Natural Parameter Hierarchy

This document describes the hierarchical structure of the different levels on which Natural profile parameters can be set. Various examples are given to illustrate the scenario.

The following topics are covered:

- Natural Parameter Hierarchy Overview
- General Rules for Parameter Usage
- Natural Standard Parameter Module
- Alternative Parameter Module
- Predefined Dynamic Parameter Sets
- Predefined User Parameter Profiles
- Dynamic Parameter Entry
- Natural Security Definitions
- Session Parameter Settings
- Program/Statement Level Settings
- Development Environment Settings
- Examples of Various Parameter Strings

For details of the individual profile parameters, refer to the *Parameter Reference* documentation.

Natural Parameter Hierarchy Overview

Natural profile parameters affect the appearance and the response of a Natural user’s working environment. These parameters are set at different hierarchically organized levels as illustrated in the table below (priority from high to low).

Level	Short Description/References to Detailed Descriptions
During Session	• Development Environment Settings • Program/Statement Level Settings • Session Parameter Settings • Natural Security Definitions
Dynamic during Session Start	• Dynamic Parameter Entry • Predefined User Parameter Profiles • Predefined Dynamic Parameter Sets
Static	• Alternative Parameter Module • Natural Standard Parameter Module

The hierarchically organized levels are discussed in the referenced sections, starting from the lowest and ending with the highest priority.

General Rules for Parameter Usage

The following general rules apply:

- A parameter value set on a higher level overwrites the value defined on a lower level (exceptions: `PROFILE`, `SYS`, `DYNPARM` and some other parameters that work by adding values).
- Dynamic parameters during session start have sequence priority, that is, they are evaluated from left to right.

  Example:

  ```
  ESIZE=20,DATSIZE=60,ESIZE=100
  ```

  The resulting value is `ESIZE=100`.
- Not all of the parameters available at a lower level can be defined on a higher level, too.

Natural Standard Parameter Module

Natural parameters are defined in the standard (default) parameter module which is linked to the Natural nucleus. This module constitutes the bottom level of the Natural parameter hierarchy.

Special Case:

If a shared Natural nucleus is used, this parameter module must be linked to the environment-dependent nucleus module. This parameter module then constitutes the *second hierachical level* and overwrites *all* the parameters of the parameter module which is linked to the shared nucleus (if any). Exception: the `CSTATIC` subprograms of the shared nucleus, see *Statically Linked Non-Natural Programs*.

Alternative Parameter Module

In addition to the Natural standard parameter module, the Natural administrator can define any number of additional (alternative) parameter modules. Such a module is stored in a TP or operating-system library and can be used as alternative parameter module by the parameter PARM when Natural is started.

These parameters cause the parameters of the standard parameter module to be completely overwritten.

Exception: CSTATIC entries, see *Statically Linked Non-Natural Programs*.

Important:
PARM should appear as the first parameter in a dynamic parameter string, because otherwise the alternative parameter module overwrites all parameter settings previously entered in the dynamic parameter string.

You can use the macro NTUSER to restrict the use of an alternate parameter module to a certain user or to several users.

Predefined Dynamic Parameter Sets

The Assembler macro NTSYS can be used to predefine parameter sets which are named in a Natural parameter module. These sets can be addressed under their names when Natural is invoked, provided that the corresponding parameter module is active.

When invoked, the predefined parameter sets react in the same way as dynamically entered parameters in that position.

See also the profile parameter SYS.

Predefined User Parameter Profiles

You can use the Natural utility SYSPARM to create individual profiles which are stored in a system file. Each profile is given a unique character name. You can set values for any dynamic Natural parameters in such a profile.

The profiles created with the utility SYSPARM are activated by using the parameter PROFILE when Natural is invoked.

You can use the profile parameter USER to restrict the use of a profile to a certain user or to several users.

When invoked, the predefined parameter profiles behave in the same way as dynamically entered parameters in that position.

Dynamic Parameter Entry

Almost all of the parameters can be dynamically overwritten when Natural is started. Dynamic parameters are evaluated strictly sequential.

