Keyword Subparameters for AM=SMARTS (Com-plete)

The following keyword subparameter is available: DEST

DEST - Logical Printer

DEST=*print-server-queue*	The environment variable SAG_APS_LPD_*xyz* defines a logical printer under complete, where *xyz* is the name of the print server queue. If the environment variable SAG_APS_LPD_*xyz* exists for the specified DEST, the output is directly routed to that line printer. For more information, see the *Complete Initialization and Startup Manual*, section *Defining Terminals and Printers*.
DEST=*printer-file-name*	If no print server queue for that printer is available, DEST specifies a printer file name. It specifies the location of the output file in the file system. The name of the output file is generated from the userId and a sequence number. Since the DEST clause is restricted to an 8 character maximum, it is useless to define a file with absolute PFS path specification. The name specified in the DEST clause is relative to the print file root directory. The print file root directory is specified with the environment variable NAT_PRINT_ROOT.

Example:

```
NAT_PRINT_ROOT=/nat/printer
DEST=printer1
UserId=xyz
```

The first output will be written to file /nat/printer/printer1/xyz1.

To specify a file with absolute path definition, the OUTPUT clause of the DEFINE PRINTER statement must be used.

PRINT Keyword Subparameters for AM=IMS

The following keyword subparameters are available: BLKSIZE | DRIVER

BLKSIZE - Size of the Print Buffer

BLKSIZE=nnnnn specifies the size of the print buffer sent to the IMS/TM destination.

DRIVER - Name of Natural IMS Print Driver

DRIVER=*name* specifies the *name* of the Natural IMS print driver to be used.

For possible values, see NIMPARM Macro Parameters and Support of the Natural WRITE (*n*) Statement in the section Natural under IMS/TM in the Natural TP Monitor Interfaces documentation.

PRINT Keyword Subparameters for DEFINE PRINTER Statement

With the following keyword subparameters, you can set default values for the DEFINE PRINTER statement options of the same names (see the Natural Statements documentation). When a printer is closed, all DEFINE PRINTER statement options are reset to their default values.

The following keyword subparameters are available: PROFILE | NAME | FORMS | DISP | COPIES | CLASS | PRTY

PROFILE - Name of Printer Control Characters Table

PROFILE=*name* specifies the *name* of printer control characters table (NTCCTAB macro).

NAME - Name of Listing

NAME=*name* specifies the listing *name*.

FORMS - Name of Listing Forms

FORMS=*name* specifies the listing forms *name*.

DISP - Listing Disposition

DISP=*disposition* specifies the listing *disposition* (HOLD, KEEP, DELETE or LEAVE).

COPIES - Number of Copies

COPIES=*nnn* specifies the number of copies to be printed (1 - 255).

CLASS - Spool Class

CLASS=*class* specifies the spool class (1 byte).

PRTY - Listing Priority

PRTY=*nnn* specifies the listing priority (1 - 255).

PROFILE - Activate Dynamic Parameter Profile

This Natural profile parameter is for mainframes only.

When you invoke Natural with dynamic profile parameters, instead of having to specify a whole string of individual parameters each time you invoke Natural, you can specify the string of parameters once, store this string under a profile name and then invoke Natural with only one dynamic parameter as follows:

```
PROFILE=profile-name
```

The parameters defined under this profile are passed to Natural as dynamic profile parameters.You create and maintain these profiles with the utility SYSPARM (described in the Natural Utilities documentation).

Possible settings	1-8 characters	Profile-name. Or special options, see below.
Default setting		
Dynamic specification	YES	
Specification within session	NO	

Special Options

In addition, the PROFILE parameter provides the following special options:

PROFILE=AUTO	Natural takes the current TP user ID (as contained in the system variable *INIT-USER) as profile name, which means that the profile defined under the name corresponding to that ID is used. If no such profile is found, a profile named "AUTO" is used instead (if available). You can define such an "AUTO" profile as default profile for users without individual profiles.
PROFILE=TERMINAL	Natural takes the current terminal ID (as contained in the system variable *INIT-ID) as profile name, which means that the profile defined under the name corresponding to that ID is used.
PROFILE=PROGRAM	Natural takes the name of the program currently executing as Natural (as contained in the system variable *INIT-PROGRAM) as profile name, which means that the profile defined under this name is used.

By default, the profile is read from the current FNAT system file.

To read it from a different system file, you can specify the desired database ID, file number, password and cipher code after the *profile-name* (or after one of the above special options) as follows:

```
PROFILE=(profile-name,dbid,fnr,password,cipher-code)
```


To ensure that all profile parameters are read from the same system file (other than FNAT), specify the following in the parameter module:

```
PROFILE=(,dbid,fnr)
```

If the PROFILE parameter is specified within a parameter module, it is evaluated **after** the other parameters in the parameter module, but **before** any dynamically specified profile parameters are evaluated; this means that parameters specified within the profile can be overridden by individually specified dynamic parameters.

To restrict the use of a profile, you can use the USER profile parameter.

Unlike other parameters, a PROFILE parameter specification cannot be overwritten by another PROFILE. So you can have multiple parameter profiles which are evaluated all in a sequence.

The PROFILE parameter cannot be used with ADARUN MODE=SINGLE.

PROGRAM - Non-Natural Program Receiving Control after Termination

PROGRAM - Non-Natural Program Receiving Control after Termination

This Natural profile parameter is for:

- Mainframes (except VM/CMS)
- UNIX and Windows

This parameter specifies a non-Natural back-end program which is to receive control after the termination of the Natural session.

Possible settings	1-8 characters	Non-Natural back-end program.
	numeric value	Setting a numeric value, for example PROGRAM=0, indicates "no back-end processing". This is particularly relevant when Natural is invoked by a front-end program, because a default may be taken if PROGRAM is blank or not specified (see Front-End Invoked via XCTL in the Natural TP Monitors documentation).
Default setting	none	
Dynamic specification	YES	It can also be set dynamically from within a Natural program by calling the Natural subprogram CMPGMSET which is provided in the library SYSEXTP.
Specification within session	YES	

Data for the program specified with the PROGRAM parameter can be supplied with the TERMINATE statement (described in the Natural Statements documentation).

For the conventions of calling non-Natural back-end programs, see Back-End Program Calling Conventions in the Natural Operations for Mainframes documentation.

CICS-Specific Information:

In addition to back-end programs, the Natural CICS interface also supports back-end transactions which may be specified via ’RET=*XXXX*’ or ’STR=*XXXX*’ instead of a program name, with *XXXX* being a valid CICS transaction ID.

- RET=*XXXX* indicates that control has to be passed to CICS together with a return transaction ID by a CICS RETURN TRANSID (’*XXXX*’) command.
- STR=*XXXX* indicates that a new transaction has to be started by a CICS START TRANSID (’*XXXX*’) TERMID (*INIT-ID), before relinquishing control via a CICS RETURN command.

MF

PROX - Specify URL of Proxy Server

This Natural profile parameter is for UNIX only.

It specifies the Uniform Resource Locator (URL) of the (Intranet) proxy server through which all requests have to be routed (optional).

Possible settings	URL of proxy server
Default setting	None
Dynamic specification	NO
Specification within session	NO

PROXPORT - Specify Port Number of Proxy

This Natural profile parameter is for UNIX only.

It specifies the port number of the proxy, if any is set.

Possible settings	Port number of proxy, 4 characters at maximum.
Default setting	80
Dynamic specification	NO
Specification within session	NO

PS - Page Size for Natural Reports

This Natural profile and session parameter is for all platforms.

It specifies the maximum number of lines per page to be used for Natural reports created with the DISPLAY, PRINT (under Windows and UNIX only) or WRITE statement. When used as a profile parameter, the PS parameter is honored in batch mode only and defines the physical page size. In online mode, the physical page size is always set to the physical screen height.

Possible settings	1 - 250	Maximum number of lines per page.	
	0	The physical page size is to be used. If PS=0 is specified for the first report to be output (Report 0), the physical-device page-size minus 1 will be used. If PS=0 is specified for Reports 1 - 31, this will cause automatic new-page processing to be inhibited, that is, no automatic page-break processing will be performed.	
Default setting	0		
Dynamic specification	YES		
Specification within session	YES	Applicable Statements:	DISPLAY FORMAT INPUT PRINT (under Windows and UNIX only) SET GLOBALS WRITE
		Applicable Command:	GLOBALS

For details on session parameter specification at statement or element level and evaluation at compilation time or at runtime, refer to Session Parameter Specification/Evaluation Overview.

PSEUDO - CICS Pseudo-Conversational Mode

This Natural profile parameter is for mainframes only.

It controls the mode of operation under CICS. When Natural is executing under control of the TP monitor CICS, two modes are possible: conversational and pseudo-conversational.

Possible settings	ON	PSEUDO=ON enables pseudo-conversational mode. In this mode, a Natural session is a sequence of different transactions. After each output to the terminal, all Natural work areas are saved and the transaction is terminated. When the user responds to a message by pressing ENTER (or any other input key), a new transaction is initiated. The Natural work areas are restored, the terminal input is read and the Natural session is continued. The transaction identification of each following transaction can be set dynamically by calling the subroutine CMTRNSET, which is provided in the library SYSEXTP.
	OFF	PSEUDO=OFF disables pseudo-conversational mode and enables conversational mode. In conversational mode, a Natural session is one transaction which is active for as long as the Natural session is active. **Note for CICS:** A specification of PSEUDO=OFF is ignored for Natural server sessions under CICS.
Default setting	ON	
Dynamic specification	YES	
Specification within session	NO	

For more information, refer to Natural under CICS, section TYPE - Thread Type for Group in the Natural TP Monitor Interfaces documentation.

RCA - Resolve Addresses of Static Non-Natural Programs

This Natural profile parameter is for mainframes only.

It controls the "dynamic" linking of static non-Natural programs to the Natural nucleus during initialization of the Natural session.

Possible settings	ON	At Natural startup, the list of all static non-Natural programs to be linked to Natural is scanned and a load request is issued for all modules whose addresses are unresolved. If a load request fails, no error message is issued. The use of RCA=ON is *not* recommended, as it causes a lot of processing overhead at Natural startup.
	OFF	No dynamic linking of static non-Natural programs is performed.
	name-list	If RCA=*name-list* is specified the list of static non-Natural programs to be linked to Natural is extended by the specified name list. A load request is issued for these modules even if they are already linked. In this way, it is possible to replace linked non-Natural programs. If a load request fails, an error message is issued. If more than one name is specified, each must be separated from the next by a comma and the list must be enclosed within parentheses as shown below: `RCA=(PROGRAM1,PROGRAM2,PROGRAM3)`
Default setting	OFF	
Dynamic specification	YES	
Specification within session	NO	

Static non-Natural programs have to be defined for being linked to Natural either internally (by using the macro NTINV within the modules NATPARM and NATCONFG) or externally (by using the profile parameter CSTATIC).

If the external name of the non-Natural program is different from the internal one (as used by the CALL statement), you can use either the profile parameter RCALIAS or the macro NTALIAS to define which external name is to be used for the load request.

Under CICS:
A PPT entry has to be defined to allow the load request for a non-Natural program. Static non-Natural programs are called via standard linkage conventions rather than EXEC CICS LINK requests.

RCALIAS - External Name Definition for Non-Natural Programs

This Natural profile parameter is for mainframes only.

It can be used to define the external names of static non-Natural programs which are defined by profile parameter RCA and loaded for dynamic linking during the initialization of a Natural session. It corresponds to the NTALIAS macro in the parameter module NATPARM.

Possible settings	*internal-program-name* *external-program-name*	List of name pairs: *internal-program-name* defines the internal name of a non-Natural program (used by the CALL statement) that must also be defined by the parameter RCA or CSTATIC (only if RCA=ON). *external-program-name* defines the corresponding external alias name for the load request during session initialization.
	OFF	No external names for RCA modules are defined. This value can be specified dynamically only.
Default setting	OFF	
Dynamic specification	YES	This parameter can only be specified dynamically. In the Natural parameter module NATPARM, the macro NTALIAS must be used instead.
Specification within session	NO	

The following topics are covered below:

- RCALIAS Parameter Syntax
- NTALIAS Macro Syntax
- Examples of NTALIAS Macro
- Example of RCALIAS Parameter

RCALIAS Parameter Syntax

The parameter syntax of RCALIAS is as follows:

```
RCALIAS=(
internal-program-name1,external-program-name1,internal-program-name2,external-program-name2,...)
```

NTALIAS Macro Syntax

The NTALIAS macro is specified as follows:

```
NTALIAS internal-program-name,external-program-name
```


Examples of NTALIAS Macro

```
NTALIAS PROGRAM1,ALIAS1
NTALIAS PROGRAM2,ALIAS2
```

Example of RCALIAS Parameter

```
RCA=(PROGRAM1,PROGRAM2),RCALIAS=(PROGRAM1,ALIAS1,PROGRAM2,ALIAS2)
```


RCFIND - Handling of Response Code 113 for FIND Statement

RCFIND - Handling of Response Code 113 for FIND Statement

This Natural profile parameter is for:

- Mainframes
- UNIX and Windows

It determines the action to be taken if Adabas Response Code 113 (requested ISN not found) is returned during the execution of a FIND statement processing loop.

Possible settings	ON	Response Code 113 causes the program to be terminated.
	OFF	Response Code 113 will be ignored, and processing of the FIND loop will continue by reading the next record.
Default setting	ON	
Dynamic specification	YES	
Specification within session	NO	

RCGET - Handling of Response Code 113 for GET Statement

This Natural profile parameter is for:

- Mainframes
- UNIX and Windows

It determines the action to be taken if Adabas Response Code 113 (requested ISN not found) is returned during the execution of a GET statement.

Possible settings	ON	Response Code 113 causes the program to be terminated.
	OFF	Response Code 113 will be ignored, the system variable *ISN will be set to "0", and processing will continue.
Default setting	ON	
Dynamic specification	YES	
Specification within session	NO	

RDACT - Remote Debugger Activation

This Natural profile parameter is for mainframes only.

It is reserved for internal use by Natural. Do not change its setting.

RDACTIVE - Activate Remote Debugger

This Natural profile parameter is for UNIX and Windows platforms only.

It specifies whether a remote debugger on a Windows NT computer is to be used or not. RDACTIVE is only relevant if the **system command** DEBUG has been used. It is used in combination with the profile parameters RDNODE (where the relevant node name is specified) and PDPORT (where the port number is specified).

Possible settings	ON	Remote debugging is enabled. For DCOM (Windows only) or RPC server, the remote debugging session is opened automatically.
	OFF	UNIX: No debugging is possible.
		Windows: Remote debugging is not possible. Local debugging can be performed.
Default setting	OFF	
Dynamic specification	YES	
Specification within session	NO	

RDCEXIT - Define Natural Data Collector User Exits

This Natural profile parameter is for mainframes only.

It is used to define user exits for the Natural Data Collector of the SYSRDC utility and, optionally, a work area size for each exit. If linked, the exit gets control from the Natural Data Collector at certain points within Natural. Specific session information is passed to the exits.

Possible settings	A list of user exit names with a work area size for each user exit.	With RDCEXIT, any exit names can be specified. In the Natural parameter module NATPARM, the exit names are automatically added to the CSTATIC profile parameter's setting list.	
Default setting	none		
Dynamic specification	YES	If RDCEXIT is specified dynamically, the exits must be defined in the profile parameters CSTATIC or RCA (RCA can also be specified dynamically). Optionally, the size of the exit work area can be specified after the exit name.	
		Possible settings	400 - 32760
		Default setting	400
		Example: `RDCEXIT=(MYEXIT,2000,RDCEX1)`	
Specification within session	NO		

For details, refer to Debugging and Monitoring in the SYSRDC Utility documentation.

RDCSIZE - Size of Buffer for the Natural Data Collector

This Natural profile parameter is for mainframes only.

It is used to specify the buffer size for the Natural Data Collector which is used by the SYSRDC utility. In addition, it controls the trace recording function of the data collector.

Alternatively, you can use the equivalent Natural profile parameter DS or macro NTDS (see Using Optional Macros in a Natural Parameter Module in the Natural Operations for Mainframes documentation) to specify the buffer size.

Possible settings	2 - 128	Buffer size in KB. To activate the data collector (without trace recording), you specify RDCSIZE=2. To also activate the trace recording, you have to set RDCSIZE to a setting greater than "2". If the requested space is not available, the Natural Data Collector cannot be used.
	0	Deactivates the data collector.
Default setting	0	
Dynamic specification	YES	
Specification within session	NO	

For details, refer to the SYSRDC Utility documentation.

RDNODE - Remote Debugger Node Name

RDNODE - Remote Debugger Node Name

This Natural profile parameter is for:

- Mainframes
- UNIX and Windows

Reserved for internal use by Natural. Do not change its setting.

RDPORT - Remote Debugger Port

This Natural profile parameter is for:

- Mainframes
- UNIX and Windows

Reserved for internal use by Natural. Do not change its setting.

RDS - Define Remote Directory Server

For static specification, this parameter is available as a keyword subparameter of the NTRPC macro. For dynamic specification, this parameter is available as a subparameter of the profile parameter RPC.

RDS is specified on the client side only.

allows you to define up to 10 remote directory servers. For each remote directory server, you specify up to 5 subparameters.

Possible settings	*server-name*	The server name (1 - 8 characters). There is no default, the value must be specified.
	server-node	The server node (1 - 8 characters). There is no default, the value must be specified.
	subprogram	The name of the subprogram titled CALLNAT, (1 - 8 characters) to be used as interface (default is RDSSCDIR).
	logon-indicator	A logon indicator. If nothing is specified, blank is the default. L The client initiates a Natural logon to the server with the library name of the current library on the client. blank No server logon will be executed. Note for Windows platforms: instead of "L", check the selection box.
	transport-protocol-name	The transport protocol to be used. ACI is the only possible value and the default.
Default setting	None	Subparameter defaults, see above.
Dynamic specification	YES	See below.
Specification within session	NO	

For dynamic specification the syntax is as follows.

Using 1 server:

```
RDS=(server-name,server-node-name,subprogram,logon-indicator,transport-protocol-name)
```

Using 2-10 servers:

```
RDS=((server-name,server-node
name,subprogram,logon-indicator,transport-protocol-name)(server-name,server-node
name,subprogram,logon-indicator,transport-protocol-name)...(server-name,server-node
name,subprogram,logon-indicator,transport-protocol-name))
```


For additional information on Natural RPC, see the Natural Remote Procedure Call documentation.

READER - VSE/ESA System Logical Units for Input

This Natural profile parameter is for mainframes only.

It indicates the VSE/ESA system logical units which are to be used by Natural for input.

Possible settings	READER=(*n*,*device*,...)	*n* is "0" for CMSYNIN and "1" for CMOBJIN *device* is either "SYSRDR" or "SYSIPT"
Default setting	READER=(0,SYSRDR,1,SYSIPT)	By default, the primary input stream (CMSYNIN) is read from SYSRDR and the input stream (CMOBJIN) is read from SYSIPT (if required). If CMSYNIN or CMOBJIN are disk or tape files, the associated READER subparameter is ignored.
Dynamic specification	YES	
Specification within session	NO	

This overwriting of a system logical unit number only applies if the relevant file is a card file.

RECAT - Dynamic Recataloging

This Natural profile parameter is for:

- Mainframes
- UNIX and Windows

This parameter indicates the action to be taken if Natural detects an inconsistency in the global data area definition as defined in the program currently being executed; that is, the global data area in the program does not correspond to the definition of the global data area currently in use.

Possible settings	ON	 This profile parameter only applies to Natural objects of Version 2.3 and above. An error message is issued if an inconsistency concerning a Version 2.2 program and/or global data area is detected. Natural automatically adjusts the object and disables the **system commands** CATALOG, PURGE and SAVE. **Note:** If the adjusted object invokes an object from a steplib library that also has to be adjusted, the object from the steplib library will be copied to the library of the invoking object.
	OFF	Natural issues an error message.
Default setting	OFF	
Dynamic specification	YES	
Specification within session	NO	

REINP - Issue Internal REINPUT Statement for Invalid Data

REINP - Issue Internal REINPUT Statement for Invalid Data

This Natural profile and session parameter is for all platforms.

By default, Natural automatically issues an internal REINPUT statement if invalid data have been entered in response to an INPUT statement. With this parameter, you can switch this mechanism off. This will allow you to handle such input errors yourself in your application.

Within a Natural session, the profile parameter REINP can be overridden by the session parameter REINP.

Possible settings	ON	An internal REINPUT statement is issued when invalid data have been entered.	
	OFF	An internal REINPUT statement is not issued when invalid data have been entered.	
Default setting	ON		
Dynamic specification	YES		
Specification within session	YES	Applicable Statements:	SET GLOBALS
		Applicable Command:	GLOBALS

For details on session parameter specification at statement or element level and evaluation at compilation time or at runtime, refer to Session Parameter Specification/Evaluation Overview.

RELO - Storage Thread Relocation

This Natural profile parameter is for mainframes only.

It controls the relocation of the Natural thread after a terminal I/O in a thread environment (CICS, Com-plete, IMS/TM, UTM and Natural as a Server under OS/390).

Possible settings	ON	The Natural thread and all the buffers contained therein can be relocated to another storage area if the original storage area has been occupied by another user after a terminal I/O.
	OFF	No relocation is performed. The Natural thread and all the buffers therein remain located at the same virtual address after the terminal I/O. This setting applies to CICS and Com-plete only. In all other thread environments, Natural cannot guarantee that the thread remains located at the same address. **Notes for CICS:** • When using TYPE=GETM threads under CICS, RELO=OFF has the same effect as the PSEUDO=OFF setting of the PSEUDO profile parameter. See also TYPE (thread type for group) in the section Natural under CICS in the TP Monitor Interfaces documentation. • A specification of RELO=OFF is ignored for Natural server sessions under CICS using TYPE=GETM threads.
	FORCE	This will force a relocation of the Natural thread and all the buffers contained therein to another storage area. This can be useful for testing purposes in some environments. This setting applies to CICS, Com-plete and IMS/TM only.
Default setting	ON	
Dynamic specification	YES	
Specification within session	NO	

RFILE - File for Recordings

This Natural profile parameter is for mainframes only.

It determines where recordings (that is, the data recorded by the Recording function) are stored.

Possible settings	SPAD	Recordings will be stored in the scratch-pad file. (If no scratch-pad file is defined, the recordings will be stored in the system file FUSER.)
	FUSER	Recordings will be stored in the system file FUSER.
	FNAT	Recordings will be stored in the system file FNAT.
Default setting	SPAD	
Dynamic specification	YES	
Specification within session	NO	

For details on the Recording function, see the Recording utility, Debugging and Monitoring, in the Natural Utilities documentation..

RI - Release ISNs

This Natural profile parameter is for mainframes only.

It indicates whether ISNs (internal sequence numbers) for records which were read and placed in hold status but were not updated are to be retained in hold status.

Possible settings	ON	Natural releases the ISN of each record which has been placed in hold status but was not updated (for example because the record was rejected as a result of a WHERE clause or an ACCEPT/REJECT statement). This reduces the number of ISNs which are contained in the hold queue. This may, however, cause additional performance overhead as an Adabas call is required for each ISN released.
	OFF	The ISN of each record which has been placed in hold status is *not* released until the end of the transaction.
Default setting	OFF	
Dynamic specification	YES	
Specification within session	NO	

In nested processing loops, a record which due to RI=ON is released in an inner processing loop is no longer kept in hold status for any outer loop.

RJESIZE - Initial Size of NATRJE Buffer

This Natural profile parameter is for mainframes only.

With the Natural utility NATRJE (described in the Natural Utilities documentation), JCL jobs can be collected and then submitted all at once. RJESIZE is used to set the initial size of the buffer which holds the JCL jobs before they are submitted.

Alternatively, you can use the equivalent Natural profile parameter DS or macro NTDS (see Using Optional Macros in a Natural Parameter Module in the Natural Operations for Mainframes documentation) to specify RJESIZE.

Possible settings	1 - 256	Buffer size in KB. If the initial size is not sufficient, Natural automatically increases (repeatedly if necessary) the buffer size by 8 KB.
	0	Disables the NATRJE utility
Default setting	8	
Dynamic specification	YES	
Specification within session	NO	

RM - Retransmit Modified Fields

This Natural profile parameter is for mainframes only.

Some TP monitors translate input data automatically to upper-case characters. As Natural's screen optimization only retransmits modified data back to the screen, the TP-monitor translation may cause input for a field which has been modified not to be retransmitted.

Possible settings	ON	Natural always sends back all modified fields.
	OFF	Natural sends back modified fields only if they have been changed.
Default setting	OFF	
Dynamic specification	YES	
Specification within session	NO	

ROSY - Read-Only Access to System Files

This Natural profile parameter is for:

- Mainframes
- UNIX and Windows

It disables modifications on the Natural system files FNAT, FUSER and FSEC.

Possible settings	ON	No data can be written to, modified on or deleted from the system files. Natural issues an error message instead of performing any action that would modify any of these system files.
	OFF	Data can be written to, modified on and deleted from the system files.
Default setting	OFF	
Dynamic specification	YES	
Specification within session	NO	

Notes:

- If your system files are specified as read-only (ROSY=ON), the Natural functions SETUP/RETURN, Recording, NATPAGE and invoking of helproutines cannot be used, because they write data to the Natural system files FNAT and/or FUSER.
- Therefore, it is recommended that you allocate and use a so-called scratch-pad file to hold these temporary data. The scratch-pad file is optional and must be defined as recoverable by using the macro NTLFILE or the profile parameter LFILE. The above functions then write their data to this file instead of FNAT/FUSER.
- With ROSY=OFF, a scratch-pad file should also be defined if you use the Recording and NATPAGE functions with database transaction logic, as that might lead to unpredictable results with FNAT/FUSER.

RPC - Remote-Procedure-Call Settings

This Natural profile parameter is for mainframes only.

It allows you to specify subparameters which control the handling of Natural RPC. It corresponds to the macro NTRPC in the parameter module NATPARM.

Possible settings	Subparameters	see Keyword Subparameters below.
Default setting	None	
Dynamic specification	YES	This parameter can only be specified dynamically. In the Natural parameter module, the macro NTRPC must be used instead.
Specification within session	NO	

The following topics are covered below:

- RPC Parameter Syntax
- NTRPC Macro Syntax
- Keyword Subparameters
- Example of RPC Parameter
- Example of NTRPC Macro

RPC Parameter Syntax

The parameter syntax of RPC is as follows:

```
RPC=(keyword_subparameter1=value,keyword_subparameter2=value,...)
```

keyword_subparameter - see Keyword Subparameters below.

NTRPC Macro Syntax

The syntax of the NTRPC macro in the Natural parameter module is as follows:

```
NTRPC keyword_subparameter1=value,keyword_subparameter2=value,...
```

keyword_subparameter - see below.

Keyword Subparameters

There are three groups of keyword subparameters available that apply to

- **both Client and Server**
  (ACIVERS | MAXBUFF | RPCSIZE | SERVER)
- **the Server only**
  (LOGONRQ | NTASKS | SRVNAME | SRVNODE | SRVUSER | TRACE | TRANSP)

- **the Client only**
  (AUTORPC | COMPR | DFS | RDS | TIMEOUT | TRYALT)

Refer to the descriptions of the corresponding profile parameters which are applicable to UNIX and Windows.

RPC Parameter Example

```
RPC=(RPCSIZE=80,SRVNAME=MYSERV,SERVER=ON,DFS=(SRV2,NODE1,,ACI))
```

NTRPC Macro Example

```
NTRPC RPCSIZE=80,
      SRVNAME=MYSERV,
      SERVER=ON,
      DFS=(SRV2,NODE1,,ACI),
      RDS=((SRVX,NODEX),(SRVY,NODEY)
```

For additional information on Natural RPC, see the Natural Remote Procedure Call documentation.

RPCSIZE - Size of Buffer Used by Natural RPC

RPCSIZE - Size of Buffer Used by Natural RPC

This parameter is for mainframes only.

For static specification, this parameter is available as a keyword subparameter of the NTRPC macro. For dynamic specification, this parameter is available as a subparameter of the profile parameter RPC.

RPCSIZE determines the size of the buffer used by Natural RPC. It can be specified on both the client and the server side.

Possible settings	1 - 2097151	Buffer size in KB.
	0	Natural RPC cannot be used.
Default setting	0	
Dynamic specification	YES	
Specification within session	NO	

For additional information on Natural RPC, see the Natural Remote Procedure Call documentation.

RUNSIZE - Size of Runtime Buffer

This Natural profile parameter is for mainframes only.

It sets the size of the Natural runtime buffer.

Alternatively, you can use the equivalent Natural profile parameter DS or macro NTDS (see Using Optional Macros in a Natural Parameter Module in the Natural Operations for Mainframes documentation) to specify RUNSIZE.

Possible settings	10 - 64	Buffer size in KB.
Default setting	16	
Dynamic specification	YES	
Specification within session	NO	

The Natural runtime buffer contains information on the following items:

- defined STEPLIBS,
- the file translation table (profile parameter TF),
- log information of the most recent command,
- the environment stack (for user settings),
- active global data areas,
- invoked subroutines (subroutine name and object name),
- invoked objects (address in the buffer pool for a fast location).

If the specified size of the runtime buffer is exceeded by a Natural user, the size for the invoked objects information is decreased accordingly. However, when this size decreases, the number of possible buffer-pool fast locations decreases, too; if it is about to become 0, an error message is issued.

SA - Sound Terminal Alarm

This Natural profile and session parameter is for all platforms.

It specifies whether the terminal alarm feature is to be used.

Within a Natural session, the profile parameter SA can be overridden by the session parameter SA.

Possible settings	ON	The terminal alarm is sounded each time the user is prompted for input by Natural. **Note:** The use of this feature requires that the terminal alarm hardware feature has been installed for the terminal.	
	OFF	No terminal alarm is used for input prompting, however, the alarm may still be activated with the SOUND ALARM option of the REINPUT statement.	
Default setting	OFF		
Dynamic specification	YES		
Specification within session	YES	Applicable Statements:	SET GLOBALS
		Applicable Command:	GLOBALS

For details on session parameter specification at statement or element level and evaluation at compilation time or at runtime, refer to Session Parameter Specification/Evaluation Overview.

SCTAB - Scanner Characters

This Natural profile parameter is for mainframes only.

It allows you to overwrite the definitions in the scanner character-type table NTSCTAB as contained in the configuration module NATCONFG. The NTSCTAB table defines the properties of characters

- used in mask definitions for the SCAN/MASK functions,
- recognized as delimiters in the SEPARATE statement.

SCTAB corresponds to the NTSCTAB macro in the Natural parameter module NATPARM.

Possible settings	See SCTAB Parameter Syntax below.	
Default setting	As specified within the macro NTSCTAB in NATCONFG.	
Dynamic specification	YES	This parameter can only be specified dynamically. In the Natural parameter module NATPARM, the macro NTSCTAB must be used instead.
Specification within session	NO	

The following topics are covered below:

- SCTAB Parameter Syntax
- NTSCTAB Macro Syntax
- Example of NTSCTAB Macro
- Example of SCTAB Parameter

SCTAB Parameter Syntax

The SCTAB parameter is specified as follows:

```
SCTAB=(character1,attribute-type1,attribute-type2,...,character2,attribute-type1,attribute-type2,...)
```

character

You specify a character, and after it its attribute type(s).

You can specify the character either as the one-byte character itself (enclosed in apostrophes) or as the hexadecimal representation of that character.

attribute-type(s)

Attribute types can be:

UPPER	upper-case alphabetical
LOWER	lower-case alphabetical
NUM	numeric
HEX	hexadecimal
ALFANUM	alphanumeric
SPECIAL	special
NDELIM	non-delimiter

It is possible to specify more than one character in the list of values. You must enclose the entire string of character/attribute pairs in parentheses.

NTSCTAB Macro Syntax

The NTSCTAB macro is specified as follows:

```
NTSCTAB character1,attribute-type1,attribute-type2,...
NTSCTAB character2,attribute-type1,attribute-type2,...
...
```

For each character to be overwritten, you have to specify a separate NTSCTAB macro.

Example of NTSCTAB Macro

```
NTSCTAB  5E,LOWER,NDELIM
NTSCTAB  'ß',SPECIAL
NTSCTAB  7B,SPECIAL
NTSCTAB  Ä,UPPER,NDELIM
```

Example of SCTAB Parameter

```
SCTAB=(5E,LOWER,NDELIM,'ß',SPECIAL,7B,SPECIAL,'Ä',UPPER,NDELIM)
```


SD - Time Delay between Two Screens

This Natural profile parameter is for UNIX and Windows platforms only.

It delays the time related to screen output display. This is the time delay between two screens during a non-conversational write operation (see the Natural terminal command %N).

Possible settings	0 - 100	The unit for the specified setting is a tenth of a second, which means that "SD=10" leads to a delay of one second.
Default setting	0	
Dynamic specification	NO	
Specification within session	NO	

SENDER - Screen Output Destination for Asynchronous Processing

This Natural profile parameter is for mainframes and only applies under CICS, Com-plete, IMS/TM and UTM.

It specifies the destination where output from an asynchronous application is to be displayed. The destination specified applies to hardcopy output and primary reports.

Possible settings	1 to 8 characters	Output destination, e.g. printer.
Default setting	none	
Dynamic specification	YES	
Specification within session	NO	

Any additional reports are sent to the destinations specified with the DEFINE PRINTER statement (just as in a synchronous online session).

The following platform-specific characteristics apply:

Platform:	**Comment:**
CICS	The profile parameter SENDER specifies the CICS transient data (TD) destination and the terminal or printer for terminal output from asynchronous sessions. If the specified destination does not exist, the session output is sent to the specified terminal or printer. If the specified terminal or printer does not exist either, the session terminates abnormally. The default terminal output format for asynchronous sessions is a 3270 data stream. If the SENDER terminal specification is not a 3270 device, the Natural application must switch to the correct terminal type before the first output statement (for example, by specifying SET CONTROL 'T=PRNT' for a printer or by starting with profile parameter TTYPE=PRNT). If you are routing all output to a (spool) destination, such as CSSL, the Natural application must be switched to line mode, for example, by specifying SET CONTROL 'T=*xxxx*' or by starting with profile parameter TTYPE=*xxxx*, where *xxxx* is BTCH or ASYL. In this case, two other profile parameters are relevant: EJ and INTENS. If you set EJ=ON, all lines are routed with a leading ASA control character. With EJ=OFF, there is no leading ASA control character. INTENS should be set to "1", particularly if you have set EJ=OFF. For other CICS-specific functionality, see Asynchronous Natural Sessions under CICS (in the Natural TP Monitor Interfaces documentation).
IMS/TM	The profile parameter SENDER specifies the default LTERM. This LTERM is always used when no other printer has been specified. You should always dynamically define a SENDER parameter in the NIIBOOT module. This is important when Natural tries to output error messages when starting a session: if no SENDER parameter is specified, there is no valid LTERM and NATIMS terminates the session.
UTM	The profile parameter SENDER specifies the ID for the initialization of an asynchronous transaction; that is, the ID which identifies the transaction as asynchronous. If output from an asynchronous transaction is to be printed, the setting specified with the SENDER parameter also identifies the printer on which the output is to be printed.

SENDER - Screen Output Destination for Asynchronous Processing

For further information, see also the profile parameter OUTDEST and Asynchronous Processing (in the Natural Operations for Mainframes documentation).

SERVER - Start Natural Session as an RPC Server Session

For static specification, this parameter is available as a keyword subparameter of the NTRPC macro. For dynamic specification, this parameter is available as a subparameter of the profile parameter RPC.

SERVER specifies whether or not the Natural session will be started as an RPC server session. It can be specified on both the client and the server side.

Possible settings	ON	The Natural session will be started as an RPC server session.
	OFF	The Natural session will not be started as an RPC server session.
Default setting	OFF	
Dynamic specification	YES	
Specification within session	NO	

For further information, see the Natural Remote Procedure Call documentation.

SF - Spacing Factor

This Natural profile and session parameter is for all platforms.

It specifies the default number of spaces to be inserted between field settings of columns on Natural reports created using a DISPLAY statement.

Within a Natural session, the profile parameter SF can be overridden by the session parameter SF.

Possible settings	1 - 30	Number of spaces. The SF parameter cannot be set to "0"; that is, at least one blank character must be placed between report columns.	
Default setting	1		
Dynamic specification	YES		
Specification within session	YES	Applicable Statements:	SET GLOBALS
		Applicable Command:	GLOBALS

For details on session parameter specification at statement or element level and evaluation at compilation time or at runtime, refer to Session Parameter Specification/Evaluation Overview.

SHELL - Grant Shell Access to Natural User

This Natural profile parameter is for UNIX only.

It can only be set by Natural administrators. This parameter allows or disallows the shell exit from the Natural Main Menu for specific users.

Be careful when granting shell access to a Natural user. This might introduce security problems, because the user would then be able to use the underlying features of the operating system.

Possible settings	YES	Shell exit is allowed. If the shell exit is allowed for a user, the name of the shell that will be started for this user can be specified by the administrator. This shell name must be the full path name of an executable UNIX program, for example, "/bin/csh".
	NO	Shell exit is not allowed.
Default setting	NO	
Dynamic specification	NO	
Specification within session	NO	

SI - Shift-In Code for Double-Byte Character Set

This Natural profile parameter is for mainframes only.

It is only relevant for Asian countries which use double-byte character sets (DBCS). The parameter is used to specify a shift-in code.

Possible settings	0F	Shift-in code for IBM hardware.
	29	Shift-in code for Fujitsu hardware.
Default setting	none	
Dynamic specification	YES	
Specification within session	NO	

The shift-in code is used to indicate the point at which the code of character representation is shifted from double-byte mode back into normal (single-byte) mode. The beginning of the double-byte character representation (shift-out code) is indicated by the setting defined with the profile parameter SO.

SKEY - Storage Protection Key

This Natural profile parameter is for mainframes and only applies under Com-plete.

It determines whether Natural runs under the same storage protection key as Com-plete.

Possible settings	ON	Natural runs under the same storage protection key as Com-plete.
	OFF	Natural runs under a different storage protection key than Com-plete.
Default setting	ON	
Dynamic specification	YES	
Specification within session	NO	

See the Com-plete documentation for details on storage protection keys.

SL - Source Line Length

This Natural profile parameter is for mainframes only.

It specifies the number of characters to be interpreted on each Natural source code line. This also applies to the line mode editor which is activated with the system command EDT.

Within a Natural session, the profile parameter SL can be overridden by the session parameter SL.

Possible settings	20 - 250	**In batch mode:**	The number of characters to be processed on each line in the datasets CMSYNIN and CMOBJIN. For details on these datasets, refer to the operating-system-specific parts of the section Natural in Batch Mode (in the Natural Operations for Mainframes documentation).
		In online mode:	The number of characters to be interpreted when using the Natural program editor in EDT mode (as activated with the system command EDT).
Default setting	72		
Dynamic specification	YES		
Specification within session	YES	Applicable Statements:	SET GLOBALS
		Applicable Command:	GLOBALS

For details on session parameter specification at statement or element level and evaluation at compilation time or at runtime, refer to Session Parameter Specification/Evaluation Overview.

SM - Programming in Structured Mode

This Natural profile and session parameter is for all platforms.

It determines whether or not structured mode must be used.

If structured mode (SM=ON) was specified by the Natural administrator during Natural installation, this parameter cannot be set to "OFF". Otherwise with SM=OFF, the mode may be changed as desired.

Within a Natural session, the profile parameter SM=OFF can be overridden by the session parameter SM=ON.

Possible settings	ON	Structured mode syntax must be used. If specified during the session, GLOBALS SM=OFF is disabled. (The message "Reporting mode not permitted" will be displayed).	
	OFF	Programming can be done in either structured mode or reporting mode.	
Default setting	OFF		
Dynamic specification	YES		
Specification within session	YES	Applicable Statements:	SET GLOBALS
		Applicable Command:	GLOBALS

For details on session parameter specification at statement or element level and evaluation at compilation time or at runtime, refer to Session Parameter Specification/Evaluation Overview.

If Natural Security is installed,

- the setting of the mode option in the library’s security profile determines whether the SM profile parameter can be used; see also Programming mode (in the Natural Security documentation).
- this parameter may be disabled by Natural Security to the effect that structured mode is invariably in effect for a given library.

SNAT - Sound a Bell at Syntax Error

This Natural profile parameter is for UNIX and Windows platforms only.

It is used to sound a bell when the compiler detects a syntax error in a Natural program.

Possible settings	ON	A bell will sound when a syntax error is encountered.
	OFF	No bell will sound in the case of syntax errors.
Default setting	OFF	
Dynamic specification	NO	
Specification within session	NO	

SO - Shift-Out Code for Double-Byte Character Set

This Natural profile parameter is for mainframes only.

It is only relevant for Asian countries which use double-byte character sets (DBCS). The parameter is used to specify a shift-out code.

Possible settings	0E	Shift-out code for IBM hardware.
	28	Shift-out code for Fujitsu hardware.
Default setting	none	
Dynamic specification	YES	
Specification within session	NO	

The shift-out code is used to indicate the point at which the code of character representation is shifted out of normal (single-byte) mode into double-byte mode. The end of the double-byte character representation (shift-in code) is indicated by the setting defined with the profile parameter SI.

SORT - Control of Sort Program

This Natural profile parameter is for mainframes only.

It is used to control the sort program used for the processing of SORT statements. It corresponds to the NTSORT macro in the Natural parameter module NATPARM.

SORT or NTSORT can be used to specify various options that control the handling of the sort program used when a SORT statement is executed.

The sort program to be used can be either Natural's internal one (the default for all environments) or an external one. The type of sort to be used depends on the setting of the keyword subparameter EXT.

Possible settings		For an explanation of the individual options and their possible settings, see SORT Parameter Syntax below.
Default setting		`WRKSIZE=10,STORAGE=MAIN,EXT=OFF,EXTNAME=SORT,EXTEOJ=OFF`
Dynamic specification	YES	This parameter can only be specified dynamically. In the Natural parameter module NATPARM, the macro NTSORT must be used instead.
Specification within session	NO	

The following topics are covered below:

- SORT Parameter Syntax
- NTSORT Macro Syntax
- Keyword Subparameters

SORT Parameter Syntax

The SORT parameter is specified as follows:

```
SORT=(WRKSIZE=nnn,STORAGE=medium,EXT=ON/OFF,EXTNAME=name,EXTOPT=(options,...),EXTEOJ=ON/OFF)
```

NTSORT Macro Syntax

The NTSORT macro is specified as follows:

```
NTSORT
WRKSIZE=nnn,STORAGE=medium,EXT=ON/OFF,EXTNAME=name,EXTOPT=(options,...),EXTEOJ=ON/OFF
```

The individual keyword subparameters are explained below.

Keyword Subparameters

The following keyword subparameters are available: WRKSIZE | STORAGE | EXT | EXTNAME | EXTOPT | EXTEOJ

WRKSIZE - Size of Work Buffer Used by Sort Program

WRKSIZE specifies the size *nnn* (in KB) of the work buffer used by the sort program.

Possible values:	10 to 256, or 0.
Default value:	10

If you specify WRKSIZE=0, no sort operations can be performed.

The work buffer specified by WRKSIZE accommodates internal sort control data. The remaining storage is used to collect and sort the records. The size of the sort control data depends on various factors (the WRKSIZE itself, the sort record length, the number of sort keys, their size and format) and can therefore not be calculated in a formal way.

STORAGE - Type of Storage Medium

STORAGE specifies the type of storage *medium* to be used by Natural's internal sort program.

In general, the SORT statement first tries to use the remaining storage in WRKSIZE. If the number of records does exceed this storage, the SORT statement tries to use intermediate storage to additionally process records:

MAIN	(default), only the remaining storage of WRKSIZE is used, no other intermediate storage is available.
BP	The SORT buffer pool is used as intermediate storage.
SD	SD files are used as intermediate storage. This value is honored under Com-plete only.
SMARTS	SMARTS portable file system.

Note Concerning SD Files under Com-plete

The files are allocated as temporary SD files. They are allocated for a stack level. This means, the name of the SORT SD files are &&STsnnn with:

&&	Indicator for a temporary SD file
ST	Standard prefix for the SD SORT file
s	Stack level
nnn	Sequence number within a single SORT run

Note Concerning Files under Com-plete/SMARTS

SMARTS work files are located in the SMARTS Portable File System. The path must be specified with the SMARTS environment variable $NAT_WORK_ROOT. A special directory named "sort" for sort workfiles is added and for each user, a different directory is created named by the user ID. The resulting directory is then $NAT_WORK_ROOT/sort/userid. The naming of the sort work files corresponds to the SD files under Com-plete.

EXT - Use of External Sort Program

EXT specifies if an external sort program is to be used or not:

ON	An external sort program will be used. The use of an external sort program is possible only in batch environments, including IMS/BMP, TSO, TIAM and CMS.
OFF	The Natural sort program will be used (this is the default).

EXTNAME - Name of External Sort Program

This subparameter does not apply to BS2000/OSD.

EXTNAME specifies the *name* (1 to 8 characters) of the external sort program to be used. The default name is "SORT".

EXTOPT - Additional Options for External Sort Program

This subparameter does not apply to BS2000/OSD.

EXTOPT specifies additional *options* for the external sort program.

Natural generates the necessary field and format parameters and passes them to the external sort program. With EXTOPT, you can specify additional parameters to be passed to the external sort program. You can only specify parameters that are part of the control statement syntax of your external sort program.

You can specify up to two option strings which are delimited by a slash (/). The first option string is appended to the SORT control statement, the second option string is used to build an OPTION control statement. You may omit the option string before or after the slash. If the option string after the slash is omitted no OPTION control statement is generated at all.

The whole option string must be enclosed in quotes (’). For compatibility reasons, it is still possible to have the option string enclosed in brackets instead.

For compatibility reasons, a single option string without a leading or trailing slash is handled differently. Depending on the underlying operating system, the options are appended to the following control statements:

OS/390 and VM/CMS:	SORT control statement
VSE/ESA:	OPTION control statement

EXTOPT Example:

The additional parameters can be specified as in the following example:

```
EXTOPT=(SIZE=E2000000,NOEQUALS,DYNALLOC=(3350,8))
EXTOPT='SIZE=E2000000,NOEQUALS,DYNALLOC=(3350,8)'
EXTOPT='SIZE=E2000000,NOEQUALS,DYNALLOC=(3350,8)/NOCHECK'
EXTOPT='/NOCHECK'
EXTOPT='WORK=4/'
```

EXTEOJ - Action in the Event of an Error

EXTEOJ specifies the action to be taken if an error is detected during the execution of the external sort program:

ON	If an error is detected, sort processing is terminated. ON requires that the sort program used is able to detect a return code of 16 from both the E15 and E35 sort exit routines.
OFF	If an error is detected, Natural withholds further calls to the sort program and ignores each record as it is passed to the E35 sort exit routine (this is the default).

SORTSZE - Size of Sort Buffer

This Natural profile parameter is for UNIX and Windows platforms only.

It indicates the amount of storage to be reserved for use by the sort program. This sort buffer is only allocated when executing a Natural program which contains a SORT statement.

Possible settings	500 - 2048	Buffer size in KB. Increasing the setting leads to faster sort processing, in particular when all data to be sorted fit into the sort buffer.
Default setting	500	
Dynamic specification	YES	
Specification within session	NO	

SRVNAME - Name of RPC Server

For static specification, this parameter is available as a keyword subparameter of the NTRPC macro. For dynamic specification, this parameter is available as a subparameter of the profile parameter RPC.

SRVNAME specifies the name of the RPC server, with which it registers on the node specified with the profile parameter SRVNODE.

It is specified on the server side only.

Possible settings	1 - 192 characters	Valid server name.
Default setting	none	
Dynamic specification	YES	
Specification within session	NO	

You may either specify a physical server name of up to 32 characters or a logical service name of up to 192 characters. In case of a logical service name, the SRVNODE must be set to * (intentionally left empty).

In case of an EntireX Broker node, the value of SRVNAME correponds to the value of the SERVER attribute of a service entry in the broker attribute file, as below:

```
CLASS=RPC, SERVICE=CALLNAT, SERVER=srvname
```

Example:

```
SRVNAME='PRODUCTION_SERVER'            /* physical server name */
SRVNAME='MY_LOGICAL_SERVICE,MY_SET'    /* logical server name */
```

For more details about Location Transparency and logical service names, refer to the EntireX documentation.

For additional information on Natural RPC, see the Natural Remote Procedure Call documentation.

SRVNODE - Name of Node

For static specification, this parameter is available as a keyword subparameter of the NTRPC macro. For dynamic specification, this parameter is available as a subparameter of the profile parameter RPC.

SRVNODE specifies the name of the node upon which an RPC server registers.

It is specified on the server side only.

Possible settings	1 - 192 characters	Node name.
Default setting	none	
Dynamic specification	YES	
Specification within session	NO	

You may either specify a physical node name of up to 32 characters, a logical node name of up to 192 characters or * (intentionally left empty) to indicate that SRVNAME contains a logical service name.