This general overwrite facility can, however, be limited generally or for certain parameters through the use of the profile parameter DYNPARM (only dynamically, for instance in a profile).

You can use the macro NTDYNP in the parameter module NATPARM to make analog settings. This, however, will prohibit the use of the profile parameter DYNPARM.

You can use the file CMPRMIN to define dynamic parameters in batch mode under z/OS, BS2000/OSD and z/VSE, or in batch-like systems such as TSO, TIAM, CMS or BMP environments under IMS TM.

The advantage of this method is that you need not modify the JCL when you wish to change Natural settings. In addition, it overcomes the length limitation of the parameter string (for example, 100 characters under z/OS).

Natural Security Definitions

Apart from protecting the libraries, files and commands, Natural Security enables the setting of certain session-relevant profile parameters.The definitions apply to the current library of the user.

The users can also define settings for their private or default libraries.

The current security settings (session parameters) can be displayed using the Natural system command PROFILE.

The Natural Security parameter definitions are evaluated after the regular profile parameters, that is, they can overwrite them.

Session Parameter Settings

The Natural system command GLOBALS or, in Reporting Mode, the Natural statement SET GLOBALS can be used to display and to set (modify) certain session-relevant profile parameters within and for the duration of a Natural session.

These definitions apply to the command mode and to all programs that are executed during the current session.

See also *Session Parameters for Runtime Assignment of Parameter Values* or SET GLOBALS.

Program/Statement Level Settings

The Natural statement FORMAT can be used in a program to set parameter values which are valid for that specific program.

In addition, it is possible to set certain parameters at statement level by a terminal command.

Development Environment Settings

You can use the Natural Main Menu option *Development Environment Settings* to invoke a submenu which enables selection of the tools that are available for monitoring and setting up the Natural development environment.

Examples of Various Parameter Strings

The examples below are based on the following parameter settings:

Parameter	Param. Module, Shared Nucleus (special case)	Param. Module Front-End	Alternative Param. Module ALTPARM	User Profile MYPROF
DATSIZE	32 (default)	40	50	60
DSIZE	4	6	2 (default)	Not specified
ESIZE	20	28 (default) NTSYS A: 40 NTSYS B: 50	NTSYS A: 60	80

The following examples show the results for various dynamic parameter strings.

Example 1: No dynamic parameters

Resulting Values	Origin
DATSIZE 40	Front-end module
DSIZE 6	Front-end module
ESIZE 28	Front-end module
Others: Default	Front-end module

Example 2: PARM=ALTPARM

Resulting Values	Origin
DATSIZE 50	ALTPARM
Others: Default	ALTPARM

Example 3: SYS=A

Resulting Values	Origin
DATSIZE 40	Front-end module
DSIZE 6	Front-end module
ESIZE 40	NTSYS front-end module

Example 4: PARM=ALTPARM,SYS=A

Resulting Values	Origin
DATSIZE 50	ALTPARM
DSIZE 2	ALTPARM
ESIZE 60	NTSYS, ALTPARM

Example 5: PARM=ALTPARM,SYS=B

Resulting Values	Origin
Error	ALTPARM does not have a NTSYS B specification

Example 6: SYS=A,PROFILE=MYPROF

Resulting Values	Origin
DATSIZE 60	MYPROF
DSIZE 6	Front-end module
ESIZE 80	MYPROF

Example 7: SYS=A,PROFILE=MYPROF,ESIZE=100

Resulting Values	Origin
DATSIZE 60	MYPROF
DSIZE 6	Front-end module
ESIZE 100	Dynamic parameter

Example 8: PROFILE=MYPROF,SYS=A

Resulting Values	Origin
DATSIZE 60	MYPROF
DSIZE 6	Front-end module
ESIZE 40	NTSYS front-end module

Example 9: DSIZE=8,SYS=A,PROFILE=MYPROF,PARM=ALTPARM

Resulting Values	Origin
DATSIZE 50	ALTPARM
Others Default	ALTPARM