In case of an EntireX Broker node, a physical node name may refer to an Entire Net-Work node or to an TCP/IP address. Note that the broker stub in use must support the naming notation. For details about the structure of node names and their support by the broker stubs, refer to the EntireX documentation.

The examples below are based on the EntireX notation.

Example:

```
SRVNODE=ETB001                                /* Entire Net-Work node */
SRVNODE=PCBROKER                              /* host name for a TCP/IP address */
SRVNODE=157.189.160.95:1958:TCP             /* TCP/IP address with port number */
SRVNODE=tcpip://host.com:1958               /* host name for a TCP/IP address with port number */
SRVNODE=LOGBROK=MY_LOGICAL_NODE,MY_SET      /* logical node name */
SRVNODE=*                                   /* logical service name in SRVNAME */
```

If a host name is used for the TCP/IP address, the name must either be known to your DNS server or it must be defined in the hosts file of your TCP/IP configuration.

If the port number is ommited, either a default port number is used by the EntireX broker stub or a host name must be used, and the host name must be known to your DNS server or must be defined in the services file of your TCP/IP configuration.

For more details about Location Transparency and logical node names, refer to the EntireX documentation.

For additional information on Natural RPC, see the Natural Remote Procedure Call documentation.

SRVUSER - User ID for RPC Server Registry

For static specification, this parameter is available as a keyword subparameter of the NTRPC macro. For dynamic specification, this parameter is available as a subparameter of the profile parameter RPC.

SRVUSER specifies the user ID needed to register an RPC server on the node specified with the profile parameter SVRNODE. If this specification is omitted, the timestamp will be used.

In case of an EntireX Broker node, SRVUSER is also used to logon to the EntireX Broker. A password is either taken from Natural Security (see *NSC below) or specified via the application programming interface USR2072N.

SRVUSER is specified on the server side only.

Possible settings	1 - 16 characters	Valid user ID, *USER or *NSC. If SRVUSER is set to *USER, the Natural server uses the current Natural user ID (*USER) to logon to the node. If SRVUSER is set to *NSC and Natural Security is installed, the Natural server uses the current Natural user ID (*USER) and the password defined for this user ID in Natural Security to logon to the node.
Default setting	timestamp	
Dynamic specification	YES	
Specification within session	NO	

For additional information on Natural RPC, see the Natural Remote Procedure Call documentation.

SSIZE - Size of Source Area Allocated by the Editors

This Natural profile parameter is for:

- Mainframes
- UNIX and Windows

It determines the size of the buffer used by the Software AG Editor.

Alternatively, you can use the equivalent Natural profile parameter DS or macro NTDS (see Using Optional Macros in a Natural Parameter Module in the Natural Operations for Mainframes documentation) to specify the SSIZE value.

Possible settings	0, or 40 - 128	Buffer size in KB. If SSIZE=0 or if the required space is not available, the Software AG Editor cannot be used.
Default setting	64	
Dynamic specification	YES	
Specification within session	NO	

STACK - Place Data/Commands on the Stack

This Natural profile parameter is for:

- Mainframes
- UNIX and Windows

It is used to place data/commands on the Natural stack.

Note:
If STACK is used, the profile (or session) parameters DC, HI, IA, ID and STACKD must not be set to ":".

Possible settings	any character string	See below.
Default setting	HELLO	
Dynamic specification	YES	
Specification within session	NO	

The stack can contain a sequence of Natural commands and/or user-specified commands, together with their data, for execution at the beginning of the Natural session.

The command stack is processed before the user is prompted for input on the screen (TP mode) or data are read from CMSYNIN/CMOBJIN files (see Natural in Batch Mode in the Natural Operations for Mainframes documentation).

If an INPUT statement is encountered during stack processing, the corresponding input screen is generated only if the required input data were not supplied with the command when the stack was created. Any reports generated during stack processing are displayed as usual.

Each system or user-defined command can be optionally followed by data which are used to satisfy requests for information required during the processing of the command. The character string provided as data for the STACK parameter must be enclosed in parentheses. If the command is a user command (that is, the name of a user program), any data provided resolve the data requirements of INPUT statements within the user program.

Conventions:

- Multiple settings for one INPUT statement are separated by a comma (,).
- Data for multiple INPUT statements are separated by a colon (:).
- Commands are separated by the stack delimiter character defined by profile parameter STACKD. The default setting is a semicolon (;).

Examples:

```
STACK=(LOGON USER1;UCMD1 A,B;UCMD2 C,D:E;FIN)
STACK=OFF                                    No STACK data.
STACK=UCMND Execute command UCMND            No embedded blanks.
STACK=(CMD DATA:DATA;CMD...)                 Place commands/data on stack.
```

Since some commands (for example, GLOBALS) do not read parameters by INPUT, a blank character should be used rather than a colon to delimit a command from the first parameter data element.


```
STACK='LOGON SYSTEM'
```

Because the macro assembler does not allow embedded blanks within parentheses, the character string must be enclosed in apostrophes when specified as static parameter.

STACKD - Stack Delimiter Character

This Natural profile parameter is for mainframes only.

It defines the character to be used as the command delimiter for the STACK parameter and for command input under the Natural Development Server (product code: NDV) in a Natural Single Point of Development environment.

To avoid that the value specified for the STACK parameter or the data passed as command input under the Natural Development Server is not interpreted as intended, the STACKD parameter value should be set to a character that is not contained in the data passed if the data contains the default value of the stack delimiter character (see example below). The STACKD parameter should be changed to a character other than the default character if the ID parameter has been set to the semicolon.

Possible settings	any special character	The character must not be the same as the one specified with the ID profile/session parameter (input delimiter character), DC profile/session parameter (decimal character) or IA profile/session parameter (input assign character).
Default setting	; (semicolon)	
Dynamic specification	YES	
Specification within session	NO	

The character specified may be enclosed with single quotes. If the input delimiter character is to be a comma, it must be specified as ID=’,’, because the character "," separates individual parameters.

Example:

```
STACKD=’/’,ID=’;’ STACK=(DUMP IOB;+100/FIN)
```

To avoid that the semicolon after "DUMP IOB" is interpreted as a command delimiter, STACKD is set to the value ’/’.

STARTUP - Program Name for System Variable *STARTUP

STARTUP - Program Name for System Variable *STARTUP

This Natural profile parameter is for UNIX and Windows platforms only.

It specifies a program name for the Natural system variable "*STARTUP ". The program whose name is contained in "*STARTUP" is executed each time the NATURAL command line is invoked. In a Natural program, you can assign another program name to "*STARTUP".

Possible settings	1 - 8 characters	Valid program name.
Default setting	none	
Dynamic specification	NO	
Specification within session	NO	

Note:
If you have Natural Security installed, STARTUP is not evaluated; the startup program to be used is read from the library profile defined in Natural Security.

STEPLIB - Default Steplib Library

This Natural profile parameter is for:

- Mainframes
- UNIX and Windows

This parameter determines the name of the default Natural steplib (concatenated library) to be used.

Possible settings	1 to 8 characters	Steplib name.
Default setting	SYSTEM	
Dynamic specification	YES	
Specification within session	NO	

SUBSID - Subsystem ID under OS/390 and VSE/ESA

This Natural profile parameter is for mainframes and OS/390 and VSE/ESA only.

It identifies the Natural subsystem to be used.

Possible settings	1 to 4 characters	Natural subsystem. If you specify less than 4 characters, blanks will be appended so as to get a 4-byte setting.
Default setting	NAT4	
Dynamic specification	YES	
Specification within session	NO	

For information on the Natural subsystem, see Natural Subsystem under OS/390 or Natural Subsystem under VSE/ESA in the Natural Operations for Mainframes documentation.

SYMGEN - Generate Symbol Table

This Natural profile and session parameter is for UNIX and Windows platforms only.

It specifies whether a symbol table is to be generated or not.

The symbol table contains all symbols used within a Natural program (for example, variable names). It is part of the generated program and is required, for example, for the Natural Debugger and the dialog editor.

Within a session, the profile parameter SYMGEN can be overridden by the session parameter SYMGEN.

Possible settings	ON	A symbol table will be generated.	
	OFF	No symbol table will be generated.	
Default setting	OFF		
Dynamic specification	YES		
Specification within session	YES	Applicable Statements:	SET GLOBALS
		Applicable Command:	GLOBALS

SYNERR - Control of Syntax Errors

This Natural profile parameter is for:

- Mainframes
- UNIX and Windows

It determines whether or not syntax errors will be passed to the error transaction program.

Possible settings	ON	Syntax errors are passed to the error transaction program.
	OFF	Syntax errors are not passed to the error transaction program.
Default setting	OFF	
Dynamic specification	YES	
Specification within session	NO	

The error transaction program is defined either with the profile parameter ETA or within the Natural Security library profile.

SYS - Define and Activate a Set of Dynamic Profile Parameters

This Natural profile parameter is for mainframes only.

It enables you to activate a set of dynamic profile parameters which is predefined in the Natural parameter module. This avoids the repeated specification of long sequences of profile parameters for the Natural session start. Alternatively, a similar functionality is provided by the profile parameter PROFILE.

In the Natural parameter module (NATPARM), you use NTSYS macros to predefine sets of dynamic profile parameters. You identify such a set of parameters by giving it a unique set name.

Possible settings	*set-name*	*set-name* (1 to 8 characters) defined by the NTSYS macro in the Natural parameter module (NATPARM).
Default setting	None	
Dynamic specification	YES	This parameter can only be specified dynamically.
Specification within session	NO	

The specified parameter set must be defined in the Natural parameter module (NATPARM) currently active, e.g. in an alternative parameter module, if it is specified by the PARM parameter before the SYS parameter.

A parameter set is evaluated right in its position of SYS in the parameter string, as you would have included the defined parameter string at this position instead.

The following topics are covered below:

- SYS Parameter Syntax
- NTSYS Macro Syntax
- Example of NTSYS Macro

SYS Parameter Syntax

The parameter syntax of SYS is as follows:

```
SYS=set-name
```

NTSYS Macro Syntax

The NTSYS macro is specified for each set of parameters as follows:

```
NTSYS set-name,'parameter-string1','parameter-string2',...
```


set-name

The *set-name* identifies the subsequent set of parameters, it can be 1 to 8 characters long and must begin with an alphabetical character.

parameter-string

After the *set-name*, you specify individual profile parameters and their values. For the profile parameters you can specify, see Profile Parameters in the Parameter Reference documentation.

- The entire set of parameters you specify with an NTSYS macro must constitute a valid string of dynamic parameters. The specified parameter string is not checked for validity during the NATPARM assembly.
- If *parameter-string1* exceeds 255 characters, you must define a second parameter string *parameter-string2*, etc.
- All parameter strings of one NTSYS macro are concatenated to one set of parameters.
- An apostrophe within a substring is represented by two apostrophes.

Example of NTSYS Macro

```
NTSYS SET1,'FUSER=(,50),LC=ON,NC=ON,ULANG=2,TQ=OFF','STACK=(LOGON ULIB1)'
NTSYS SET2,'FUSER=(,51),ULANG=4,WH=ON,KC=ON,STACK=(LOGON ULIB2)'
```


SYSCIP - Adabas Cipher Key for Natural System Files

This Natural profile parameter is for mainframes and only applies to Adabas databases.

It provides a default Adabas cipher key for access to Natural system files (FNAT, FUSER, FDIC, FSEC, FSPOOL) which have been loaded with the ciphered option.

Possible settings	8 characters	The cipher code specified with the SYSCIP parameter applies to all Natural system files for which no individual cipher codes are specified.
	blanks	If the Natural system files are not ciphered, set SYSCIP to blanks.
Default setting	blanks	
Dynamic specification	YES	If you specify the SYSCIP parameter dynamically in conjunction with any of the individual system file parameters FNAT, FUSER, FDIC, FSEC and FSPOOL, you must specify the SYSCIP parameter **before** any individual system file parameter.
Specification within session	NO	

Note:
Cipher codes for individual system files can be specified with the parameters FNAT, FUSER, FDIC, FSEC and FSPOOL.

SYSPSW - Adabas Password for Natural System Files

This Natural profile parameter is for mainframes and only applies to Adabas databases.

It provides a default Adabas password for access to Natural system files (FNAT, FUSER, FDIC, FSEC, FSPOOL) which have been password-protected.

Possible settings	8 characters	If a Natural system file is password-protected, a password which permits updates to the file must be specified. The password specified with the SYSPSW parameter applies to all Natural system files for which no individual passwords are specified. If the OPRB parameter is specified, the SYSPSW password is used for the initial Adabas open call and must permit access and/or update to all the files specified in OPRB as required.
	blanks	If the Natural system files are not password-protected, set SYSPSW to blanks.
Default setting	blanks	
Dynamic specification	YES	If you specify SYSPSW dynamically in conjunction with any of the individual system file parameters FNAT, FUSER, FDIC, FSEC and FSPOOL, you must specify SYSPSW **before** any individual system file parameter.
Specification within session	NO	

Note:
Passwords for individual system files can be specified with the profile parameters FNAT, FUSER, FDIC, FSEC and FSPOOL.

TAB - Standard Output Character Translation

This Natural profile parameter is for mainframes only.

It allows you to overwrite the definitions in the translation table NTTAB as contained in the configuration module NATCONFG. The NTTAB table is the standard output translation table.

TAB corresponds to the NTTAB macro in the Natural parameter module NATPARM.

Possible settings	See TAB Parameter Syntax below.	
Default setting	As specified within the macro NTTAB in NATCONFG.	
Dynamic specification	YES	This parameter can only be specified dynamically. In the Natural parameter module NATPARM, the macro NTTAB must be used instead.
Specification within session	NO	

The following topics are covered below:

- TAB Parameter Syntax
- NTTAB Macro Syntax
- Example of NTTAB Macro
- Example of TAB Parameter

TAB Parameter Syntax

The TAB parameter is specified as follows:

```
TAB=(a1,a2,b1,b2,c1,c2,...)
```

You specify pairs of characters, the first character of a pair being the character to be translated, the second character of a pair being the character into which the first character is to be translated.

You can specify each character either as the one-byte character itself (enclosed in apostrophes) or as the hexadecimal representation of that character.

NTTAB Syntax

The NTTAB macro is specified as follows:

```
NTTAB a1,a2,b1,b2,c1,c2,...
```


Example of NTTAB Macro

```
NTTAB 5E,'Ä','ö',78,FF,00,'ü','Ü'
```

In this example, the character represented by H'5E' is translated into "Ä", "ö" into the character represented by H'78', the character represented by H'FF' into the character represented by H'00', and "ü" into "Ü".

Example of TAB Parameter

With the TAB parameter, you must enclose the entire string of character pairs in parentheses, for example:

```
TAB=(5E,'Ä','ö',78,FF,00,'ü','Ü')
```


TAB1 - Alternative Output Translation

This Natural profile parameter is for mainframes only.

It allows you to overwrite the definitions in the translation table NTTAB1 as contained in the configuration module NATCONFG. The NTTAB1 table is the alternative output translation table for the secondary character set used when the profile/session parameter PM=C is set.

TAB1 corresponds to the NTTAB1 macro in the Natural parameter module NATPARM.

Possible settings	See TAB1 Parameter Syntax below.	
Default setting	As specified within the macro NTTAB1 in NATCONFG.	
Dynamic specification	YES	This parameter can only be specified dynamically. In the Natural parameter module NATPARM, the macro NTTAB1 must be used instead.
Specification within session	NO	

The following topics are covered below:

- TAB1 Parameter Syntax
- NTTAB1 Macro Syntax
- Example of NTTAB1 Macro
- Example of TAB1 Parameter

TAB1 Parameter Syntax

The TAB1 parameter is specified as follows:

```
TAB1=(a1,a2,b1,b2,c1,c2,...)
```

You specify pairs of characters, the first character of a pair being the character to be translated, the second character of a pair being the character into which the first character is to be translated.

You can specify each character either as the one-byte character itself (enclosed in apostrophes) or as the two-byte hexadecimal representation of that character.

NTTAB1 Macro Syntax

The NTTAB1 macro is specified as follows:

```
NTTAB1 a1,a2,b1,b2,c1,c2,...
```


Example of NTTAB1 Macro

```
NTTAB1 5E,'Ä','ö',78,FF,00,'ü','Ü'
```

In this example, the character represented by H'5E' is translated into "Ä", "ö" into the character represented by H'78', the character represented by H'FF' into the character represented by H'00', and "ü" into "Ü".

Example of TAB1 Parameter

With the TAB1 parameter, you must enclose the entire string of character pairs in parentheses, for example:

```
TAB1=(5E,'Ä','ö',78,FF,00,'ü','Ü')
```


TAB2 - Alternative Input Translation

This Natural profile parameter is for mainframes only.

It allows you to overwrite the definitions in the translation table NTTAB2 as contained in the configuration module NATCONFG. The NTTAB2 table is the alternate input translation table for the secondary character set used when the profile/session parameter PM is set to PM=C.

TAB2 corresponds to the NTTAB2 macro in the Natural parameter module NATPARM.

Possible settings	See TAB2 Parameter Syntax below.	
Default setting	As specified within the macro NTTAB2 in NATCONFG.	
Dynamic specification	YES	This parameter can only be specified dynamically. In the Natural parameter module NATPARM, the macro NTTAB2 must be used instead.
Specification within session	NO	

The following topics are covered below:

- TAB2 Parameter Syntax
- NTTAB2 Macro Syntax
- Example of NTTAB2 Macro
- Example of TAB2 Parameter

TAB2 Parameter Syntax

The TAB2 parameter is specified as follows:

```
TAB2=(a1,a2,b1,b2,c1,c2,...)
```

You specify pairs of characters, the first character of a pair being the character to be translated, the second character of a pair being the character into which the first character is to be translated.

You can specify each character either as the one-byte character itself (enclosed in apostrophes) or as the two-byte hexadecimal representation of that character.

NTTAB2 Macro Syntax

The NTTAB2 macro is specified as follows:

```
NTTAB2 a1,a2,b1,b2,c1,c2,...
```


Example of NTTAB2 Macro

```
NTTAB2 5E,'Ä','ö',78,FF,00,'ü','Ü'
```

In this example, the character represented by H'5E' is translated into "Ä", "ö" into the character represented by H'78', the character represented by H'FF' into the character represented by H'00', and "ü" into "Ü".

Example of TAB2 Parameter

With the TAB2 parameter, you must enclose the entire string of character pairs in parentheses, for example:

```
TAB1=(5E,'Ä','ö',78,FF,00,'ü','Ü')
```


TABA1 - EBCDIC-to-ASCII Translation

This Natural profile parameter is for mainframes only.

It allows you to overwrite the definitions in the translation table NTTABA1 as contained in the configuration module NATCONFG. This table is used for EBCDIC-to-ASCII translation.

TABA1 corresponds to the NTTABA1 macro in the Natural parameter module NATPARM.

Possible settings	See TABA1 Parameter Syntax below.	
Default setting	As specified within the macro NTTABA1 in NATCONFG.	
Dynamic specification	YES	This parameter can only be specified dynamically. In the Natural parameter module NATPARM, the macro NTTABA1 must be used instead.
Specification within session	NO	

The following topics are covered below:

- TABA1 Parameter Syntax
- NTTABA1 Macro Syntax
- Example of NTTABA1 Macro
- Example of TABA1 Parameter

TABA1 Parameter Syntax

The TABA1 parameter is specified as follows:

```
TABA1=(a1,a2,b1,b2,c1,c2,...)
```

You specify pairs of characters, the first character of a pair being an EBCDIC character to be translated, the second character of a pair being the ASCII character into which the EBCDIC character is to be translated.

You can specify each character either as the one-byte character itself (enclosed in apostrophes) or as the two-byte hexadecimal representation of that character.

NTTABA1 Macro Syntax

The NTTABA1 macro is specified as follows:

```
NTTABA1 a1,a2,b1,b2,c1,c2,...
```


Example of NTTABA1 Macro

```
NTTABA1 5E,'Ä','ö',78,FF,00,'ü','Ü'
```

In this example, the character represented by H'5E' is translated into "Ä", "ö" into the character represented by H'78', the character represented by H'FF' into the character represented by H'00', and "ü" into "Ü".

Example of TABA1 Parameter

With the TABA1 parameter, you must enclose the entire string of character pairs in parentheses, for example:

```
TABA1=(5E,'Ä','ö',78,FF,00,'ü','Ü')
```


TABA2 - ASCII-to-EBCDIC Translation

This Natural profile parameter is for mainframes only.

It allows you to overwrite the definitions in the translation table NTTABA2 as contained in the configuration module NATCONFG. This table is used for ASCII-to-EBCDIC translation.

TABA2 corresponds to the NTTABA2 macro in the Natural parameter module NATPARM.

Possible settings	See TABA2 Parameter Syntax below.	
Default setting	As specified within the macro NTTABA2 in NATCONFG.	
Dynamic specification	YES	This parameter can only be specified dynamically. In the Natural parameter module NATPARM, the macro NTTABA2 must be used instead.
Specification within session	NO	

The following topics are covered below:

- TABA2 Parameter Syntax
- NTTABA2 Macro Syntax
- Example of NTTABA2 Macro
- Example of TABA2 Parameter

TABA2 Parameter Syntax

The TABA2 parameter is specified as follows:

```
TABA2=(a1,a2,b1,b2,c1,c2,...)
```

You specify pairs of characters, the first character of a pair being an ASCII character to be translated, the second character of a pair being the EBCDIC character into which the ASCII character is to be translated.

You can specify each character either as the one-byte character itself (enclosed in apostrophes) or as the two-byte hexadecimal representation of that character.

NTTABA2 Macro Syntax

The NTTABA2 macro is specified as follows:

```
NTTABA2 a1,a2,b1,b2,c1,c2,...
```


Example of NTTABA2 Macro

```
NTTABA2 5E,'Ä','ö',78,FF,00,'ü','Ü'
```

In this example, the character represented by H'5E' is translated into "Ä", "ö" into the character represented by H'78', the character represented by H'FF' into the character represented by H'00', and "ü" into "Ü".

Example of TABA2 Parameter

With the TABA2 parameter, you must enclose the entire string of character pairs in parentheses, for example:

```
TABA2=(5E,'Ä','ö',78,FF,00,'ü','Ü')
```


TABL - SYS Library Output Translation

This Natural profile parameter is for mainframes only.

It allows you to overwrite the definitions in the translation table NTTABL as contained in the configuration module NATCONFG. The NTTABL table is used to translate output produced by programs located in "SYS" libraries.

TABL corresponds to the NTTABL macro in the Natural parameter module NATPARM.

Possible settings	See TABL Parameter Syntax below.	
Default setting	As specified within the macro NTTABL in NATCONFG.	
Dynamic specification	YES	This parameter can only be specified dynamically. In the Natural parameter module NATPARM, the macro NTTABL must be used instead.
Specification within session	NO	

The following topics are covered below:

- TABL Parameter Syntax
- NTTABL Macro Syntax
- Example of NTTABL Macro
- Example of TABL Parameter

TABL Parameter Syntax

The TABL parameter is specified as follows:

```
TABL=(a1,a2,b1,b2,c1,c2,...)
```

You specify pairs of characters, the first character of a pair being the character to be translated, the second character of a pair being the character into which the first character is to be translated.

You can specify each character either as the one-byte character itself (enclosed in apostrophes) or as the two-byte hexadecimal representation of that character.

NTTABL Macro Syntax

The NTTABL macro is specified as follows:

```
NTTABL a1,a2,b1,b2,c1,c2,...
```


Example of NTTABL Macro

```
NTTABL 5E,'Ä','ö',78,FF,00,'ü','Ü'
```

In this example, the character represented by H'5E' is translated into "Ä", "ö" into the character represented by H'78', the character represented by H'FF' into the character represented by H'00', and "ü" into "Ü".

Example of TABL Parameter

With the TABL parameter, you must enclose the entire string of character pairs in parentheses, for example:

```
TABL=(5E,'Ä','ö',78,FF,00,'ü','Ü')
```


TD - Time Differential

This Natural profile parameter is for:

- Mainframes
- UNIX and Windows

It indicates a time differential to be applied to the Natural time/date setting to ensure that the current local time/date is used, rather than the computer center time/date.This parameter is applicable in an environment in which remote nodes are being used in a computer network.

Possible settings	AUTO	Natural compares the physical (store clock) and logical (system environment) machine times and uses the difference between the two as the setting for the TD parameter. For a time change to take effect for Natural (for example, to change time to summer time or back to winter time), it is therefore sufficient to reset the logical machine time.
	+/-hh (+/-hh,mm) (+/-hh,mm,ss)	Hours, minutes and seconds from (-23,59,59) to (+23,59,59). A plus (optional) or minus sign indicates, whether the TD value is to be added or subtracted. The specified time is added to / subtracted from the physical machine time to set the time/date to be used by Natural.
	1 to 32 characters	Name of the time zone to be used. This must be defined as a valid time zone in the NTTZ macro of the NATCONFG module, see Configuration Tables - Module NATCONFG.
Default setting	0	
Dynamic specification	YES	
Specification within session	NO	

Examples:

```
TD=6           (6 hours ahead)
TD=(5,30)      (5 hours and 30 minutes ahead)
TD=(-6,12,30)  (6 hours, 12 minutes and 30 seconds behind)
```

VSE/ESA-Specific Information:
With VSE-type operating systems, "// ZONE" and "//DATE" JCL statements are honored with TD=AUTO. This can also affect the setting of the profile parameter DD. See also the profile parameters YD and DD.

TF - Translation of Database ID/File Number

This Natural profile parameter is for:

- Mainframes
- UNIX and Windows

The parameter TF applies to user files only, **not** to system files.

This parameter can be used to translate a database ID/file number to another database ID/file number during the execution of an application. It corresponds to the macro NTTF in the parameter module NATPARM.

Possible settings	*production-DBID*	Must be in the range of 0 to 254, or 256 to 65535, or can be "*" which stands for all DBIDs.
	production-FNR	Must be in the range of 1 to 65535, or "*" which stands for all FNRs.
	test-DBID	Must be in the range of 0 to 254, or 256 to 65535, or can be "*" which leaves the DBID unchanged.
	test-FNR	Must be in the range of 1 to 65535, or "*" which leaves the DBID unchanged.
Default setting	None	
Dynamic specification	YES	This parameter can only be specified dynamically. In the Natural parameter module NATPARM, the macro NTTF must be used instead.
Specification within session	NO	

This feature is relevant when developing an application in a production environment. It enables you to develop an application in a test database and then transfer the finished application to the production database without having to change or re-compile the application.The Natural objects are cataloged with the production DBID/FNR, but whenever a database access is executed, the production DBID/FNR is translated into the test DBID/FNR according to the TF parameter specifications; that is, the test database is used. This means that testing can take place in the actual production environment, but not with production data.

The TF parameter or the NTTF macro can be specified several times so as to specify different combinations of file numbers.

The following topics are covered below:

- TF Parameter Syntax
- NTTF Macro Syntax
- Example of TF Parameter
- Example of NTTF Macro

TF Parameter Syntax

The TF profile parameter is specified as follows:

```
TF=(production-DBID,production-FNR,test-DBID,test-FNR)
```

NTTF Macro Syntax

The NTTF macro is specified as follows:

```
NTTF production-DBID,production-FNR,test-DBID,test-FNR
```

Example of TF Parameter

```
TF=(777,39,17,88),TF=(251,*,9,*)
```

Example of NTTF Macro

Equivalent specification in the Natural parameter module:

```
NTTF 777,39,17,88
NTTF 251,*,9,*
```


THSEP - Dynamic Thousands Separator

This Natural profile parameter is for UNIX and Windows only. It can only be set with the Natural Configuration Utility (Windows) or the NATPARM Utility (UNIX).

With THSEP, you can enable or disable the use of thousands separators at compilation time.

Note:
In the Natural source, the thousands character can be specified by either a comma (’,’) or a period (’.’) depending on the current setting of the profile and session parameter DC (decimal character). If DC=’,’, then the thousands separator character must be specified as a period in the edit mask, otherwise it must be specified as a comma.

Possible settings	ON	Thousands separator used. Every thousands separator character that is not part of a string literal is replaced internally with a control character.
	OFF	Thousands separator not used, i.e. no thousands separator control character is generated by compiler. This is the compatibility setting.
Default setting	OFF	
Dynamic specification	NO	
Specification within session	NO	

See also profile parameter THSEPCH.

THSEPCH - Thousands Separator

This Natural profile parameter is for UNIX and Windows only. It can only be set with the Natural Configuration Utility (Windows) or the NATPARM Utility (UNIX).

With THSEPCH, you specify the character to be used as a thousands separator at runtime.

Note:
In the Natural source, the thousands character is always specified by a comma (’,’) or a period (’.’), independently of the THSEPCH setting. See the profile parameter THSEP for further information.

Possible settings	any character	At runtime, the control character is replaced with this character.
Default setting	,	By default, a comma is used as control character.
Dynamic specification	NO	
Specification within session	NO	

Examples:

A Natural program that is developed with DC=’.’ and cataloged with THSEP=ON uses the edit mask (EM=ZZ,ZZZ,ZZ9.99).
The numeric value will be displayed as 1,234,567.89.

With DC=’,’ and THSEPCH=’.’, the value will be displayed as 1.234.567,89.

TIMEOUT - Wait Time for RPC Server Response

For static specification, this parameter is available as a keyword subparameter of the NTRPC macro. For dynamic specification, this parameter is available as a subparameter of the profile parameter RPC.

TIMEOUT specifies the number of seconds the client is to wait for an RPC server response. If this time is exceeded, the remote procedure call will be terminated with a corresponding error message.

TIMEOUT is specified on the client side only.

Possible settings	0 - 32767	Seconds.
Default setting	55	
Dynamic specification	YES	
Specification within session	YES	At runtime, this value can be overwritten using the Parameter Maintenance function of the SYSRPC utility.

For further information, see the Natural Remote Procedure Call documentation.

TMODEL - IBM 3270 Terminal Model

This Natural profile parameter is for IBM mainframes only.

Under CICS, this parameter is ignored, because the terminal screen size is defined by the CICS terminal control table.

TMODEL controls the IBM 3270 terminal model number for online environments, e.g. under IMS/DC. It is used to determine the number of lines and columns of the terminal screen.

Possible settings	0	The screen size is determined by the environment-dependant driver module. If possible, it gets the screen size information from its subsystem. Otherwise, the definitions of default model 2 are used, e.g. under IMS/DC.
	2	The screen size is 24 lines and 80 columns.
	3	The screen size is 32 lines and 80 columns.
	4	The screen size is 43 lines and 80 columns.
	5	The screen size is 27 lines and 132 columns.
Default setting	0	
Dynamic specification	YES	
Specification within session	NO	

Note:
If your TMODEL specification is incompatible with the physical terminal screen size, the output data may be displayed incorrectly or hardware errors may occur.

TMPSORTUNIQ - Alternate Algorithm for Generating Sort Work File Names

TMPSORTUNIQ - Alternate Algorithm for Generating Sort Work File Names

This Natural profile parameter is for UNIX and Windows platforms only.

If the parameter is specified on the Natural startup command, Natural will use a different algorithm from normal when generating work file names for sort operations. It forces Natural to generate a unique file name for any required work files without embedding the values of the ETID parameter and user ID into the file name.

The normal, default algorithm creates work file names with the values of the ETID setting and the user ID embedded. Whereas this algorithm usually results in file names which are unique to a particular Natural session, this may cause problems in environments where the ETID contains characters which are invalid within a file name, or where multiple Natural sessions are running which use the same user ID and no ETID specification (thus possibly resulting in work file names for sort operations being created which are not unique).

Possible settings	specified or not specified	If specified, Natural will use a different algorithm from normal when generating work file names for sort operations.
Default setting	not specified	
Dynamic specification	YES	
Specification within session	NO	

TPF (Internal Use)

This Natural profile parameter is for mainframes only.

This parameter is reserved for internal use by Natural. Do not change its setting.

TQ - Translate Quotation Marks

This Natural profile parameter is basically for UNIX and Windows platforms. On mainframes, use the keyword subparameter TQMARK of the profile parameter CMPO.

The profile parameter TQ controls the translation of a quotation mark (") within a Natural text constant. It takes effect at compilation time only.

Possible settings	ON	Each quotation mark within a text constant is output as a single apostrophe.
	OFF	Quotation marks within text constants are not translated, they are output as quotation marks.
Default setting	ON	
Dynamic specification	YES	
Specification within session	NO	

Note:
Do not confuse quotation mark (") with double apostrophes (’’). Double apostrophes within a text constant are always output as a single apostrophe (’), regardless of the setting of the TQ parameter.

Example 1 (TQ=ON):

```
WRITE ’THERE’S A QUOTATION MARK’
```

is displayed as: THERE’S A QUOTATION MARK

Example 2 (TQ=OFF):

```
WRITE ’THERE"S A QUOTATION MARK’
```

is displayed as: THERE"S A QUOTATION MARK

Note:
Do not confuse quotation mark (") with double apostrophes (’’). Double apostrophes within a text constant are always output as a single apostrophe (’), regardless of the setting of the TQ parameter.

Example 3 (TQ=ON or OFF):

```
WRITE ’DOUBLE APOSTROPHES’’ OUTPUT IS A SINGLE APOSTROPHE’

’DOUBLE APOSTROPHES’ OUTPUT IS A SINGLE APOSTROPHE’
```


TRACE - Define Trace Level for Natural RPC Servers

For static specification, this parameter is available as a keyword subparameter of the NTRPC macro. For dynamic specification, this parameter is available as a subparameter of the profile parameter RPC.

TRACE activates the RPC trace facility and determines the trace level *n* to be used. For further information, see Using the Server Trace Facility p.p. in the Natural Remote Procedure Call documentation.

Possible settings	0	Nothing is traced.
	1	Only messages (inclusive Natural errors) are traced.
	2	All messages and data from/to client are traced.
Default setting	0	
Dynamic specification	YES	
Specification within session	NO	

The values 3-9 displayed in the selection box are for future use. If one of these values is chosen, then TRACE=2 is selected.

TRACE - Define Components to be Traced

This Natural profile parameter is for mainframes only.

It is intended primarily for Software AG internal use for debugging purposes. It can be used to define the components for which trace data are to be written. It does not activate trace recording.

Trace recording can be activated by the profile parameters ITRACE (internal trace) and ETRACE (external trace) or during the session by the corresponding terminal commands %TRI and %TRE.

Do not use this parameter without prior consultation of Software AG Support.

TRACE corresponds to the macro NTTRACE in the Natural parameter module NATPARM.

Possible settings	List of *trace-ID*s	*trace-ID*s (each 1-8 bytes), define the names of the Natural components to be traced.
Default setting	None	
Dynamic specification	YES	This parameter can only be specified dynamically. In the Natural parameter module NATPARM, the macro NTTRACE must be used instead.
Specification within session	NO	

The setting lists of multiple TRACE parameter specifications are not concatenated; that is, a TRACE parameter overrides any previously specified TRACE parameter and any NTTRACE macro definitions.

The following topics are covered below:

- TRACE Parameter Syntax
- NTTRACE Macro Syntax
- Example of TRACE Parameter
- Example of NTTRACE Macro

TRACE Parameter Syntax

The TRACE parameter is specified as follows:

```
TRACE=( trace-ID1,trace-ID2,...)
```

NTTRACE Macro Syntax

The NTTRACE macro is specified as follows:

```
NTTRACE trace-ID1,trace-ID2,...
```

Multiple specifications of the NTTRACE macro are concatenated to one trace list.

Example of TRACE Parameter

```
TRACE=(NATGETM,NATFREM,DYNPARMS)
```

This defines traces to be written for the Natural nucleus components "storage aquisition", "storage release" and "dynamic parameter evaluation".

Example of NTTRACE Macro

Equivalent specification in the Natural parameter module:

```
NTTRACE NATGETM,NATFREM,DYNPARMS
```


TRANSP - Server Transport Protocol

For static specification, this parameter is available as a keyword subparameter of the NTRPC macro. For dynamic specification, this parameter is available as a subparameter of the profile parameter RPC.

TRANSP determines which server transport protocol is used. If ACI is used, you can additionally specify the transport method.

It is specified on the server side only.

Possible settings	ACI	ACI is used. The transport method is defined by the EntireX Broker.
	ACI,TCP	ACI is used with TCP/IP.
	ACI,NET	ACI is used with Entire Net-work, i.e. using the Adabas protocol.
	ACI,TCP-NET	Trying to use ACI with TCP. If not available, ACI is used with NET.
	ACI,NET-TCP	Trying to use ACI with NET. If not available, ACI is used with TCP.
Default setting	ACI	
Dynamic specification	YES	
Specification within session	NO	

The use of TRANSP is no longer required as you may now specify the full node name with SRVNODE. It is still supported for compatibility reasons.

For further information, see the Natural Remote Procedure Call documentation.

TRYALT - Try Alternative Server Address

For static specification, this parameter is available as a keyword subparameter of the NTRPC macro. For dynamic specification, this parameter is available as a subparameter of the profile parameter RPC.

TRYALT determines whether an RPC client should try to execute an RPC request on an alternative server (ON) or not (OFF). For further information, see Specifying RPC Server Adresses in the Natural Remote Procedure Call documentation.

TRYALT is specified on the client side only.

Possible settings	ON	If a request could not be executed on the node you specified, the RPC client tries to find an alternative server address to send that request to.
	OFF	No such attempt will be made.
Default setting	OFF	
Dynamic specification	YES	
Specification within session	YES	At runtime, this value can be overwritten using the Parameter Maintenance function of the SYSRPC utility.

For further information, see the Natural Remote Procedure Call documentation.

TS - Translate Output from Programs in System Libraries

This Natural profile and session parameter is for mainframes only.

The TS parameter applies only to primary output (CMPRINT).

It is used to translate output from Natural system libraries (that is, libraries whose names begin with "SYS") using a translation table. This may be necessary for locations which have non-standard lower-case usage (for example, Middle East countries).

Error messages or warnings are only translated if the English version of the text is displayed and the Natural session is not running with English language code (ULANG=1). If the text is displayed in the local language (for example, Hebrew), it is not translated into upper-case characters. The translation of messages and warnings does not depend on the library from where the program is executed.

Within a Natural session, the profile parameter TS can be overridden by the session parameter TS.

Possible settings	ON	Output is translated. With TS=ON, the profile parameter LC=OFF and the session parameter AD=T, both of which translate input to upper case, are ignored, as they would cause undesired character translation for special character sets.	
	OFF	Output is not translated.	
Default setting	OFF		
Dynamic specification	YES		
Specification within session	YES	Applicable Statements:	SET GLOBALS
		Applicable Command:	GLOBALS

For details on session parameter specification at statement or element level and evaluation at compilation time or at runtime, refer to Session Parameter Specification/Evaluation Overview.

Note:
The translation table can be modified with the NTTABL macro or the corresponding dynamic profile parameter TABL.

TSIZE - Size of Buffer for Adabas Text Retrieval

This Natural profile parameter is for mainframes only.

It determines the size of the buffer to be used for the Adabas Text Retrieval facility.

Alternatively, you can use the equivalent Natural profile parameter DS or macro NTDS (see Using Optional Macros in a Natural Parameter Module in the Natural Operations for Mainframes documentation) to specify the TSIZE value.

Possible settings	1 - 128	Buffer size in KB. If the requested space is not available, the Adabas Text Retrieval facility cannot be used.
	0	Adabas Text Retrieval facility is not used.
Default setting	0	
Dynamic specification	YES	
Specification within session	NO	

TTYPE - Terminal Type

This Natural profile parameter is for mainframes only.

It allows you to specify the terminal type used - in TP environments in which this information is not supplied automatically - so that Natural can activate the appropriate converter routine for attribute sequences to operate that type of terminal.

Possible settings	1-4 characters	The setting specified with the TTYPE parameter must be defined as a valid terminal device type in the NTDVCE macro of the NATCONFG module, see Configuration Tables - Module NATCONFG.
Default setting	IBM	3270
	Siemens	The setting defined in PDN, unless overridden by the parameter T975X (see Natural TP Monitor Interfaces, Natural under TIAM, Parameters in Macro NAMTIAM).
Dynamic specification	YES	
Specification within session	YES	The TTYPE parameter has the same function as the terminal command %T=.

Note:
If you use the TTYPE parameter, it is no longer necessary to execute a program containing a SET CONTROL 'T=...' statement at the start of the session in order to set the terminal type.

UDB - User Database ID

This Natural profile parameter is for:

- Mainframes
- UNIX and Windows

It specifies the DBID to be used for a database access.

Possible settings	0 - 65535, except 255	Valid database ID. Database ID 255 is reserved for logical system files for Software AG products, see profile parameter LFILE.)
Default setting	database ID applicable for FUSER	
Dynamic specification	YES	
Specification within session	NO	

Notes:

- The DBID 0 and the databases selected with the UDB parameter must be of the same type (Adabas/Adabas, SQL/SQL or XML/XML for example).
- If no DBID is specified in the DDM used, the DBID specified with the UDB profile parameter determines which database is accessed. Thus it is possible to have different user environments without multiple FUSER files being required.
- If no DBID is specified in the DDM and the UDB profile parameter is not specified, the DBID that applies to the FUSER system file is used.

ULANG - User Language

This Natural profile parameter is for:

- Mainframes
- UNIX and Windows

It indicates the language to be used for date edit masks, system messages, user messages, helptexts, helproutines, and multi-lingual maps. The setting is used to set the Natural system variable "*LANGUAGE".

Possible settings	1 - 60	Language code. For example, 1 is assigned to English, 2 is assigned to German, 3 is assigned to French. For a detailed list of language codes, see the table in the documentation of the *LANGUAGE variable.
Default setting	1	
Dynamic specification	YES	
Specification within session	NO	

For additional information about language indicators and possible settings, see Configuration Tables - Module NATCONFG in the Natural Operations for Mainframes documentation.

MF

UPSI - VSE User Program Switches

This Natural profile parameter is for mainframes only and is used for debugging in Natural under VSE/ESA.

See also Debugging Facilities for Natural under VSE/ESA in the Natural Operations for Mainframes documentation.

It specifies UPSI settings for the Natural VSE Interface corresponding to the VSE/ESA UPSI system control statement. The UPSI profile parameter is in particular relevant in cases where UPSI system control statement settings have produced side effects in the sense that they have a different meaning for other programs such as for front-end Natural or for programs called by Natural.

Possible settings	1-8 characters	Any combination of the characters 0, 1, X.
Default setting	XXXXXXXX	
Dynamic specification	YES	
Specification within session	NO	

The syntax for the UPSI string is the same as for the VSE/ESA UPSI system control statement.

The Natural VSE/ESA batch interface takes the UPSI settings in JCL and merges the UPSI profile settings into it according to the following rules:

0	The corresponding bit is 0.
1	The corresponding bit is 1.
X	The corresponding bit remains unchanged.

USEDIC - Common Logical Name for Dictionary Servers

This Natural profile parameter is for UNIX and Windows platforms only.

It is used to specify a common logical name for dictionary servers defined with Natural RPC to enable remote dictionary access on a mainframe or UNIX host.

Possible settings	any valid server name	Remote dictionary access will be possible.
Default setting	blank	Remote dictionary access will not be possible.
Dynamic specification	NO	
Specification within session	NO	

USER - Restrict Use of Profile Parameter Strings and Modules

This Natural profile parameter is for:

- Mainframes
- UNIX and Windows

The profile parameter USER can be used to restrict the use of dynamic parameter strings as specified in a SYSPARM profile, NTSYS macro or parameter dataset (CMPRMIN) or to restrict an alternative parameter module (NATPARM).

Possible settings	List of user IDs	The IDs of the users who will be allowed to use the subsequently specified string of profile parameters. Only the specified users will then be allowed to use that parameter string.
Default setting	None	
Dynamic specification	YES	This parameter can only be specified dynamically. To restrict the use of an alternative parameter module (NATPARM), the corresponding macro NTUSER must be used instead.
Specification within session	NO	

The USER parameter applies only to the string of dynamic parameters specified **after** it. The NTUSER macro applies to the parameter module in which it is specified. The default Natural parameter module linked to the environment-dependent Natural nucleus cannot be restricted.

When the dynamic profile parameters are evaluated and the USER parameter is encountered, Natural checks if the current user ID (that is, the current setting of the system variable *INIT-USER) is contained in the list of user IDs specified with the USER parameter. If it is not, the user receives a corresponding error message (NAT7017), and the processing of dynamic profile parameters is terminated immediately.

When an alternative parameter module is to be used, Natural loads the alternative parameter module specified by the PARM parameter and checks if the current user ID (that is, the current setting of the system variable *INIT-USER) is contained in the list of user IDs specified by the NTUSER macro in the alternative parameter module. If it is not, the user receives a corresponding error message (NAT7014), and the alternative parameter module is discarded.

To restrict the use of:

- a SYSPARM profile,
  you specify the USER parameter as the first parameter in the profile. The subsequent string of profile parameters in the profile, that is, the entire profile, can then only be used by the user specified with the USER parameter.
- a parameter string defined by an NTSYS macro or in a CMPRMIN dataset,
  you specify the USER parameter as the first parameter in the parameter string.
- an alternative parameter module,
  you specify the macro NTUSER in the alternative parameter module.

The following topics are covered below:

USER Parameter Syntax

The parameter syntax of USER is as follows:

```
USER=(user-id1,user-id2,...)
```

NTUSER Macro Syntax

The NTUSER macro is specified in a Natural parameter module as follows:

```
NTUSER user-id1,user-id2,user-id3,...
NTUSER user-id4,user-id5,...
...
```

Example of NTUSER Macro

The following is an example of protecting a Natural parameter macro:

```
NTPRM ...
...
NTUSER ADMIN1,ADMIN2
```

Example of USER Parameter

```
USER=(ADMIN1,ADMIN2),FNAT=(12,177,SECPASSW,74832055)
```


USERBUF (Internal Use)

This Natural profile parameter is for mainframes only.

This parameter is reserved for internal use by Natural. Do not change its setting.

USEREP - Repository Usage

This Natural profile parameter is for Windows platforms only.

It enables you to use the repository.

Possible settings	ON	Repository usage enabled.
	OFF	Repository usage disabled.
Default setting	ON	
Dynamic specification	YES	
Specification within session	NO	

USIZE - Size of User Buffer

USIZE - Size of User Buffer

This Natural profile parameter is for UNIX and Windows platforms only.

It defines the size of the user buffer in virtual memory. The user buffer contains all data dynamically allocated by Natural.

Possible settings	10 - 1024	Buffer size in MB.
	0	With USIZE=0, the memory capacity will be unrestricted.
Default setting	10	
Dynamic specification	NO	
Specification within session	NO	

UTAB1 - Lower-to-Upper-Case Translation

This Natural profile parameter is for mainframes only.

It allows you to overwrite the definitions in the translation table NTUTAB1 as contained in the configuration module NATCONFG. The NTUTAB1 table is used for lower-to-upper-case translation.

UTAB1 corresponds to the NTUTAB1 macro in the Natural parameter module NATPARM.

Possible settings	See UTAB1 Parameter Syntax below.	
Default setting	As specified within the macro NTUTAB1 in NATCONFG.	
Dynamic specification	YES	This parameter can only be specified dynamically. In the Natural parameter module NATPARM, the macro NTUTAB1 must be used instead.
Specification within session	NO	

The following topics are covered below:

- UTAB1 Parameter Syntax
- NTUTAB1 Macro Syntax
- Example of NTUTAB1 Macro
- Example of UTAB1 Parameter

UTAB1 Parameter Syntax

The UTAB1 parameter is specified as follows:

```
UTAB1=(a1,a2,b1,b2,c1,c2,...)
```

You specify pairs of characters, the first character of a pair being a lower-case character to be translated, the second character of a pair being the upper-case character into which the lower-case character is to be translated.

You can specify each character either as the one-byte character itself (enclosed in apostrophes) or as the two-byte hexadecimal representation of that character.

NTUTAB1 Macro Syntax

The NTUTAB1 macro is specified as follows:

```
NTUTAB1 a1,a2,b1,b2,c1,c2,...
```

Example of NTUTAB1 Macro

```
NTUTAB1 5E,'Ä','ö',78,FF,00,'ü','Ü'
```


In this example, the character represented by H’5E’ is translated into "Ä", "ö" into the character represented by H’78’, the character represented by H’FF’ into the character represented by H’00’, and "ü" into "Ü".

Example of UTAB1 Parameter

With the UTAB1 parameter, you must enclose the entire string of character pairs in parentheses, for example:

```
UTAB1=(5E,’Ä’,’ö’,78,FF,00,’ü’,’Ü’)
```


UTAB2 - Upper-to-Lower-Case Translation

This Natural profile parameter is for mainframes only.

It allows you to overwrite the definitions in the translation table NTUTAB2 as contained in the configuration module NATCONFG. The NTUTAB2 table is used for upper-to-lower case translation.

UTAB2 corresponds to the NTUTAB2 macro in the Natural parameter module NATPARM.

Possible settings	See UTAB2 Parameter Syntax below.	
Default setting	As specified within the macro NTUTAB2 in NATCONFG.	
Dynamic specification	YES	This parameter can only be specified dynamically.In the Natural parameter module NATPARM, the macro NTUTAB2 must be used instead.
Specification within session	NO	

The following topics are covered below:

- UTAB2 Parameter Syntax
- NTUTAB2 Macro Syntax
- Example of NTUTAB2 Macro
- Example of UTAB2 Parameter

UTAB2 Parameter Syntax

The UTAB2 parameter is specified as follows:

```
UTAB2=(a1,a2,b1,b2,c1,c2,...)
```

You specify pairs of characters, the first character of a pair being a upper-case character to be translated, the second character of a pair being the lower-case character into which the upper-case character is to be translated.

You can specify each character either as the one-byte character itself (enclosed in apostrophes) or as the two-byte hexadecimal representation of that character.

NTUTAB2 Macro Syntax

The NTUTAB2 macro is specified as follows:

```
NTUTAB2 a1,a2,b1,b2,c1,c2,...
```

Example of NTUTAB2 Macro

```
NTUTAB2 5E,'Ä','ö',78,FF,00,'ü','Ü'
```


In this example, the character represented by H’5E’ is translated into "Ä", "ö" into the character represented by H’78’, the character represented by H’FF’ into the character represented by H’00’, and "ü" into "Ü".

Example of UTAB2 Parameter

With the UTAB2 parameter, you must enclose the entire string of character pairs in parentheses, for example:

```
UTAB1=(5E,’Ä’,’ö’,78,FF,00,’ü’,’Ü’)
```


VSIZE - Size of Buffer Area for Natural/VSAM

This Natural profile parameter is for mainframes and applies only if the Natural VSAM interface is installed.

It sets the maximum size of the buffer area required by Natural for VSAM. If set to 0 or if the requested space is not available, the Natural for VSAM Interface cannot be used.

Possible settings	1 - 512	Buffer size in KB. The size actually required depends on the specifications in the NVSPARM macro (described in the Natural for VSAM documentation). If the requested space is not available, the Natural VSAM interface cannot be used. An appropriate error message at the initialization of the Natural VSAM interface tells you which buffer specified in NVSPARM does not fit into the VSIZE area; you can then either reduce individual buffer sizes in NVSPARM or increase the size of the VSIZE area.
	0	With VSIZE=0, the Natural VSAM interface cannot be used.
Default setting	0	
Dynamic specification	YES	
Specification within session	NO	

Note:
If Natural is installed for VSAM, the corresponding Natural buffers are requested at the initialization of the Natural session. If you do not need VSAM support during a Natural session, it is recommended that you invoke Natural with VSIZE=0 to avoid overhead caused by handling of unused buffers.

WFOPFA - Opening of Work Files

This Natural profile parameter is for UNIX and Windows platforms only.

It specifies when work files are to be opened by Natural.

Possible settings	ON	A work file is opened at the time when it is first accessed by a given READ WORK FILE or WRITE WORK FILE statement. This means that only those work files which are actually accessed are opened, while the contents of unopened work files are not deleted.
	OFF	All work files referenced in a Natural object are opened when the execution of this object starts. (This may delete the content of a work file when closing if the work file was referenced by a WRITE WORK FILE statement that was never executed.)
Default setting	ON	
Dynamic specification	YES	
Specification within session	NO	

WFOPFA=OFF only affects main programs; for routines, WFOPFA=ON always applies.

WH - Wait for Record in Hold Status

This Natural profile and session parameter is for all platforms. It applies to Adabas databases only.

It specifies the action to be taken if a required record is not available for processing, because it has been placed in hold status by another user.

Within a Natural session, the profile parameter WH can be overridden by the session parameter WH.

Possible settings	ON	The user is placed in wait status until either the requested record becomes available, or an error message is issued due to Adabas exceeding a time limit or other limit while attempting to place the record in hold status.	
	OFF	An error message is returned if any of these records cannot be placed in hold status.	
Default setting	OFF		
Dynamic specification	YES		
Specification within session	YES	Applicable Statements:	SET GLOBALS
		Applicable Command:	GLOBALS

For details on session parameter specification at statement or element level and evaluation at compilation time or at runtime, refer to Session Parameter Specification/Evaluation Overview.

Note:
When a Natural statement is executed which results in Adabas records being read and an update/delete operation could follow, Natural requests that Adabas places these records in hold status. See the Adabas Command Reference documentation for further information on hold processing.

WORK - Work-File Assignments

This Natural profile parameter is for all platforms, however, the settings are different on mainframes and under UNIX/Windows.

This document refers to Natural for Mainframes, for other platform-specific information refer to the Natural for UNIX or Windows documentation.

The profile parameter WORK allows you to define the work files to be used during the session. Within a session, up to 32 logical work files (numbered 1 to 32) can be used.

WORK corresponds to the NTWORK macro in the parameter module NATPARM. To provide different work file definitions, WORK or NTWORK can be specified multiple times.

Possible settings		See Keyword Subparameters below.
Default setting	See below.	Depending on the access method and the environment, there may be different default settings.
Dynamic specification	YES	The parameter WORK can only be specified dynamically. In NATPARM, the macro NTWORK must be used.
Specification within session	NO	

The software components for accessing work files in different environments are called access methods. For the duration of a Natural session, each logical work file can be assigned to one access method only. The access method for a work file is determined by the keyword subparameter AM (see below).

In OS/390 under TSO and in batch mode, work files need not be predefined in the JCL. Provided they are defined by subparameter AM=STD, they can be allocated dynamically during the session by a Natural program using the DEFINE statement or application interface USR2021 (in library SYSEXT).

This document covers the following topics:

- WORK Parameter Syntax
- NTWORK Macro Syntax
- Keyword Subparameters for All Environments
- Keyword Subparameters for AM=STD in All Environments
- Keyword Subparameters for AM=STD in OS/390 Environments
- Keyword Subparameters for AM=STD in VSE/ESA Environments
- Keyword Subparameters for AM=STD in BS2000/OSD Environments
- Keyword Subparameters for AM=CICS
- Keyword Subparameters for AM=COMP (Com-plete)
- Keyword Subparameters for AM=SMARTS (Com-plete)

See also Print and Work File Handling with External Datasets in a Server Environment in the Natural Operations for Mainframes documentation.

WORK Parameter Syntax

With the WORK parameter, you first specify one or more logical work file numbers, and then several keyword subparameters, which define the characteristics for these work files:

```
WORK=((work-file-numbers),keyword-subparameters,...)
```

work-file-numbers

The file numbers must be specified first and enclosed in parentheses. The numbers can be from 1 to 31. They can be specified in any sequence. Multiple numbers must be separated from one another by commas or blanks. To specify a range of numbers, you can use a hyphen (-).

keyword-subparameters

The various types of keyword subparameters are described below.

For work files with different characteristics, you specify different WORK parameters. If any previous definition (or default) for the same work file exists, only the values for the specified keyword subparameters are overwritten, all other values remain unchanged.

Examples:

```
WORK=((2,12,18),AM=STD,DEST='WORK**')
WORK=((1,3,6-11,15),AM=COMP,OPEN=INITOBJ,CLOSE=CMD)
```

NTWORK Macro Syntax

With an NTWORK macro, you first specify one or more logical work file numbers, and then several keyword subparameters, which define the characteristics for these work files:

```
NTWORK (work-file-numbers),keyword-subparameters,...
```

work-file-numbers

The file numbers must be specified first and enclosed in parentheses. The numbers can be from 1 to 31. They can be specified in any sequence. Multiple numbers must be separated from one another by commas. To specify a range of numbers, you can use a hyphen (-).

keyword-subparameters

The various types of keyword subparameters are described below.

For work files with different characteristics, you specify different NTWORK macros. If any previous definition (or default) for the same work file exists, only the values for the specified keyword subparameters are overwritten, all other values remain unchanged.

Examples:

```
NTWORK (2,12,18),AM=STD,DEST='WORK**'
NTWORK (1,3,6-11,15),AM=COMP,OPEN=INITOBJ,CLOSE=CMD
```


Keyword Subparameters for All Environments

The following keyword subparameters are available: AM | DEST | OPEN | CLOSE | LRECL | TRUNC | PAD | PADCHRO | PADCHRI

AM - Type of Access Method

AM=*xxx* specifies the type of access method to be used.

For an online session, all work files to be used have to be assigned to a specific access method.

For a batch session, any work files not assigned to a specific access method will be automatically detected and assigned by the standard batch access method (AM=STD), provided that they have been predefined in the JCL. See also FAMSTD - Overwriting of Print and Work File Access Method Assignments.

STD	Standard sequential files (batch, TSO, TIAM, CMS OS simulation).
COMP	Com-plete work files.
SMARTS	SMARTS work files. Work file on a SMARTS Portable File System (PFS).
CICS	CICS transient data or temporary storage.
CMS	CMS Disk and SFS files.
PC	Entire Connection.
USER	Third-party vendor work-file interface.
OFF	Unassigned. No automatic assignments if FAMSTD=OFF is set.
0	Unassigned. Automatic assignments if FAMSTD=OFF is set. This is the default value.

Note:
WORK=OFF is equivalent to: `WORK=((1-32)),AM=OFF)`.
It does not affect any of the other keyword subparameter specifications.

DEST - External Dataset Name

DEST=*name* specifies the external dataset name (1 - 8 characters).

This corresponds to the *operand1* of the DEFINE WORK FILE statement (and can be overwritten by a DEFINE WORK FILE specification).

The meaning of this keyword subparameter depends on the access method.

AM=STD	DEST is the logical dataset name (DDNAME, LINK name, DTF name). If the destination is to be for multiple files, two asterisks (**) have to be specified for the file number. These will be replaced by the corresponding logical file number for each work file. A DEST value including two asterisks must be enclosed in apostrophes when using it as a dynamic parameter. The default value is DEST='CMWKF**' for IBM and DEST='W**' for SIEMENS environments. Under VSE/ESA, only 7-character names are supported.
AM=CICS	There is no default value for work files under CICS. Here, the DEST subparameter is mandatory; that is, CICS work files defined without a valid DEST specification are ignored. The Natural CICS interface also supports a variable (see TERMVAR parameter in the NCIPARM generation macro; &TID is the default) as part of the DEST value which, when being specified, is replaced by the actual CICS terminal ID. See also Natural Print and Work Files under CICS in the Natural TP Monitors documentation.
AM=CMS	For usage of DEST under CMS, refer to Natural under VM/CMS (in the Natural Operations for mainframes documentation).
AM=COMP	DEST defines the name of the Com-plete SD-file. The length is restricted to a maximum of 8 characters. If the file is defined with TYPE=TID, the DEST value is appended by the Com-plete stack level. The length is restricted to a maximum of 7 characters accordingly. SD-file names starting with '&&' are treated as temporary files which are deleted automatically after Natural termination.

OPEN - Time of File Opening

OPEN=*xxx* determines when the file is to be opened:

Value	The file is opened
INIT	for output at session initialization.
OBF	according to the default OPEN value for the different environments (Batch, CICS, Com-plete, TSO).
OBJ	when the execution of the first object which accesses the file starts. This is the default value.
INITOBF	for output at session initialization. Any subsequent re-opening of the file sets the default OPEN value for the different environments (Batch, CICS, Com-plete, TSO).
OBJ1	when the execution of the first object on level 1 which accesses the file starts. Otherwise, it is opened when it is first accessed.
ACC	when it is first accessed by a statement.
INITOBJ	for output at session initialization. Any subsequent re-opening of the file will be performed when the execution of the first object which accesses the file starts.
INITOBJ1	when the execution of the first object on level 1 which accesses the file starts. Otherwise, it is opened when it is first accessed.
INITACC	for output at session initialization. Any subsequent re-opening of the file will be performed when it is first accessed by a statement.

CLOSE - Time of File Closure

CLOSE=*xxx* determines when the file is to be closed:

Value	The file is closed
OBJ	either when processing of the object in which it was first accessed is finished, or when command mode, NEXT mode or MAINMENU is reached.
CMD	when command mode, NEXT mode or MAINMENU is reached. This is the default value.
FIN	at session end. With CLOSE=FIN, a DEFINE WORK FILE statement causes an error if the work file was opened already. A CLOSE WORK FILE statement for the work file is ignored. When the end-of-file condition occurs during the READ WORK FILE statement, Natural closes the work file immediately.
USER	This value specifies that a work file is closed only if the file is open and one of the following conditions is true: • a CLOSE WORK FILE statement is issued, • a DEFINE WORK FILE statement is issued, • at session termination.

LRECL - Default and Maximum Record Length of Dataset

LRECL=*nnn* determines the record length (in bytes) of the dataset.

Possible values:	0, or 5 - 32767.
Default value:	0

This subparameter is used particularly to check for truncation and padding. For more information on AM=STD, see the keyword subparameter LRECL in the section WORK Keyword Subparameters for AM=STD in All Environments below.

TRUNC - Truncation of Output Records

TRUNC=*xxx* determines whether the output records are truncated or not:

ON	Output records that are longer than the record length (LRECL) of the dataset will be truncated.
OFF	Error NAT1512 will be issued if an output record is longer than the dataset record length. This is the default value.

PAD - Padding of Output Records

PAD=*xxx* determines whether the output records are padded or not (applies only to datasets of fixed record length):

ON	Output records that are shorter than the record length (LRECL) of the dataset will be padded with padding characters defined by keyword subparameter PADCHRO. This is the default value.
OFF	Error NAT1510 will be issued if an output record is shorter than the dataset record length.

PADCHRO - Padding Character of Output Records

This subparameter defines the character which is used for padding of output records if PAD=ON is defined for the work file.

Possible values:	*'x'* *x'xx'*	(one character *x* within single quotes) (one hex character *xx*)
Default value:	*x*'00'	

PADCHRI - Padding Character of Input Records

This subparameter defines the character which is used for padding of input records.

Possible values:	*'x'* *x'xx'*	(one character *x* within single quotes) (one hex character *xx*)
Default value:	*x*'40'	(blank)

WORK Keyword Subparameters for AM=STD in All Environments

The following keyword subparameters are available: RECFM | BLKSIZE | LRECL

RECFM - Default Record Format of Dataset

RECFM=*xxxx* determines the default record format of the dataset.

The following formats are supported:

F	Fixed
V	Variable
U	Undefined
B	Blocked
S	Spanned
A	ASA
M	Machine control characters

The following values and also combinations of values are possible:

Possible value:	F, FA, FM, FB, FBA, FBM, V, VA, VM, VB, VBA, VBM, VBS, VBSA, VBSM, U, UA, UM.
Default value:	RECFM=VB (variable blocked).

The RECFM specification only applies if no record format is predefined in the JCL or (OS/390 only) in the dataset DCB.

BLKSIZE - Default Block Size of Dataset

BLKSIZE=*nnnnn* determines the default block size (in bytes) of the dataset.

Possible values:	0, or 8 to 32767.
Default value:	4628

The BLKSIZE specification only applies if no block size is predefined in the JCL or (OS/390 only) in the dataset DCB.

LRECL - Default and Maximum Record Length of Dataset

LRECL=*nnn* determines the record length (in bytes) of the dataset.

Possible values:	0, or 5 - 32767.
Default value:	0

This subparameter is used particularly to check for truncation and padding.

- For RECFM=V (B) the LRECL value includes a 4-byte record descriptor word.
- If LRECL = 0 is defined, the following applies:
  - With RECFM=V (B), LRECL defaults to BLKSIZE-4.
  - With RECFM=U, LRECL defaults to BLKSIZE.
  - With RECFM=F (B), the maximum record length in the Natural program being executed is taken when the file is opened. If no record length from a program is available when the file is opened, for example with OPEN=INIT, this leads to an error.

The LRECL specification only applies if no record length is predefined in the JCL or (OS/390 only) in the dataset DCB.

Keyword Subparameters for AM=STD in OS/390 Environments

The following keyword subparameters are available: REREAD | FREE | BUFNO | DISP | VMAX

REREAD - Closing of Tape File Datasets

REREAD=*xxx* sets the REREAD option for the closing of the tape file:

ON	The REREAD option is set for the CLOSE SVC. This causes the volume to be repositioned to reprocess the dataset. This is the default value.
OFF	The REREAD option is not set for the CLOSE SVC.

FREE - Dataset De-allocation at File Closure

FREE=*xxx* determines whether the dataset is de-allocated when the file is closed:

ON	The FREE option is set for the CLOSE SVC, which means that the dataset is de-allocated when it is closed (and not at step termination).
OFF	The FREE option is not set for the CLOSE SVC. This is the default value.

BUFNO - Default Number of OS/390 I/O Buffers of Dataset

BUFNO=*nnn* defines the default number of OS/390 I/O buffers of the dataset.

Possible values	0 - 255.
Default value	0. In this case, OS/390 allocates five I/O buffers per default.

The number of I/O buffers can improve the performance of work file access dramatically. Note that the storage for I/O buffers is allocated below the 16 MB line.

The BUFNO specification applies only if the BUFNO parameter is not specified in the JCL for the dataset.

DISP - Open Work File for Modification

DISP=*xxx* determines that the work file is opened for modification.

This corresponds to the JCL DD statement subparameter DISP=MOD.

MOD	New records are added at the end of the file.
NOMOD	The work file is rewritten from the start. This is the default value.

VMAX - Control LRECL for Variable Record Format

VMAX=*xxx* controls the LRECL setting for an output file with variable record format (RECFM=V).

ON	Providing a nonzero BLKSIZE value exists for the file, VMAX=ON sets LRECL=BLKSIZE-4 for variable record format, regardless of the LRECL setting in the DCB or the LRECL subparameter.
NAT	LRECL is set to the length +4 of the largest record in the application program if this value is less than LRECL in the DCB for the dataset.
OFF	LRECL from the DCB for the dataset is used. This is the default value.

Keyword Subparameters for AM=STD in VSE/ESA Environments

The following keyword subparameters are available: SYSNR | LABEL | REWIND

SYSNR - Logical VSE SYS Number

SYSNR=*nn* determines the logical VSE SYS number.

Possible values:	1 - 99.
Default value:	By default, the SYS number is identical to the work file number.

LABEL - Tape Label Processing

LABEL=*xxx d*etermines the tape label processing:

ON	The tape is in standard label format. This is the default value.
OFF	The tape is unlabeled with front tape mark.
NOTM	The tape is unlabeled without front tape mark.

REWIND - Action at File Closure

REWIND=*xxx* determines the action to be taken when a tape file is closed:

ON	The tape is rewound when the file is closed. This is the default value.
OFF	The tape is not rewound when the file is closed.
UNLOAD	The tape is unloaded when the file is closed.

Keyword Subparameters for AM=STD in BS2000/OSD Environments

The following keyword subparameter is available: DISP

DISP - File Open Mode

DISP=*xxx* determines the open mode of the file:

EXT	The open mode is set to EXTEND.
NOEXT	The open mode is set to the default value OUTPUT. This is the default value.

Keyword Subparameters for AM=CICS

The following keyword subparameters are available: TYPE | DISP

TYPE - Type of CICS Storage Medium

TYPE=*xxxx* specifies the type of CICS storage medium to be used:

MAIN	Temporary main storage.
AUX	Temporary auxiliary storage.
TD	Transient data.

The default value used depends on the DEST keyword subparameter setting. If the DEST subparameter value matches a valid CICS transient data queue, the TYPE subparameter defaults to TD, otherwise MAIN will be taken as the default value.

DISP - CICS Temporary Storage Queue Disposition

DISP=(*xxx*,*xxx*) specifies the CICS temporary storage queue disposition.

Possible value pairs are:

(NEW,KEEP)	The storage queue is deleted when the file is opened. This is the default value.
(NEW,DELETE)	The storage queue is deleted when the file is opened and when it is closed.
(OLD,DELETE)	The storage queue is deleted when the file is closed.
(OLD,KEEP)	The storage queue is not deleted.

Note:
The DISP specification does not apply to CICS extra-partition transient data queues.

Keyword Subparameters for AM=COMP

The following keyword subparameters are available: TYPE | BLOCKS | BLKSIZE

TYPE - Type of Storage Access

TYPE=*xxx* specifies the type of storage access to be used:

SHR	Shared access, that is, the work file is accessible by all users.
TID	The work file is only available to the current Com-plete terminal ID.
DYN	The work file is only available to the current terminal stack level.

BLOCKS - Number of Storage Blocks

BLOCKS=*nnnn* specifies the number of storage blocks to be allocated.

Possible values:	1 to 9999
Default value:	20

BLKSIZE - Size of Storage Blocks

BLKSIZE=*nnnn* determines the default block size (in bytes) of the dataset.

Possible values:	0, or 8 - 32767
Default value:	4628

Keyword Subparameters for AM=SMARTS

The following keyword subparameters are available: DEST | TYPE | DISP

DEST - Work File Name

DEST=*name* specifies the workfile name (1-8 characters).

Since the DEST clause is restricted to an 8 character maximum, it is useless to define a file with absolute PFS path specification.

The name specified in the DEST clause is relative to the workfile root directory. The work file root directory is specified with the environment variable NAT_WORK_ROOT.

To specify a file with absolute path definition, the DEFINE WORK statement must be used.

TYPE - Type of Storage Access

TYPE=*xxx* specifies the type of storage access to be used. Possible values are:

BIN	Each line is written to the work file without terminating end-of-line character. This is the default value.
TXT	Each line is written to the work file with a terminating end-of-line character (x'15').

DISP - File Open Mode

DISP=*(Disp1,Disp2,Disp3)* specifies the mode of the work file. Possible values are:

Disp1=xxx	Specifies whether an existing file should be deleted or new data should be appended to the file.	
	NEW	An existing file will be deleted if the file is opened for writing. This is the default value.
	OLD or **MOD**	New data written are appended at the end of the file.
Disp2=xxx	Specifies whether a file should be kept or removed after access.	
	KEEP	Permanent file that will be kept after close. This is the default value.
	DELETE	Temporary file that will be removed after close.
Disp3=xxx	Specifies whether a user has exclusive access to the file or not.	
	SHR	Shared access, that is, the work file is accessible by all users. This is the default value.
	OWN	Exclusive access, the work file is accessible to the current Comp-lete user ID. Files with exclusive access are located in an additional directory which has the name of the current user ID.

WPSIZE - Sizes of Natural Work Pools

This Natural profile parameter is for mainframes only.

It determines the sizes of the Natural work pools below and above the 16 MB line for one Natural session.

Natural uses work pools below and above the 16 MB line. In these work pools, all temporary buffers physical storage requests are satisfied.

Natural uses physical storage in special situations only, for example, for passing parameter areas outside the thread (while the thread is released) during the execution of the CALL statement with the "call by value option" indicated by a SET CONTROL ’P=V’ statement under CICS.

The advantage of work pools is that, if there are many requests for physical storage, Natural can satisfy these requests by itself rather than by passing it to the operating system.

Possible settings	*size-below*	*size-below* (0-1024) is the size of one work pool in KB below the 16 MB line. If the work pool is exhausted, another work pool with the same size is allocated. The value "0" means that no work pool is allocated, i.e. all requests for physical storage below 16 MB are passed directly to the operating system.
	size-above	*size-above* (0-16384) is the size of one work pool in KB above the 16 MB line. If the work pool is exhausted, another work pool with the same size is allocated. The value "0" means that no work pool is allocated, that is, all requests for physical storage above 16 MB are passed directly to the operating system.
	maximum-below	*maximum-below* (0-2097151) limits the total physical storage in KB which can be allocated below the 16 MB line. The value "0" means no physical storage can be allocated below the 16 MB line.
	maximum-above	*maximum-above* (0-2097151) limits the total physical storage storage in KB which can be allocated above the 16 MB line. The value "0" means no physical storage can be allocated above the 16 MB line.
Default setting	(32,128,2097151,2097151)	
Dynamic specification	YES	
Specification within session	NO	

The WPSIZE parameter is specified as follows:

```
WPSIZE=(size-below,size-above,maximum-below,maximum-above)
```

Subparameters not to be changed can be omitted, e.g. you can specify WPSIZE=(,1000) if you want to set the work pool size only above 16MB to 1000 KB.

Natural allocates the work pools outside the Natural storage thread according to the specified settings. A work pool is allocated during the first request for physical storage and is released during the next terminal I/O.

WPSIZE - Sizes of Natural Work Pools

For non-thread environments (e.g. batch, TSO), the recommended setting is WPSIZE=(0,0). This may save virtual storage.

WSISIZE - Buffer for Natural Workstation Interface

This Natural profile parameter is for mainframes and only applies if Natural Workstation Interface is installed.

Alternatively, you can use the equivalent Natural profile parameter DS or macro NTDS (see Using Optional Macros in a Natural Parameter Module in the Natural Operations for Mainframes documentation) to specify the buffer size.

Possible settings	10 - 256	Size of buffer area in KB. If the required space is not available, the Natural Workstation Interface cannot be used.
	0	The Natural Workstation Interface cannot be used.
Default setting	0	
Dynamic specification	YES	
Specification within session	NO	

XADB - Database for Coordination of Transaction Processing

This Natural profile parameter is for UNIX and Windows platforms only. It is reserved for future use.

It specifies the databases, where transaction processing is to be coordinated by a TP-monitor system. Mark all such databases in the parameter's database list box.

Possible settings	ON / OFF	Use the check boxes to specify the relevant databases.
Default setting	none	
Dynamic specification	NO	
Specification within session	NO	

XREF - Active Cross-Reference Feature

This Natural profile parameter is for:

- Mainframes
- UNIX and Windows

It defines whether the active cross-reference feature is to be used.

Possible settings	ON	Natural cross-reference data for database fields, subroutines and maps are stored in the appropriate Predict entries each time a Natural program is cataloged.
	OFF	No cross-reference activity is performed.
	FORCE	A check is made to determine whether a Predict entry for the program exists: if so, CATALOG is allowed and cross-reference data are stored; if not, CATALOG is disallowed.
	DOC	Same as XREF=FORCE, but without generating any cross-reference data.
Default setting	OFF	
Dynamic specification	YES	
Specification within session	NO	

XSIZE - Size of Buffer for User Subsystem

This Natural profile parameter is for mainframes only.

It specifies the size of the buffer area to be used for user subsystems called by Natural programs.

Alternatively, you can use the equivalent Natural profile parameter DS or macro NTDS (see Using Optional Macros in a Natural Parameter Module in the Natural Operations for Mainframes documentation) to specify the XSIZE value.

If Natural Connection is installed and asynchronous lines are used, the XSIZE profile parameter is reserved for internal use by Natural Connection and must not be used otherwise.

Possible settings	1 - 64	Size of the buffer area in KB.
	0	If XSIZE=0 or if the required space is not available, the user subsystem cannot be used.
Default setting	0	
Dynamic specification	YES	
Specification within session	NO	

YD - Year Differential

This Natural profile parameter is for mainframes only.

It can be used to adjust the current machine date (as read by using the internal machine time) by adding/subtracting a number of years to/from it. This may be useful for countries that use different calendars.

Possible settings	-499 to 499	The parameter is specified as `YD=+`*nnn* or `YD=-`*nnn* where *nnn* is the number of years.
Default setting	0	
Dynamic specification	YES	
Specification within session	NO	

Notes:

- If the current year is a leap year, but the year resulting from the YD setting is not, the 1st March will be used instead of the 29th February.
- The year resulting from the sum of the profile parameters TD, DD and YD must be in the range of 1583 through 2699.

YSLW - Year Sliding or Fixed Window

This Natural profile parameter is for all platforms.

YSLW determines the range of years covered by the "year sliding window" or "year fixed window". The sliding-window or fixed-window mechanism assumes a date with a 2-digit year to be within a "window" of 100 years. Within these 100 years, every 2-digit year setting is uniquely related to a specific century, so that there is no confusion about which century is meant.

Possible settings	**Normal Setting**	0	When you set the parameter to 0, the current century is assumed. No sliding or fixed-window mechanism is used.
	Sliding Window	1 - 99	By setting the parameter to a value between 1-99, you determine when the 100-year range begins in the past. The YSLW setting is subtracted from the current year to determine the first year of the window range. **Example:** If the current year is 2002 and you specify YSLW=40, the sliding window will cover the years 1962 to 2061. A 2-digit year setting *nn* from 62 to 99 is then interpreted accordingly as 19*nn*, while a 2-digit year setting *nn* from 00 to 61 is interpreted as 20*nn*.
	Fixed Window	1582-2600	**On mainframe platforms:** By setting the parameter to a value between 1582-2600, you determine the first year of a 100-year range. The upper boundary of the 100-year range is evaluated by adding 99 to the value specified. **Example:** If you specify YSLW=1985, the fixed window will cover the years 1985 to 2084. A 2-digit year setting *nn* from 85 to 99 is then interpreted accordingly as 19*nn*, while a 2-digit year setting *nn* from 00 to 84 is interpreted as 20*nn*.
Default setting	0		No sliding or fixed-window mechanism is used.
Dynamic specification	YES		
Specification within session	NO		

The YSLW parameter is evaluated at runtime when an alphanumeric date setting with a 2-digit year component is moved into a date variable. This applies to data settings which are:

- used with the mathematical function VAL;
- used with the IS(D) option in a logical condition;
- read from the stack as input data;
- or entered in a map as input data.

See also the section Processing of Date Information in the Natural Programming Guide documentation.

ZD - Zero-Division Check

This Natural profile and session parameter is for all platforms.

It determines the action to be taken when an attempt is made to perform a division operation in which the divisor is "0".

Within a Natural session, the profile parameter ZD can be overridden by the session parameter ZD.

Possible settings	ON	Natural issues an error message if a division by "0" is attempted.	
	OFF	Natural returns a result of "0" for any division operation in which the divisor is "0".	
Default setting	ON		
Dynamic specification	YES		
Specification within session	YES	Applicable Statements:	SET GLOBALS
		Applicable Command:	GLOBALS

For details on session parameter specification at statement or element level and evaluation at compilation time or at runtime, refer to Session Parameter Specification/Evaluation Overview.

ZP - Zero Printing

This Natural profile and session parameter is for all platforms.

It determines how a field which contains a setting of all zeros is to be output; that is, it is used to suppress the display of a numeric field (format N, I, P or F) or time field (format T) which contains a value of all zeros.

Within a Natural session, the profile parameter ZP can be overridden by the session parameter ZP.

Possible settings	ON	Each field value which consists of all zeros is output as one zero, right justified (for numeric fields) or all zeros (for time fields).	
	OFF	Each field value which consists of all zeros is suppressed.	
Default setting	ON		
Dynamic specification	YES		
Specification within session	YES	Applicable Statements:	DISPLAY FORMAT INPUT PRINT REINPUT SET GLOBALS WRITE
		Applicable Command:	GLOBALS

For details on session parameter specification at statement or element level and evaluation at compilation time or at runtime, refer to Session Parameter Specification/Evaluation Overview.

ZSIZE - Size of Entire DB Buffer Area

This Natural profile parameter is for mainframes and only applies to Entire DB.

It determines the size of the buffer area required by Entire DB.

Alternatively, you can use the equivalent Natural profile parameter DS or macro NTDS (see Using Optional Macros in a Natural Parameter Module in the Natural Operations for Mainframes documentation) to specify the ZSIZE value.

Possible settings	0 - 64	Size of the buffer area in KB.
Default setting	0	If ZSIZE=0 or if the required space is not available, the Entire DB Interface cannot be used.
Dynamic specification	YES	
Specification within session	NO	

Session Parameters

Natural session parameters may be used with certain Natural statements to control such factors as the size of a report, how fields are to be displayed, etc.

This section covers the following topics:

- Introduction to Session Parameters
- Session Parameters in Alphabetical Order

Note:
For a description in PDF format of the following session parameters, refer to the correspondig section in the Natural profile parameter documentation:

CC, CF, DC, DFOUT, DFSTACK, DFTITLE, DU, EJ, ENDIAN, FC, FCDP, IA, ID, IM, LE, LS, LT, ML, MP, MT, NC, OPF, PD, PM, PS, REINP, SA, SF, SM, SYMGEN, TS, WH, ZD, ZP.

Introduction to Session Parameters

This document covers the following topics:

- Session Parameter Specification/Evaluation Overview
- Session Parameter Usage
- How to Set Session Parameters
- Evaluation of Session Parameters

Session Parameter Specification/Evaluation Overview

X - Parameter may be specified dynamically / with GLOBALS system command / with FORMAT statement.
S - Parameter may be specified at Statement level.
E - Parameter may be specified at Element level.
C - Parameter is evaluated at Compilation time.
R - Parameter is evaluated at Runtime.

Session Parameter / Short Description		Default Value	Dynamic Parameter	GLOBALS Command	Statements								
					SET GLOBALS	FORMAT	DISPLAY	INPUT	ASSIGN, COMPUTE, MOVE, REINPUT	WRITE, WRITE TITLE, WRITE TRAILER	PRINT	CALLNAT, PERFORM	MOVE EDITED
AD	Attribute Definition					X	SE	SE	SE	SE	SE	S	
AL	Alphanumeric Length for Output	none				X	SE	SE		SE	SE		
BX	Box Definition	none				X	SE	SE		SE			
CC	Error Processing in Batch Mode	OFF	X	X	R								
CD	Color Definition	NE				X	SE	SE	S	SE	SE		
CF	Character for Terminal Commands	%	X	X	R								
CO	Compiler Output	OFF	X	X									
CV	Control Variable	none					SE	SE		SE	SE		
DC	Character for Decimal Point Notation	.	X	X	R								
DF	Date Format	S				X	SE	SE		SE	SE		
DFOUT	Date Format for Output	S	X	X	R								
DFSTACK	Date Format for Stack	S	X	X	R								
DFTITLE	Output Format of Date in Standard Report Title	S	X	X	R								
DU	Dump Generation	OFF	X	X	R								
DY	Dynamic Attributes	none					SE	SE		SE	SE		
EJ	Page Eject	ON	X	X	R								
EM	Edit Mask	none				X	SE	SE		SE	SE		S
ENDIAN	Endian Mode for Compiled Objects	DEFAULT	X	X									
ES	Empty Line Suppression	OFF				X	S			S			
FC	Filler Character	blank				X	SE						
FCDP	Filler Character for Dynamically Protected Input Fields	ON	X	X	R								
FL	Floating Point Mantissa Length	none				X	SE	SE		SE	SE		

Session Parameter / Short Description		Default Value	Dynamic Parameter	GLOBALS Command	Statements								
					SET GLOBALS	FORMAT	DISPLAY	INPUT	ASSIGN, COMPUTE, MOVE, REINPUT	WRITE, WRITE TITLE, WRITE TRAILER	PRINT	CALLNAT, PERFORM	MOVE EDITED
FS	Default Format/Length Setting for User-Defined Variables	OFF	X	X	R								
GC	Filler Character for Group Headers	blank				X	SE						
HC	Header Centering	C				X	SE						
HE	Helproutine	none						SE					
HW	Heading Width	ON				X	SE						
IA	Input Assign Character	=	X	X	R								
IC	Insertion Character	none				X	SE						
ID	Input Delimiter Character	,	X	X	R								
IM	Input Mode	F	X	X	R								
IP	Input Prompting Text	ON				X		SE					
IS	Identical Suppress	OFF				X	SE			SE			
KD	Key Definition	OFF				X							
LC	Leading Characters	none				X	SE						
LE	Reaction when Limit for Processing Loop Exceeded	OFF	X	X	C								
LS	Line Size	physical	X	X	C	X	S	S		S			
LT	Limit for Processing Loops	99999999	X	X	R								
MC	Multiple-Value Field Count	1				X	S	S		S	S		

Session Parameter / Short Description		Default Value	Dynamic Parameter	GLOBALS Command	Statements								
					SET GLOBALS	FORMAT	DISPLAY	INPUT	ASSIGN, COMPUTE, MOVE, REINPUT	WRITE, WRITE TITLE, WRITE TRAILER	PRINT	CALLNAT, PERFORM	MOVE EDITED
ML	Position of Message Line	T (mainframe) B (Windows, UNIX)	X	X									
MP	Maximum Number of Pages of a Report	32767				X	S			S	S		
MS	Manual Skip	OFF				X		S					
MT	Maximum CPU Time	60 sec.	X	X	R								
NC	Use of Natural System Commands	OFF	X	X	R								
NL	Numeric Length for Output	none				X	SE	SE		SE	SE		
OPF	Overwriting of Protected Field by Helproutines	ON	X	X	R								
PC	Periodic Group Count	1				X	S	S		S	S		
PD	Number of Pages Captured by NATPAGE	50	X	X	R								
PM	Print Mode	none	X	X	C	X	SE	SE		SE	SE		
PS	Page Size for Natural Reports	physical	X	X	CR	X	S	S		S			
REINP	Issue Internal REINPUT Statement for Invalid Data	ON	X	X	R								
SA	Sound Terminal Alarm	OFF	X	X	R								
SB	Selection Box	none						E					

Session Parameter / Short Description		Default Value	Dynamic Parameter	GLOBALS Command	Statements								
					SET GLOBALS	FORMAT	DISPLAY	INPUT	ASSIGN, COMPUTE, MOVE, REINPUT	WRITE, WRITE TITLE, WRITE TRAILER	PRINT	CALLNAT, PERFORM	MOVE EDITED
SF	Spacing Factor	1	X	X	C	X	SE						
SG	Sign Position	ON				X	SE	SE		SE	SE		
SL	Source Line Length	72	X	X									
SM	Structured Mode	OFF	X	X									
SYMGEN	Generate Symbol Tables	OFF	X	X									
TC	Trailing Characters	none				X	SE						
TS	Translate Output from Programs in System Library	OFF	X	X									
UC	Underlining Character	-				X	SE			S			
WH	Wait for Record in Hold Status	OFF	X	X	R								
ZD	Zero Division Check	ON	X	X	R								
ZP	Zero Printing	ON	X	X	C	X	SE	SE		SE	SE		

Session Parameter Usage

In Natural, session parameters are used:

- to specify certain characters,
- to set processing time limits,
- to set a particular response for a given condition,
- to set various size limits,
- to determine various aspects of output reports.

The statements in which a session parameter may be set/evaluated are indicated in the section Session Parameter Specification/Evaluation Overview as well as in the description of each parameter.

At the installation of Natural, the Natural administrator sets these parameters to default values which are then valid for all users of Natural.

To see which parameter values apply to your session, you enter the system command GLOBALS as described in the Natural System Commands reference documentation.

How to Set Session Parameters

Natural session parameters can be set in several ways:

- via the default parameter module/file (NATPARM), which is set when Natural is installed;
- via dynamic parameters specified when invoking Natural (as described in your Natural Operations documentation);
- via the system command GLOBALS;
- via a SET GLOBALS statement (in reporting mode only);
- via a FORMAT statement;
- via parameter specification within statements where parameters also are evaluated, for example, INPUT, DISPLAY, WRITE;
- via terminal commands.

Instead of the parameter values "ON" and "OFF", you can also specify "T" (true) or "F" (false) respectively.

Changing Parameters at Session Level Using the GLOBALS Command

For your Natural session you can change some of the parameter values set by the Natural administrator.

Within your Natural session, you can change these parameters by issuing the following system command:

GLOBALS

When you issue the GLOBALS command, a screen is displayed showing the parameter values that are currently in effect for your session. On this screen, you can change the values that do not suit your requirements.

A parameter value set with a GLOBALS command remains in effect until the end of the Natural session (and applies to every object you store during the session), unless you change it again with another GLOBALS command.

Changing Parameters at Program Level Using the FORMAT Statement

You can change certain parameters for the duration of a single program (report). This is done by using a FORMAT statement in the program, which will override the session-wide settings for these parameters.

Example of a FORMAT Statement:

```
FORMAT AL=10 HC=R
```

Parameters set with a FORMAT statement apply until the end of the executed program, unless they are changed with another FORMAT statement in the program.

Not all session parameters can be changed at program level, while several parameters that can be specified at program level cannot be specified at session level; most of the latter are parameters which affect the format of an output report.

Changing Parameters at Statement Level

Most of the parameters you can change with a FORMAT statement you can also change for an individual statement; for example, for a particular DISPLAY, WRITE, INPUT or REINPUT statement.

This is done by specifying the parameter (in parentheses) after the statement name.

Example:

```
DISPLAY (SF=4) NAME JOB-TITLE CURR-CODE SALARY
```

A parameter set at statement level applies only to the statement in which it is specified. The setting at statement level overrides, for that statement only, all other settings of that parameter at other levels.

Changing Parameters at Field Level

Within a DISPLAY, WRITE, INPUT or REINPUT statement, you can also change some parameters for an individual field or output element.

This is done by specifying the parameter (in parentheses) after the field name.

Example:

```
DISPLAY NAME (AL=10) JOB-TITLE CURR-CODE SALARY
```

The parameter value then applies only to that field. The setting at field level overrides, for that field only, all other settings of that parameter at other levels. However, only some of the parameters that can be set at statement level can also be set at field level.

Session Parameter Evaluation

Parameters specified with the statements DISPLAY, FORMAT, PRINT, INPUT, REINPUT, WRITE, WRITE TITLE and WRITE TRAILER are processed during program compilation and are therefore included in the corresponding object module for the program.

The following hierarchy is used for evaluation:

1. Parameters set at element/field (highest priority)
2. Parameters set at statement level
3. Parameters set with a FORMAT statement
4. The default parameter settings (lowest priority)

Parameters set with a SET GLOBALS statement cause the execution time environment to be modified. These modifications remain in effect until overridden by another SET GLOBALS statement (or GLOBALS system command).

AD - Attribute Definition

With this session parameter, you specify field attributes at field/element or statement level.

Related session parameter: CD - Color Definition

Possible settings	See below	You can specify multiple attributes in any sequence.
Default setting	See below	
Applicable Statements:	FORMAT	Parameter may be specified dynamically with the FORMAT statement.
	DISPLAY INPUT PRINT REINPUT WRITE	Parameter may be specified at statement level and/or at element level.
	ASSIGN, CALLNAT, CALLDBPROC, COMPUTE, MOVE, OPEN DIALOG, PERFORM, SEND EVENT, SEND METHOD	Parameter may be specified at element level, however, only the attributes specified in the relevant statement description can be used.
Applicable Command:	None	

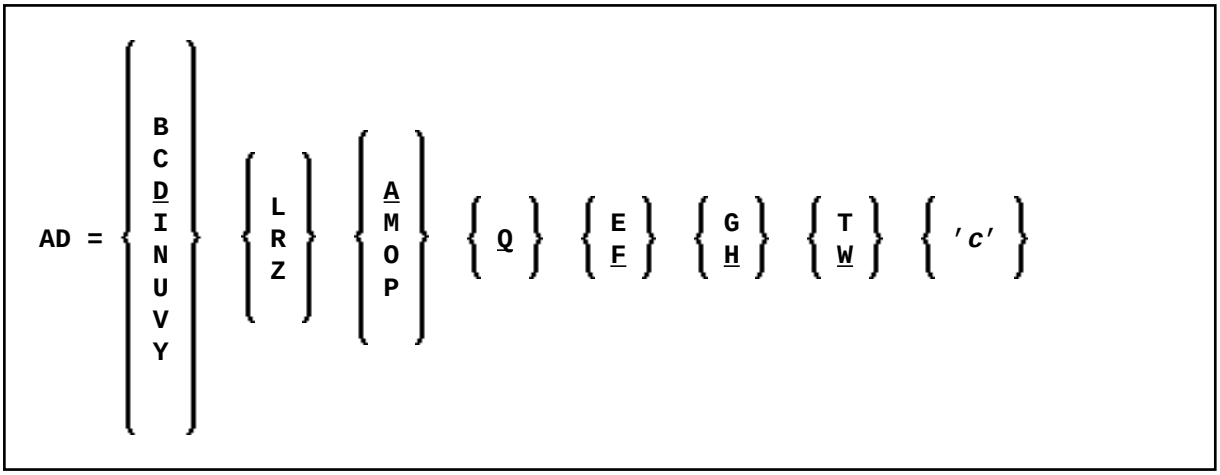
The following topics are covered below:

- AD Parameter Syntax
- Field Representation
- Field Alignment
- Field Input/Output Characteristics
- Interpretation of Alphanumeric Fields
- Mandatory Input
- Length of Input Value
- Field Upper/Lower Case Characteristics
- Filler Character

AD Parameter Syntax

```
AD=[field-representation] [field-alignment]
[field-i/o-characteristics][interpretation-of-alphanumeric-fields] [mandatory-input] [input-value-length]
[field-upper/lower-case] [filler-character]
```

You can specify multiple attributes in any sequence. Possible values are:



The meaning of the attributes and the possible values are explained below.

Examples:

```
DISPLAY #FIELDA (AD=R)
INPUT #FIELDB (AD=M)
INPUT (AD=IM) #FIELDA #FIELDB
```

Field Representation

Value	Meaning	Statements	Explanation
B	blinking (*)	ASSIGN, COMPUTE, MOVE, DISPLAY, FORMAT, INPUT, PRINT, REINPUT, WRITE	The value of the field is displayed blinking.
C	cursive/italic (*)		The value of the field is displayed cursive/italic.
D	default intensity		The value of the field is displayed with normal intensity, that is, not highlighted in any way. This is the default value.
I	intensified		The value of the field is displayed intensified.
N	non-display		A value entered in the field will not be displayed.
U	underlined		The value of the field is displayed underlined.
V	reverse video (*)		The value of the field is displayed reverse video.
Y	dynamic attributes	INPUT, DISPLAY, PRINT, WRITE	Attributes are to be controlled via an attribute control variable (Format C).

* The field representation attributes marked with an asterisk (*) require corresponding hardware features, and will be ignored at runtime if these features are not available.

Field Alignment

Value	Meaning	Statements	Explanation
L	left-justified	DISPLAY, FORMAT, INPUT, PRINT, REINPUT, WRITE	The value of the field is displayed left-justified. This is the default value for alphanumeric fields.
R	right-justified		The value of the field is displayed right-justified. This is the default value for numeric fields.
Z	leading zeros		Numeric values are displayed with leading zeros, right-justified.

Field Input/Output Characteristics

Value	Meaning	Statements	Explanation
A	input field, non-protected	INPUT	The value of the field is to be entered in response to the INPUT statement. This is the default value.
	input only	CALLNAT, CALLDBPROC, OPEN DIALOG, PERFORM, SEND EVENT, SEND METHOD	If you mark a parameter with AD=A, its value will not be passed to the called object (subprogram, stored procedure, subroutine, dialog, method), but it will receive a value from the called object. For a field defined with BY VALUE in the called object's parameter data area, the calling object cannot receive a value. In this case, AD=A only causes the field to be reset to the low value of the respective format (blanks for alphanumeric, binary zeroes for binary and zeroes for numeric fields) before the object is called. For CALLNAT, AD=A may be useful for remote subprograms executed via Natural RPC in a client/server environment to reduce the load of data sent. If a subprogram is executed locally, AD=A fields will be reset to the low value of the respective format before the object is called. If for SEND METHOD, a method is not implemented in Natural, the behavior depends on the method implementation. The parameter is then passed as an initialized variant. Whether the external component is able to return a value is described in the documentation of the external component. It can also be viewed in the Natural Component Browser.

M	output field, modifiable	INPUT	The value of the field is to be displayed during INPUT statement execution, and a different value may be entered by the user. The field is an output field and may be modified.
	modifiable	CALLNAT, CALLDBPROC, OPEN DIALOG, PERFORM, SEND EVENT, SEND METHOD	By default, the passed value of a parameter can be changed in the called object (subprogram, stored procedure, subroutine, dialog, method) and the changed value passed back to the calling object, where it overwrites the original value. For a field defined with BY VALUE in the called object's parameter data area, no value is passed back. If for SEND METHOD, a method is **not** implemented in Natural, the behavior depends on the method implementation. The parameter is then passed BY REFERENCE. Whether the external component accepts a by reference or by value parameter is described in the documentation of the external component. It can also be viewed in the Natural Component Browser.
O	output field, write-protected	INPUT	The value of the field is to be displayed during INPUT execution. The field is an output field and may not be modified.
	non-modifiable	CALLNAT, CALLDBPROC, OPEN DIALOG, PERFORM, SEND EVENT, SEND METHOD	If you mark a parameter with AD=O, the passed value can be changed in the called object (subprogram, stored procedure, subroutine, dialog, method), but the changed value cannot be passed back to the calling object; that is, the field in the calling object retains its original value. Internally, AD=O is processed in the same way as BY VALUE (see the section parameter-data-definition in the description of the DEFINE DATA statement). If for SEND METHOD, a method is implemented in Natural, the parameter is treated like it was defined BY VALUE in the methodŸs parameter data area (see the section PARAMETER Clause in the description of the INTERFACE statement). If for SEND METHOD, a method is **not** implemented in Natural, the behavior depends on the method implementation. The parameter is then passed BY VALUE. Whether the external component accepts a by reference or by value parameter is described in the documentation of the external component. It can also be viewed in the Natural Component Browser.
P	temporarily protected	INPUT, REINPUT	Used in conjunction with an attribute control variable (Format C), the DY parameter (dynamic attributes), and the REINPUT statement.

Interpretation of Alphanumeric Fields

Value	Meaning	Statements	Explanation
Q	display alphanumeric field as if it were a numeric field	ASSIGN, COMPUTE, MOVE, DISPLAY, FORMAT, INPUT, PRINT, REINPUT, WRITE	**This attribute is available on mainframe computers only. A corresponding hardware feature is required.** An alphanumeric field is interpreted as if it were a numeric field. If the field is displayed under the scope of profile or session parameter PM=I, the value of the field is interpreted from left to right instead of right to left.

Mandatory Input

Value	Meaning	Statements	Explanation
E	value mandatory	INPUT	A value must be entered in the field in response to an INPUT statement; otherwise an error message will be issued. This is only relevant for input-only fields (AD=A).
F	value optional	INPUT	A value can, but need not, be entered in the field in response to an INPUT statement. This is the default value.

Length of Input Value

Value	Meaning	Statements	Explanation
G	value size	INPUT	The value entered in the field in response to an INPUT statement must be of the same length as the field. This is only relevant for input-only fields (AD=A).
H	value size	INPUT	The value entered in the field in response to an INPUT statement may be shorter than the field. This is the default value.

Field Upper/Lower Case Characteristics

Value	Meaning	Statements	Explanation
T	translate lower to upper case	INPUT	The value entered is to be translated to upper case.
W	accept lower case	INPUT	Lower case values are to be accepted. To make AD=W effective, you have to specify the value ON for the Natural profile parameter LC. This is the default value.

Filler Character

Value	Meaning	Statements	Explanation
'c'	filler character	INPUT	The empty field is to be filled with the specified character c (for display only) if AD=A (input field, non-protected) or AD=M (output field, modifiable) is specified.

If the filler character is set to blank (X’40’), filling blanks are replaced by X’00’ to allow for insertion of characters without having to clear the remainder of the input field before.

In BS2000/OSD environments, X’00’ characters are displayed as dots on 97xx type terminals. Their appearance can be changed by means of the SIDA utility or with the configuration utility of the respective terminal emulation.

AL - Alphanumeric Length for Output

With this session parameter, you specify the default output length for an alphanumeric field; that is, when it is specified shorter than the field length, the field will be right-truncated.

Possible settings	1 to *n*	*n* = value of LS (line size) parameter minus 1
Default setting	none	
Applicable Statements:	FORMAT	Parameter may be specified dynamically with the FORMAT statement.
	DISPLAY INPUT PRINT WRITE	Parameter may be specified at statement level and/or at element level.
Applicable Command:	None	

Notes:

- It is not recommended to use the AL session parameter for input fields (attribute definition AD=A or AD=M) in an INPUT statement.
- Any edit mask specified for a field (see session parameter EM) will override the AL session parameter for this field.

Example:

```
FORMAT AL=20
```


BX - Box Definition

This session parameter is for Mainframes only.

With this parameter, you specify which parts of a box are to be displayed for field outlining.

Outlining (boxing) is the capability to generate a line around certain fields when they are displayed on the terminal screen. Drawing such "boxes" around fields is another method of showing the user the lengths of fields and their positions on the screen.

The outlining feature is only available on certain types of terminals, usually those which also support the display of double-byte character sets. If the terminal used does not support outlining, this parameter will be ignored at execution time.

Possible settings	T	Top horizontal line. See Note 1.
	B	Bottom horizontal line. See Note 1.
	L	Lefthand vertical line. See Notes 1 and 2.
	R	Righthand vertical line. See Notes 1 and 2.
	ON	Corresponds to BX=TBLR.
	OFF	Causes no boxes to be drawn around the fields concerned.
Default setting	None	
Applicable Statements:	FORMAT	Parameter may be specified dynamically with the FORMAT statement.
	DISPLAY INPUT WRITE	Parameter may be specified at statement level and/or at element level.
Applicable Command:	None	

Notes:

1. You can specify the values T, B, L, R in any order.
2. If you use the session parameter settings BX=L or BX=R, you should switch off Natural’s screen optimization using the profile parameter setting DSC=OFF or the Natural terminal command %RO.

Example:

```
DISPLAY #FIELD1 (BX=RLT) /
        #FIELD2 (BX=TLRB)
```

See also the terminal command %D=.

CD - Color Definition

With this session parameter, you specify the color attributes for fields. If no color screen is used, this parameter will be ignored at runtime.

Related session parameter: AD - Attribute Definition

Possible settings	BL	blue	
	GR	green	
	NE	neutral	
	PI	pink	
	RE	red	
	TU	turquoise	
	YE	yellow	
Default setting	NE		
Applicable Statements:	FORMAT	Parameter may be specified dynamically with the FORMAT statement.	
	DISPLAY INPUT PRINT WRITE	Parameter may be specified at statement level and/or at element level.	
	ASSIGN MOVE REINPUT	Parameter may be specified at statement level.	
Applicable Command:	None		

Example:

```
INPUT (CD=RE) #A #B
```


CO - Compiler Output

This session parameter is for UNIX platforms only.

This parameter should not be used without prior consultation of Software AG support.

With this parameter, you determine whether a compiler listing is to be displayed when you compile a Natural object.

Possible settings	ON	A compiler listing is displayed on the screen. Setting CO=ON causes a significant longer compilation time than without using this parameter (CO=OFF). Especially a CATALL command will take a rather long time to compile all objects. Therefore this parameter should be set to OFF as default.
	OFF	No compiler listing will be displayed.
Default setting	OFF	
Applicable Statements:	SET GLOBALS	
Applicable Command:	GLOBALS	

CV - Attribute Control Variable

This session parameter is used to reference an attribute control variable. An attribute control variable is defined with Format C and is used to assign field attributes dynamically.

Possible settings	B, C, D, I, N, U, V	Field representation attributes (see session parameter AD).
	P	Field protection (see session parameter AD).
	BL, GR, NE, PI, RE, TU, YE	Color (for an explanation of the color codes, see the session parameter CD).
Default setting	None	
Applicable Statements:	DISPLAY INPUT PRINT WRITE	Parameter may be specified at statement level and/or at element level.
Applicable Command:	None	

Example:

```
DEFINE DATA LOCAL
1 #ATTR(C)
1 #A   (N5)
END-DEFINE
...
MOVE (AD=I CD=RE) TO #ATTR
INPUT #A (CV=#ATTR)
   ...
```

An attribute control variable can also be used to check whether the contents of fields have been modified during the execution of an INPUT statement.

```
IF #ATTR MODIFIED ...
```

Note:
If the CV parameter is specified at statement level and at field level and the attribute control variable for the individual field is empty, the attribute control variable for the statement will be used for the field.

DF - Date Format

With the DF session parameter, you determine the length of a date when converted to alphanumeric representation without an edit mask being specified. The DF parameter is evaluated at compilation time.

The sequence of the day, month and year components and the delimiter characters used are determined by the profile parameter DTFORM.

Possible settings	S	8-byte representation with 2-digit year component and delimiters (*yy-mm-dd*). With DF=S, only 2 digits are provided for the year information; this means that if the date value contained the century, this information would be lost during the conversion.
	I	8-byte representation with 4-digit year component and no delimiters (*yyyymmdd*). * See Note.
	L	10-byte representation with 4-digit year component and delimiters (*yyyy-mm-dd*). * See Note.
Default setting	S	
Applicable Statements:	FORMAT	Parameter may be specified dynamically with the FORMAT statement.
	INPUT DISPLAY WRITE PRINT	Parameter may be specified at statement level and/or at element level.
	MOVE COMPRESS STACK RUN FETCH	Parameter may be specified at element level.
Applicable Command:	None	

When the value of a date field is converted to alphanumeric format (for example, in a MOVE, DISPLAY, WRITE or INPUT statement) and no edit mask is specified for the conversion, the default date format as determined by the profile parameter DTFORM is used as edit mask. The same is true for the input validation of a date variable used in an INPUT statement: If no edit mask is specified, the input is validated according to the date format determined by the DTFORM parameter.

*** Note:**
By using DF=I or DF=L, you can gradually change your applications to use 4-digit year representations and at the same time continue to make use of the flexibility provided by the profile parameter DTFORM.

See also Processing of Date Information, Date Format for Alphanumeric Representation - DF Parameter in the Natural Programming Guide.

DY - Dynamic Attributes

This session parameter is used to assign attributes for dynamic attribute field display.

Possible settings	See below.	
Default setting	None	
Applicable Statements:	DISPLAY INPUT PRINT WRITE	Parameter may be specified at statement level and/or at element level.
Applicable Command:	None	

Special identification characters (escape characters) are used to indicate the beginning and end of attribute definitions.

An alphanumeric field which is processed with an INPUT, DISPLAY, WRITE or PRINT statement, and which contains escape characters, is split into subfields at the escape character position. The corresponding attribute is then assigned to the subfield. A blank is substituted for the escape character.

Note:
For a part of a field for which a DY specification takes effect, any specification made with session parameter CV (control variable) will be ignored.

DY Parameter Syntax

DY={{*escape-character1*} [*color-attribute*] [*i/o-characteristics*] [*field-representation-attribute*] } **...** {*escape-character2*}

The possible settings are explained below.

escape-character1

An escape character which denotes the beginning of the attribute definition. Any special character or a hexadecimal number preceded by an apostrophe (’*xx*) may be used.

color-attribute

The color attribute to be assigned. See also session parameter CD (color definition).

BL	blue
GR	green
NE	neutral
PI	pink
RE	red
TU	turquoise
YE	yellow

i/o-characteristics

Value	Meaning
P	Subfield is to be write-protected.

A "P" may be specified to make the subfield write-protected. See also session parameter AD (attribute definition).

field-representation-attribute

Additional attributes to be assigned. See also session parameter AD (attribute definition).

Value	Meaning
B	blinking (*)
C	cursive/italic (*)
D	default intensity
I	intensified
N	non-display
U	underlined
V	reverse video (*)

escape-character2

An escape character which denotes the end of the attribute definition. Any special character (*c*) or a hexadecimal number preceded by an apostrophe (’*xx*) may be used.

...

You may specify up to seven escape sequences (escape characters and attributes) before the character indicating the end of the attribute definitions.

Examples

Example 1:

```
DY=<U>
```

The text string: `THIS <is> UNDERLINED`
is printed as: `THIS is UNDERLINED`

Example 2:

```
DY=<BL|RE/GR>
```

Assigns:

Blue to "<"
Red to "|"
Green to "/"

">" switches back to the initial field color.

Example 3:

```
DY=<P>;
```

The text string:
`Do not overwrite <this>`
is printed as:
`Do not overwrite this` (where "this" is protected)

EM - Edit Mask

With this session parameter, you can specify an edit mask for an input and/or output field that is used in a DEFINE DATA, DISPLAY, INPUT, MOVE EDITED , PRINT or WRITE statement.

Possible settings	See below	
Default setting	None	
Applicable Statements:	FORMAT	Parameter may be specified dynamically with the FORMAT statement.
	DEFINE DATA DISPLAY INPUT PRINT WRITE	Parameter may be specified at statement level and/or at element level.
	MOVE EDITED	Parameter may be specified at element level.
Applicable Command:	None	

The following topics are covered below:

- Syntax
- Blanks in Edit Masks
- Default Edit Masks
- Edit Masks for Numeric Fields
- Edit Masks for Alphanumeric Fields
- Edit Masks for Binary Fields - Format B
- Hexadecimal Edit Masks
- Edit Masks for Date and Time Fields - Formats D and T
- Edit Masks for Logical Fields - Format L

Syntax

For input fields, values must be entered exactly matching the edit mask. If you would like to display the edit mask for an input field, the field should be defined as modifiable (AD=M).

For a database field, a default edit mask may have been defined in the DDM. If you specify with the EM parameter an edit mask for a database field, this edit mask specified will be used instead of any default edit mask which may be defined for the field in the DDM.

If you specify EM=OFF for a field, no edit mask will be used for the field, not even one that may be defined in the DDM.

At statement level of a DISPLAY, FORMAT, INPUT or WRITE statement, no detail field edit mask may be specified, except EM=OFF.

An edit mask overrides any settings for the session parameters AL, NL and SG.

The characters 9, H, X and Z represent significant print positions in numeric (9,Z), hexadecimal (H), and alphanumeric (X) edit masks. For the difference between 9 and Z, see Edit Masks for Numeric Fields, below.

Examples:

```
DISPLAY AA(EM=OFF) AB(EM=XX.XX)
WRITE SALARY (EM=ZZZ,ZZ9)
```

An abbreviated notation can be used for long edit masks. The following examples demonstrate the abbreviated notation which may be used for numeric, hexadecimal, and alphanumeric edit masks:

```
EM=9(4)-9(5)  is equivalent to: EM=9999-99999
EM=H(10)      is equivalent to: EM=HHHHHHHHHH
EM=X(6)..X(3) is equivalent to: EM=XXXXXX..XXX
```

Blanks in Edit Masks

Blanks within an edit mask are represented by the character on your keyboard that in hexadecimal code corresponds to H’20’ (ASCII) or H’5F’ (EBCDIC), that is, the character "^" (or "¬").

Default Edit Masks

If no edit mask is specified for a field, a default edit mask is assigned to the field depending on the field format:

Field Format	Default Edit Mask
A	X
B	H
N, P, I	Z9
F	scientific representation
D	depends on default date format (as set with the **profile parameter** DTFORM)
T	HH:II:SS
L	blank / X

Edit Masks for Numeric Fields

An edit mask specified for a field of format N, P, I, or F must contain at least one 9 or Z. If more 9s or Zs exist than the number of positions contained in the field value, the number of print positions in the edit mask will be adjusted to the number of digits defined for the field value. If fewer 9s or Zs exist, the high-order digits before the decimal point and/or low-order digits after the decimal point will be truncated.

Characters for the Definition of Numeric Edit Masks:

Character	Function		
9	Position to be displayed (one digit of the field value).	**.**	The first period inserted is used as a decimal point. Subsequent periods are treated as literal characters.
Z	Zero suppression for leading zeros. This is the default for numeric fields. The letter Z may be repeatedly specified to represent floating zero suppression. Z must not be specified to the right of the decimal point. A zero value may be displayed as blanks using all Zs in the edit mask (see also the session parameter ZP).		

The 9s or Zs can be preceded by one or more other characters.

Sign Characters

If the first character before the 9s or Zs is +, -, S or N, a sign may be displayed:

Character	Function
+	A floating sign is to be displayed preceding (leading sign character) or following (trailing sign character) the number. The sign may be generated as a plus or minus depending on the value of the field.
-	A floating minus is to be displayed preceding (leading sign character) or following (trailing sign character) the number if the value of the field is negative.
S	A sign is to be displayed to the left of the column. A plus sign is displayed for a positive value and a minus sign is displayed for a negative value.
N	A minus sign is to be displayed to the left of the column if the value of the field is negative.

Literal Leading Characters

Any number of literal leading characters can appear before the first displayable position (as indicated by Z or 9). These must follow any sign character. If there is no sign character and the first literal leading character is +, -, S or N, it must be enclosed in apostrophes. If a literal leading character is H, X, Z or 9, it must be enclosed in apostrophes.

The first literal leading character specified will appear in the output only if the value contains leading zeros and the edit mask is defined with Z (leading zero suppression). This character will then be used as a filler character displayed instead of a blank for leading zeros. Subsequent literal leading characters will be displayed as they are input.

Literal Insertion and Trailing Characters

Literal insertion and trailing characters can also be used. The symbol (^) can be used to represent a leading, inserted, or trailing blank. By enclosing significant characters (9, H, Z, X) in apostrophes, it is possible to use any characters as leading, insertion, or trailing characters. Insignificant edit mask characters need not be enclosed in apostrophes. Within the same edit mask notation, it is possible to have groups of leading, insertion, and/or trailing character strings, some of which are bounded by apostrophes and some of which are not.

Trailing Sign Characters

A trailing sign character can be specified for numeric edit masks by using the + or -& character as the last character in the edit mask. A + will produce a trailing + or - sign depending on the value of the field. A - will produce a trailing space or - sign depending on the value of the field. If a leading and trailing sign are specified in the edit mask, both will be produced.

Examples of Numeric Edit Masks:

The table below lists the results obtained from the original values shown at the top of each column as they are output without editing mask. All values used as column headings represent N format fields. The lines below the top column represent the formats obtained using the different editing masks:

Value	**0000.03 (N4.2)**	**-0054 (N4)**	**+0087 (N4)**	**0962 (N4)**	**1830 (N4)**
Edit Mask					
EM=9.9	0.0	4.	7.	2.	0.
EM=99	00	54	87	62	30
EM=S99	+00	-54	+87	+62	+30
EM=+Z9	+0	-54	+87	+62	+30
EM=-9.99	0.03	-4.	7.	2.	0.
EM=N9	0	-4	7	2	0
EM=*9.99	0.03	4.	7.	2.	0.
EM=Z99	00	54	87	962	830
EM=*DMZZ9.9	DM**0.0	DM*54.	DM*87.	DM962.	DM830.
EM=999+	000+	054-	087+	962+	830+
EM=999-	000	054-	087	962	830
IC=$ EM=ZZZ.99	$.03	$54.	$87.	$962.	$830.
EM=H(6)					
- *ASCII:*	303030303033	30303574	30303837	30393632	31383330
- *EBCDIC:*	F0F0F0F0F0F3	F0F0F5D4	F0F0F8F7	F0F9F6F2	F1F8F3F0

By combining edit masks with the parameters IC and TC, negative numbers can be displayed in varying formats using a DISPLAY statement.

Edit Masks for Alphanumeric Fields

An alphanumeric edit mask which is only to be used with A format fields must contain at least one "X" which represents a character to be displayed. An "H" as the first character designates a hexadecimal edit mask. A blank is represented by a (^) symbol. All other characters except parentheses are permissible including leading, trailing, and insertion characters. It is also possible to specify leading, insertion, or trailing characters enclosed within apostrophes. If the character "X", a closing parenthesis, or a quotation mark is specified as an insertion character, it must be enclosed within apostrophes.

If leading characters are used before the first displayable position X, the first of these leading characters will not be displayed. Trailing characters which immediately follow the last permissible print position will be displayed.

The number of positions to be displayed will be adjusted to the length of the edit mask if the mask is shorter than the field.

Example of Alphanumeric Edit Masks:

The following program lists the alphanumeric edit masks for a field that is defined with format/length A4 and contains the value "BLUE".

```
* EXAMPLE 'EMMASK1'
DEFINE DATA LOCAL
 1 #TEXT  (A4) INIT <'BLUE'>
 END-DEFINE
WRITE NOTITLE 'MASK 1:' 5X #TEXT (EM=X.X.X.X)
       /       'MASK 2:' 5X #TEXT (EM=X^X^X^X)
       /       'MASK 3:' 5X #TEXT (EM=X--X--X)
       /       'MASK 4:' 5X #TEXT (EM=X-X-X-X-X-X)
       /       'MASK 5:' 5X #TEXT (EM=X' 'X' 'X' 'X)
       /       'MASK 6:' 5X #TEXT (EM=XX....XXX)
       /       'MASK 7:' 5X #TEXT (EM=1234XXXX)
 END
```

```
MASK 1:     B.L.U.E
MASK 2:     B L U E
MASK 3:     B--L--U
MASK 4:     B-L-U-E-
MASK 5:     B L U E
MASK 6:     BL....UE
MASK 7:     234BLUE
```

Edit Masks for Binary Fields - Format B

Edit masks for binary fields may be set using "X" or "H" notation. For binary fields, the "X" notation is supported as if "H" had been specified instead of "X".

Hexadecimal Edit Masks

If the character "H" is specified as the first character in an edit mask, the content of an alphanumeric or numeric field will be displayed in hexadecimal format. Each "H" represents two print positions that will occur for each byte in the source field. Characters other than "H" serve as insertion or trailing characters in the mask. The number of positions to be displayed will be adjusted to the length of the edit mask if the mask is shorter than the

field. The length of the edit mask will be adjusted to the length of the field if the field length is shorter than the edit mask.

Insertion or trailing characters may be optionally specified bounded by apostrophes.

All fields displayed with a hexadecimal edit mask are treated as alphanumeric. Therefore, if the edit mask is shorter than the field to be edited, numeric or alphanumeric positions will be displayed from left to right disregarding any decimal point positions.

If a hexadecimal edit mask is used as an input edit mask, every "0-9", "a-f", "A-F", blank and hex zero are accepted as a hex digit.

Note:
Blank and hex zero are regarded as "0" and a lower-case letter ("a-f") is regarded as an upper-case letter.

Edit Mask Examples for Hexadecimal Fields:

The tables below list the hexadecimal edit masks with results obtained from the original fields and values shown above each column. All numeric values ("-10", "+10", "01") to which edit masks have been applied originated in fields defined with N2 format. The alphanumeric value "AB" originated from a field defined with format/length A2.

ASCII:

Value ->	**AB**	**-10**	**+10**	**01**
EM=HH	4142	3170	3130	3031
EM=H^H	41 42	31 70	31 30	30 31
EM=HH^H	4142	3170	3130	3031
EM=H-H	41-42	31-70	31-30	30-31
EM=H	41	31	31	30

Note:
In the case of em=h(n) (hexadecimal output), the value of n must be in the range of 1<=n<=126. If n>126, an error will be displayed only if the corresponding variable has more than 126 elements.

Example:

```
A (A100) := 'A'     /* 100 characters/elements
PRINT A (EM=H(200)) /* no error, as 100<=126, although 200>126
END
```

EBCDIC:

Value ->	**AB**	**-10**	**+10**	**01**
EM=HH	C1C2	F1D0	F1F0	F0F1
EM=H:H	C1 C2	F1 D0	F1 F0	F0 F1
EM=HH:H	C1C2	F1D0	F1F0	F0F1
EM=H-H	C1-C2	F1-D0	F1-F0	F0-F1
EM=H	C1	F1	F1	F0

Edit Masks for Date and Time Fields - Formats D and T

In edit masks for fields which are defined with format D (date) or T (time), the following characters can be specified:

Date - format D, and Time - format T:

Character	Usage
DD	Day.
ZD	Day, with zero suppression.
MM	Month.
ZM	Month, with zero suppression.
YYYY	Year, 4 digits. (see the section Hints for Input Edit Mask)
YY	Year, 2 digits. (see the section Hints for Input Edit Mask)
Y	Year, 1 digit. Must not be used for input fields.
WW	Number of Week. (see the sections Hints for Input Edit Mask and Hints for Week Display in Output Edit Mask
ZW	Number of Week, with zero suppression. (see the sections Hints for Input Edit Mask and Hints for Week Display in Output Edit Mask)
JJJ	Julian day.
ZZJ	Julian day with zero suppression.
NN... or **N(n)**	Name of Day (language-dependent). The maximum length is determined by the number of N's or by n. If the name is longer than the maximum length, it will be truncated; if it is shorter, the actual length of the name will be used.
O	Number of week day The **profile parameter** DTFORM determines whether Monday or Sunday is considered the first day of the week. With DTFORM=U : (Sunday = 1, Monday = 2, etc.). With DTFORM=other : (Monday = 1, Tuesday = 2, etc.).
LL... or **L(n)**	Name of Month (language-dependent). The maximum length is determined by the number of L's or by n. If the name is longer than the maximum length, it will be truncated; if it is shorter, the actual length of the name will be used.
R	Year in Roman numerals (maximum 13 digits). Must not be used for input fields.

Syntactical Restrictions for Date Characters

For **Input** and **Output** edit masks, you **may not** use the following:

text			characters		
month	with	month name	MM or ZM	with	LL or L(n)
day name	with	week day number	NN or N(n)	with	O

For **Input** edit masks, you **may not** use the following:

text			characters		
1-digit year	nor	a year in Roman numerals	Y	nor	R
day	without	month or month name	DD or ZD	without	MM or ZM or LL or L(n)
week	without	year	WW or ZW	without	YYYY or YY
month	without	year	MM or ZM	without	YYYY or YY
julian day	without	year	JJJ or ZZJ	without	YYYY or YY
day name	without	week	NN or N(n)	without	WW or ZW
week day number	without	week	O	without	WW or ZW
julian day	with	month	JJJ or ZZJ	with	MM or ZM
julian day	with	week	JJJ or ZZJ	with	WW or ZW
month	with	week	MM or ZM	with	WW or ZW

Hints for Input Edit Mask

If only year (YY or YYYY) but no month or day is specified within an input edit mask, the values for month and day will both be set to "01". If only year (YY or YYYY) and month (MM) but no day is specified within an input edit mask, the value for day will be set to "01".

If a 2-digits year (YY) is used, the century used to fill up the year representation is the current century by default. However, this does not apply when a Sliding or Fixed Window is set. For more details, please refer to profile parameter YSLW in the Natural Parameter Reference documentation.

If a week number (WW or ZW) but no number of week day (O) or name of day (NN...) is specified, the first day of the week is assumed.

Hints for Week Display (WW or ZW) in Output Edit Mask

When DTFORM=U (USA format) is set, the week starts on Sunday; whereas for all other DTFORM settings the first weekday is Monday. Whether a week is week 52/53 of the old year or week 01 of the new year depends on which year contains more days of the week. In other words, if Thursday (Wednesday for DTFORM=U) of that week is in the previous year, the week belongs to the previous year; if it is in the next year, the week belongs to the next year.

If the number of week (WW or ZW) and a year representation (YYYY or YY or Y) is in the same edit mask, the display for year always corresponds to the week number, regardless of the year in the underlying date field.

Example:

```
DEFINE DATA LOCAL
1 D (D)
END-DEFINE
MOVE EDITED '31-12-2003' TO D(EM=DD-MM-YYYY)
DISPLAY D(EM=DD-MM-YYYY_N(10)) D(EM=DD-MM-YYYY/WW)
```

Although the underlying date is the 31 Dec. 2003, when the week number WW is contained in the edit mask, it displays as:

```
   D                   D
--------------------  -------------
31-12-2003_Wednesday  31-12-2004/01
```

Time - format T - only:

Character	Usage
T	Tenths of a second.
SS	Seconds.
ZS	Seconds, with zero suppression.
II	Minutes.
ZI	Minutes, with zero suppression.
HH	Hours.
ZH	Hours, with zero suppression.
AP	AM/PM element.

Examples of Date and Time Edit Masks:


```
 *    EXAMPLE  'EMDATI' EDIT MASKS FOR DATE AND TIME
 ************************************************************************
WRITE NOTITLE
   'DATE INTERNAL :' *DATX (DF=L) /
   '              :' *DATX (EM=N(9)' 'ZW.'WEEK 'YY)  /
   '              :' *DATX (EM=ZZJ'.DAY 'YYYY)        /
   '     ROMAN    :' *DATX (EM=R)  /
   '     AMERICAN :' *DATX (EM=MM/DD/YYYY)      12X 'OR  ' *DAT4U /
   '     JULIAN   :' *DATX (EM=YYYYJJJ)         15X 'OR  ' *DAT4J /
   '     GREGORIAN:' *DATX (EM=ZD.''L(10)''YYYY) 5X 'OR  ' *DATG ///
 *
   'TIME INTERNAL :' *TIMX                      14X 'OR  ' *TIME /
   '              :' *TIMX (EM=HH.II.SS.T) /
   '              :' *TIMX (EM=HH.II.SS' 'AP) /
   '              :' *TIMX (EM=HH)
 END
```

```
DATE INTERNAL : 2001-01-02
              : Tuesday  1.WEEK 2001
              :  2.DAY 2001
     ROMAN    : MMI
     AMERICAN : 02/01/2001           OR   01/02/2001
     JULIAN   : 200102              OR   200102
     GREGORIAN: 2.January 2001       OR   2 January 2001


TIME INTERNAL : 10:40:59             OR   10:40:59.5
              : 10.40.59.5
              : 01.40.59 PM
              : 13
```


Edit Masks for Logical Fields - Format L

For fields of format L (logical fields), edit masks can be defined as follows:

(**EM** = [*false-string/*] *true-string*)

The false-string must not be longer than 31 characters.

Example of Edit Masks for Logical Field:

```
 /* EXAMPLE 'EMLOGV'
 /* EXAMPLE OF LOGICAL VARIABLE IN LOGICAL CONDITION
 /************************************************************************
DEFINE DATALOCAL
 1 #SWITCH (L) INIT <TRUE>
 1 #INDEX (I1)
 END-DEFINE
 /************************************************************************
 FOR #INDEX 1 5
   WRITE NOTITLE #SWITCH (EM=FALSE/TRUE) 5X 'INDEX =' #INDEX
   WRITE NOTITLE #SWITCH (EM=OFF/ON) 7X 'INDEX =' #INDEX
  IF #SWITCH
   MOVE FALSE TO #SWITCH
  ELSE
   MOVE TRUE TO #SWITCH
  END-IF
 /************************************************************************
 SKIP 1
 END-FOR
 END
```

```
TRUE      INDEX =    1
ON        INDEX =    1

FALSE     INDEX =    2
OFF       INDEX =    2

TRUE      INDEX =    3
ON        INDEX =    3

FALSE     INDEX =    4
OFF       INDEX =    4

TRUE      INDEX =    5
ON        INDEX =    5
```


ES - Empty Line Suppression

With this session parameter, you can suppress the printing of empty lines generated by a DISPLAY or WRITE statement.

Possible settings	ON	A line resulting from a DISPLAY or WRITE statement which contains all blank values will not be printed. This setting is particularly useful when displaying arrays (for example, multiple-value fields or fields contained within a periodic group) to avoid printing a large number of empty lines.	
	OFF		
Default setting	OFF		
Specification within session	YES	Applicable Statements:	DISPLAY FORMAT WRITE
		Applicable Command:	None

To achieve empty suppression for numeric values, the field must be specified with ZP=OFF and ES=ON in order to have null values printed as blanks.

See also the session parameters IS and ZP.

Example:

```
DISPLAY (ES=ON) NAME CITY
```

FC - Filler Character

With this session parameter, you specify the filler character which will appear on either side of a heading produced by a DISPLAY statement across the full column width.

Unlike the GC parameter, which applies to headings across a group of columns, the FC parameter applies to individual columns, .

Possible settings	any character	Filler character for individual headings. FC only applies if the column width is determined by the field length and not by the header (see also session parameter HW); otherwise the FC setting will be ignored.	
Default setting	blank		
Specification within session	YES	Applicable Statements:	DISPLAY FORMAT
		Applicable Command:	None

Example:

```
DISPLAY (FC=*)
```


FL - Floating Point Mantissa Length

With this session parameter, you specify the mantissa length of a floating point variable during input or output.

Possible settings	1 to 16	Mantissa length. The total length is "FL + 6" for sign, exponent, and decimal character.	
Default setting	none		
Specification within session	YES	Applicable Statements:	DISPLAY FORMAT INPUT PRINT WRITE
		Applicable Command:	None

Example:

```
DISPLAY FL=5     ->     +1.2345E+03
```

GC - Filler Character for Group Headers

With this session parameter, you specify the filler character which will appear on either side of a group heading produced by a DISPLAY statement across all field columns that belong to that group.

Unlike the FC parameter, which applies to individual columns, the GC parameter applies to headings across a group of columns.

Possible settings	any character	Filler character for group headers.	
Default setting	blank		
Specification within session	YES	Applicable Statements:	DISPLAY FORMAT
		Applicable Command:	None

Example:

```
DISPLAY (GC=*)
```


HC - Header Centering

This session parameter determines the placement of column headers.

Possible settings	C	Headers will be centered.	
	L	Headers will be left-justified.	
	R	Headers will be right-justified.	
Default setting	C		
Specification within session	YES	Applicable Statements:	DISPLAY FORMAT
		Applicable Command:	None

Example:

```
DISPLAY (HC=L)
```

HE - Helproutine

With this session parameter, you specify the name of a helproutine which is to be assigned to a field.

Possible settings	see below		
Default setting	none		
Specification within session	YES	Applicable Statements:	INPUT
		Applicable Command:	None

Helproutines can be created with the Natural program editor, help maps with the Natural map editor.

The helproutine may then be invoked during processing of an INPUT statement or a map by positioning the cursor under the field and pressing the help function key (as defined with the SET KEY statement) or by entering the help character ("?" by default) into the field.

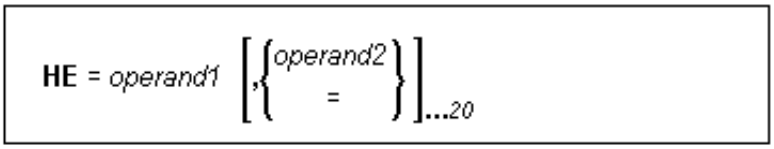
If a value is to be passed from a helproutine to an input field, the field must be defined as modifiable (AD=M).

The following topics are covered below:

- HE Parameter Syntax
- Execution of Helproutines
- Examples

HE Parameter Syntax

The syntax of this parameter is:



Operand	Possible Structure					Possible Formats											Referencing Permitted	Dynamic Definition
Operand1	C	S				A											no	no
Operand2	C	S	A			A	N	P	I	F	B	D	T	L	C	G	O	no	no

Operand1 is the name of the helproutine or map to be invoked. The name may be a 1 to 8 character alphanumeric constant or user-defined variable. If a variable is used, it must have been previously defined. The name may contain an ampersand (&); at execution time, this character will be replaced by the current value of the Natural system variable *LANGUAGE. This feature allows the use of multi-lingual helproutines or maps.

After the helproutine name, you may specify 1 to 20 parameters (operand2) which may be passed to the helproutine. They may be specified as constants or as user-defined variables which contain the values of the parameters. If an "=" is specified as a parameter, the name of the field as defined in the map definition will be passed to the helproutine; in the case of a helproutine which is assigned to a map, "=" denotes the name of the map.

Note:
The operands must be separated either by the input delimiter character (as specified with the session parameter ID) or by a comma. A comma must not be used for this purpose, however, if the comma is defined as decimal character (with the session parameter DC).

If parameters are specified, the helproutine must begin with a DEFINE DATA PARAMETER statement which defines fields that correspond in format and length with the parameters.

If the parameter notation "=" is used to pass a field or map name, the corresponding parameter in the helproutine must be specified as A65.

The value of the field for which a helproutine is specified may be referenced within the helproutine. This is done by specifying a field in the DEFINE DATA PARAMETER statement which corresponds in format and length with the original field. In the block of fields defined within the DEFINE DATA PARAMETER statement, this field must always be defined behind the parameters, if present.

If the field for which a helproutine is specified is an array element, its indices may be referenced by the helproutine. To do so, you specify index parameters with format/length I2 at the end of the DEFINE DATA PARAMETER statement. You may specify up to three index parameters according to array dimensions.

Execution of Helproutines

If a helproutine is requested - by entering a question mark "?" in the field, or by pressing the help key (as defined with a SET KEY statement), or via a REINPUT USING HELP statement - all other data that may have been entered into fields are not assigned to the program variables until all help requests have been processed.

Note:
Only one help request per INPUT statement is possible; that is, if help is requested for more than one field (for example, by entering question marks in multiple fields), only the first help request will be executed.

Examples

Example 1:

```
/* MAIN PROGRAM
DEFINE DATA
1 #A(A20/1:3)
END-DEFINE
...
SET KEY PF1=HELP
...
INPUT #A (2) (HE='HELPA',=)
...
END
```

Example 2:

```
/* HELP-ROUTINE 'HELPA'
DEFINE DATA PARAMETER
1 #VARNAME  (A65)
1 #PARM1    (A20)
1 #VARINDEX (I2)
END-DEFINE
   ...
```


HW - Heading Width

With this session parameter you determine the width of a column output with a DISPLAY statement.

Possible settings	ON	The width of a DISPLAY column is determined by either the length of the heading text or the length of the field, whichever is longer. This is true even if no heading text is output, either because the DISPLAY statement contains the keyword NOHDR or the DISPLAY statement is a subsequent DISPLAY (see also the DISPLAY statement).	
	OFF	The width of a DISPLAY column is determined by the length of the field. HW=OFF only applies to DISPLAY statements which do not create headers (that is, either a first DISPLAY statement with NOHDR option or a subsequent DISPLAY statement).	
Default setting	ON		
Specification within session	YES	Applicable Statements:	DISPLAY FORMAT
		Applicable Command:	None

Example:

```
DISPLAY (HW=OFF)
```


IC - Insertion Character

With this session parameter, you specify the character string to be inserted in the column immediately preceding the value of a field output with a DISPLAY statement. The width of the output column is enlarged accordingly.

For numeric values, the insertion characters will be placed before the first significant digit printed.

The IC and LC parameters are mutually exclusive.

Possible settings	any character	Character string to be inserted. You can specify a string of one to ten characters. Insertion characters may optionally be specified within apostrophes, in which case any characters can be specified. Any character string specified which contains a closing parenthesis or a quotation mark must be enclosed within apostrophes. A blank in a character string not enclosed within apostrophes is represented by the character "^".	
Default setting	none		
Specification within session	YES	Applicable Statements:	DISPLAY FORMAT
		Applicable Command:	None

Examples:

```
DISPLAY AA(IC=*)
DISPLAY SALARY(IC='$')
```

IP - INPUT Prompting Text

This session parameter is used to control prompting text in INPUT statements.

Possible settings	ON	Even if no text is specified preceding the input/output in an INPUT statement, the name of the field will be generated by default as a text element preceding the field as prompting text.	
	OFF	No automatic prompting text will be generated for input/output fields in an INPUT statement. Only fields explicitly preceded with a text element will receive the text as prompting text.	
Default setting	ON		
Specification within session	YES	Applicable Statements:	FORMAT INPUT
		Applicable Command:	None

Example:

```
FORMAT IP=OFF
```


IS - Identical Suppress

With this session parameter, you can suppress the printing of identical information in successive lines created by a WRITE or DISPLAY statement.

The IS parameter setting can be suspended for one record by issuing the SUSPEND IDENTICAL SUPPRESS statement.

The IS parameter may be used in combination with the parameters ES and ZP to cause empty line suppression.

Possible settings	ON	A value which is identical to the previous value for the field will not be displayed. If a DISPLAY or WRITE statement is used to create multiple output lines using the VERT or "/" notation, IS=ON applies only to the first line.	
	OFF	No automatic suppression will be used.	
Default setting	OFF		
Specification within session	YES	Applicable Statements:	DISPLAY FORMAT WRITE
		Applicable Command:	None

Example:

```
FORMAT IS=ON
```

KD - Key Definition

This session parameter is used to display the names assigned to the PF keys (see the SET KEY statement).

This information will always be displayed automatically in the two bottom lines of the physical screen with any output created by the INPUT, WRITE, DISPLAY, and PRINT statement.

As this display requires two lines, the logical page size (see the session parameter PS) must be reduced by two.

Possible settings	ON	The names assigned to the PF keys are displayed.	
	OFF	The names assigned to the PF keys are not displayed.	
Default setting	OFF		
Specification within session	YES	Applicable Statements:	FORMAT
		Applicable Command:	None

Note for graphical user interfaces:
If PF keys are defined, they are always displayed, regardless of the setting of this parameter. If no PF keys are defined, this parameter can be used to switch on/off the display of the ENTER key.

Example:

```
FORMAT KD=ON
```


LC - Leading Characters

With this session parameter, you can specify leading characters that are displayed immediately before a field output by a DISPLAY statement. The width of the output column is enlarged accordingly.

The session parameters LC and IC are mutually exclusive.

Possible settings	Any character	Up to 10 characters may be specified. Leading characters may optionally be specified enclosed within apostrophes, in which case, any characters can be specified. Any character string specified which contains a closing parenthesis or a quotation mark must be enclosed within apostrophes. The character "^" is used to represent a blank in a character string not enclosed within apostrophes.	
Default setting	None		
Specification within session	YES	Applicable Statements:	DISPLAY FORMAT
		Applicable Command:	None

Example:

```
DISPLAY LC=*
```

MC - Multiple-Value Field Count

Note: This parameter may be used in reporting mode only.

With this session parameter, you determine the number of values of a multiple-value field to be output by default when the field is specified without an index in a DISPLAY or WRITE statement.

Possible settings	1 to 191	Number of values.	
Default setting	1		
Specification within session	YES	Applicable Statements:	DISPLAY FORMAT INPUT PRINT WRITE
		Applicable Command:	None

Example:

```
FORMAT MC=5
```


MS - Manual Skip

With this session parameter, you control the cursor positioning during the processing of an INPUT statement.

Possible settings	ON	See example below. **Note:** The setting MS=ON is not supported under BS2000/OSD.	
	OFF	The cursor will be positioned to the next input field as soon as the value for the current field is entered with all positions.	
Default setting	OFF		
Specification within session	YES	Applicable Statements:	FORMAT INPUT
		Applicable Command:	None

Example:

```
INPUT (MS=ON) #A #B
```

NL - Numeric Length for Output

This session parameter determines the default input/output length for a numeric field used in a DISPLAY, INPUT, PRINT or WRITE statement.

The NL parameter must not be specified for groups.

Any edit mask specified for a field will override the NL parameter for this field.

Possible settings	*nn.m*	The length is specified as *nn.m*, where *nn* represents the number of positions before the decimal point and *m* represents the number of positions after the decimal point. The *m* notation is optional. The value of *m* must not exceed 7. The total of *n+m* must not exceed 29. **Notes:** • If NL is set less than the field length, values are truncated. No error is produced when relevant digits are truncated. • If NL is set greater than the field length, values are expanded with blanks. No error is produced when an input field is truncated.	
Default setting	None		
Specification within session	YES	Applicable Statements:	DISPLAY FORMAT INPUT PRINT WRITE
		Applicable Command:	None

Example:

```
DISPLAY #AA(NL=20) #AB(NL=3.2)
```


PC - Periodic Group Count

Parameter	Possible Values	Default Value	Applicable Statements
PC	1 to 99	1	DISPLAY FORMAT INPUT PRINT WRITE

This parameter may be used in reporting mode only.

This session parameter determines the number of periodic group occurrences to be output by default if a periodic group (or a field contained within a periodic group) is specified without an index in a DISPLAY or WRITE statement.

Example:

```
FORMAT PC=5
```


SB - Selection Box

Selection boxes in an INPUT statement are available on mainframe computers only. For other platforms, selection boxes may be defined in the map editor only.

Selection boxes can be attached to input fields. They are a comfortable alternative to help routines attached to fields, since you can code a selection box direct in your program. You do not need an extra program as with help routines.

You may define a selection box clause for every INPUT variable of type alpha, regardless if this field is an input or output field, or both.

The syntax is:

```
SB=operand1 [,operand1]...
```

where *operand1* represents a value operand which is used to fill up the selection box with items.

Operand	Possible Structure					Possible Formats											Referencing Permitted	Dynamic Definition
Operand1	C	S	A			A											yes	no

With SB, you specify the values to be displayed within the selection box.

To assign a selection box to a field:

Specify the attribute SB for an alpha INPUT field in your Natural program using the following example syntax:

```
INPUT #FLD (SB='value1', #ITEM1, #ITEM2(1:3), #ITEM3(*))
```

The following topics are covered below:

- Syntactical Considerations
- Runtime Considerations
- Mainframe Features
- Mainframe Restrictions

Syntactical Considerations

It is possible to assign both a selection box and a help routine to a field.

Selection boxes can be defined for every variable field in an INPUT statement. Exceptions are the following:

System Variables	e.g. *PROGRAM, *COM...
named Constants (mainframe only)	defined with a CONST clause in DEFINE DATA

In addition to the SB attribute, other attributes can be defined as well, e.g. AD or CD.

The selection box field does not have to be modifiable, as is the case with AD=A or AD=M. In other words, it is possible to provide a selection box (and select values) even for a write-protected output field, like AD=O. If you use AD=O, the user is forced to choose from a set of predefined values, which themselves appear in a selection

box.

Runtime Considerations

Selection Box Position

When a program containing a selection box is executed, the selection box is positioned on the screen according to the same positioning algorithm used for help windows, i.e. the size and position of the selection box are determined automatically, ’near’ the field.

Selection Box Attributes

The color and intensified attributes assigned to the field are also applied to the values displayed in the corresponding selection box.

Edit Masks in Selection Boxes

If an edit mask has been defined for the field, the edit mask is applied to all selection box values.

To define an edit mask for a field:

Using the INPUT statement, you can define an edit mask for a field. This is demonstrated in following code example.

```
0010 DEFINE DATA
0020 LOCAL
0030 1 A(A4)
0040 END-DEFINE
0050 MOVE ’ABCD’ TO A
0060 *
0070 SET KEY PF1 = HELP
0080 FORMAT KD=ON
0090 *
0100 INPUT A (AD=M EM=X.X.X.X SB=’1234’,’WXYZ’)
0110 WRITE A
0120 END
```

Selection Box Line Sizes

The line size of the selection box matches the field length to which the box corresponds.

If a value intended for the selection box exceeds the line size of the selection box, the value is truncated.

Sequence of Selection Box Values

Selection box values are displayed in the order they appear in the SB attribute.

Mainframe Features

How a Selection Box is Displayed

For a field with attached selection box, an indicator "V" is displayed next to the field.

Invoking Selection Boxes

To open a selection box:

There are two ways to open a selection box:

1. Enter a question mark (?) in the V-field and press ENTER.
   or
2. Position the cursor on the V-field and press the help key (e.g. PF1), if you have defined it to do so. See the next section for more details.

To define a help key to invoke the selection box:

You can define a help-key (e.g. PF1) to make invoking the selection box much simpler.

This is done by adding the following line of code to your program:

```
SET KEY PF1=HELP
```

Scrolling in a Selection Box

There are two ways to scroll in a selection box:

- By putting the cursor on the "More"-Line and pressing ENTER
  or
- With terminal commands %W- and %W+ assigned to PF-keys (e.g. PF7/PF8).

Selecting a Value in a Selection Box

A value is selected from the selection box and copied into the field by putting the cursor on the value and pressing ENTER.

Duplicate Lines in a Selection Box

Lines with the same content which directly follow each other are suppressed.

For example, the following code

```
INPUT #FLD (SB='123', '456', 'XYZ', 'XYZ', 'XYZ', 'ABC', 'DEF')
```

produces the following output to the selection box:

```
123
456
XYZ
ABC
DEF
```


In the preceding example, XYZ is only displayed once. The other occurrences are considered redundant since they directly follow one another.

However, this line of code

```
INPUT #FLD (SB='123', 'XYZ', '456', 'XYZ', 'ABC', 'XYZ', 'DEF')
```

produces the following output to the selection box:

```
123
XYZ
456
XYZ
ABC
XYZ
DEF
```

In this case, all three occurrences of XYZ are displayed, since they do not directly follow one another.

Blank Lines in Selection Boxes

A blank line is only displayed the first time it appears; all subsequent blank lines are suppressed.

Mainframe Restrictions

The number of operands in the SB clause is limited to 20.

The maximum number of values in a selection box is 248. When that limit has been reached, further values are not displayed. No error message is issued when the limit has been exceeded.

SG - Sign Position

This session parameter determines whether or not a sign position is to be allocated for a numeric field.

If the EM (edit mode) parameter is specified, it overrides the SG parameter.

Possible settings	ON	A sign position will be allocated.	
	OFF	No sign position will be allocated. SG=OFF causes numeric fields with negative values to be output without a "-" sign.	
Default setting	ON		
Specification within session	YES	Applicable Statements:	DISPLAY FORMAT INPUT PRINT WRITE
		Applicable Command:	None

Example:

```
FORMAT SG=OFF
```


TC - Trailing Characters

With this session parameter, you can specify trailing characters that are to be displayed immediately to the right of a field output with a DISPLAY statement. The width of the output column is enlarged accordingly.

Possible settings	Any character	Up to 10 characters may be specified. Trailing characters may optionally be specified enclosed within apostrophes, in which case any characters can be specified. Any character string specified which contains a closing parenthesis or a quotation mark must be enclosed within apostrophes.	
Default setting	None		
Specification within session	YES	Applicable Statements:	DISPLAY FORMAT
		Applicable Command:	None

Examples:

```
FORMAT TC=*
DISPLAY (TC=’*B*’)
```

UC - Underlining Character

This session parameter determines the character that is used as underlining character for the following:

- column headings generated by DISPLAY statements;
- page titles/trailers produced by WRITE TITLE / WRITE TRAILER statements with UNDERLINED option.

Possible settings	Any character	See also Note below.	
	OFF		
Default setting	- (hyphen)		
Specification within session	YES	Applicable Statements:	DISPLAY FORMAT
		Applicable Command:	None

Note:
If you do not wish column headers to be underlined, you have the following options:

- "UC= " - A blank line will be output instead of underlining.
- UC=OFF - The field values will be output immediately below the heading line, without any blank line in between.
  You can specify UC=OFF only at the statement level of a DISPLAY statement; in this case, you cannot make any other UC specifications for individual fields in that statement.

Examples:

```
FORMAT UC=*
DISPLAY (UC= ) NAME AGE (UC=+)
```


WH - Wait for Record in Hold Status

Note: This parameter only applies to Adabas databases.

This session parameter determines the action to be taken if a required record has been placed in hold status by another user.

Possible settings	ON	An error message will be returned if any of these records cannot be placed in hold status.	
	OFF	The user will be placed in wait status until either the requested record becomes available, or an error message is issued due to the database system exceeding a time limit or other limit while attempting to place the record in hold status.	
Default setting	OFF		
Specification within session	YES	Applicable Statements:	SET GLOBALS
		Applicable Command:	GLOBALS

Example:

```
SET GLOBALS WH=ON
```