Natural

Parameter Reference

Version 9.3.4

October 2026

ADABAS & NATURAL

This document applies to Natural Version 9.3.4 and all subsequent releases.

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

Copyright © 1992-2026 Software GmbH, Darmstadt, Germany and/or its subsidiaries and/or its affiliates and/or their licensors.

The name Software AG and all Software GmbH product names are either trademarks or registered trademarks of Software GmbH and/or its subsidiaries and/or its affiliates and/or their licensors. Other company and product names mentioned herein may be trademarks of their respective owners.

Detailed information on trademarks and patents owned by Software GmbH and/or its subsidiaries is located at https://softwareag.com/licenses.

Use of this software is subject to adherence to Software GmbH's licensing conditions and terms. These terms are part of the product documentation, located at https://softwareag.com/licenses and/or in the root installation directory of the licensed product(s).

This software may include portions of third-party products. For third-party copyright notices, license terms, additional rights or restrictions, please refer to "License Texts, Copyright Notices and Disclaimers of Third-Party Products". For certain specific third-party license restrictions, please refer to section E of the Legal Notices available under "License Terms and Conditions for Use of Software GmbH Products / Copyright and Trademark Notices of Software GmbH Products". These documents are part of the product documentation, located at https://softwareag.com/licenses and/or in the root installation directory of the licensed product(s).

Use, reproduction, transfer, publication or disclosure is prohibited except as specifically provided for in your License Agreement with Software GmbH.

Document ID: NATWIN-NNATPARMS-934-20261013

Table of Contents

Preface ... xi
1 About this Documentation ... 1
Document Conventions ... 2
Online Information and Support ... 2
Data Protection ... 3
2 Introduction to Profile Parameters ... 5
3 Introduction to Session Parameters ... 7
Session Parameter Usage ... 8
How to Set Session Parameters ... 8
Session Parameter Evaluation ... 10
4 ACIVERS - Define API Version for Use with EntireX Broker ACI ... 11
5 ACTPOLICY - Defines Default Activation Policy for DCOM Classes ... 13
6 AD - Attribute Definition ... 15
AD Parameter Syntax ... 16
Field Representation ... 17
Field Alignment ... 18
Field Input/Output Characteristics ... 18
Interpretation of Alphanumeric Fields ... 20
Mandatory Input ... 20
Length of Input Value ... 20
Field Upper/Lower Case Characteristics ... 21
Filler Character ... 21
7 ADAPRM - Adabas Review Support ... 23
8 AL - Alphanumeric Length for Output ... 25
9 AUTO - Automatic Logon ... 27
10 AUTOREGISTER - Automatic Registry Update ... 29
11 AUTORPC - Automatic Natural RPC Execution ... 31
12 BATCH - Batch Mode Simulation ... 33
13 BATCHMODE - Batch Mode ... 35
14 BMBLANK - Display Trailing Blanks ... 37
15 BMCONTROL - Display Control Characters ... 39
16 BMFRAME - Window Frame Characters ... 41
17 BMSIM - Similar Batch Mode Output ... 43
18 BMTIME - Display Process Time ... 45
19 BMTITLE - Display Window Title ... 47
20 BMVERSION - Display Natural Version ... 49
21 BPID - Specify Buffer Pool ID ... 51
22 BPID2 - Specify Secondary Buffer Pool ... 53
23 BPSFI - Object Search First in Buffer Pool ... 55
24 CC - Error Processing in Batch Mode ... 57
25 CD - Color Definition ... 59
26 CDYNAM - Dynamic Loading of Non-Natural Programs ... 61
27 CF - Character for Terminal Commands ... 63

28 CLEAR - Processing of CLEAR Key in NEXT Mode 65
29 CM - Command Mode 67
30 CMOBJIN - Batch Input File for Natural INPUT Data 69
31 CMPRINT - Batch Output File 71
32 CMPRTnn - Additional Report 73
33 CMSYNIN - Batch Input File for Natural Commands and INPUT Data 75
34 CMWRKnn - Natural Work Files 77
35 COMPR - Set RPC Buffer Compression 79
36 COMSERVERID - Determine DCOM Server ID 81
37 COVERAGE - Code Coverage of a Natural Session 83
COVERAGE Parameter Syntax 84
Examples of COVERAGE Parameter 85
38 CP - Default Code Page Name 87
39 CPCVERR - Code Page Conversion Error 89
40 CPOBJIN - Code Page of Batch Input File 91
41 CPPRINT - Code Page of Batch Output File 93
42 CPRPC - Define Code Page Name 95
43 CPSYNIN - Code Page of Batch Input File for Commands 97
44 CV - Attribute Control Variable 99
45 CVMIN - Control Variable Modified at Input 101
46 DBGAT - Debug Attach Server for NaturalONE 103
DBGAT Parameter Syntax 104
Example of DBGAT Parameter 105
47 DBSHORT - Interpretation of Database Field Short Names 107
48 DBUPD - Database Updating 111
49 DC - Character for Decimal Point Notation 113
50 DD - Day Differential 115
51 DF - Date Format 117
52 DFOUT - Date Format for Output 119
53 DFS - Specify RPC Client's Default Server Address 121
54 DFSTACK - Date Format for Stack 123
55 DFTITLE - Output Format of Date in Standard Report Title 125
56 DL - Display Length for Output 127
57 DTFORM - Date Format 129
58 DU - Dump Generation 131
59 DY - Dynamic Attributes 133
DY Parameter Syntax 134
Examples 136
60 DYNPARM - Control Use of Dynamic Parameters 137
Settings 138
61 ECHECK - Existence Check for Object Calling Statements 139
62 ECHO - Control Printing of Batch Input Data 141
63 EDTBPSIZE - Software AG Editor Buffer Pool Size 143
64 EDTLFILES - Number of Software AG Editor Logical Files 145
65 EJ - Page Eject 147

66 EM - Edit Mask 149
EM Parameter Syntax 150
Examples 151
Blanks in Edit Masks 151
Default Edit Masks 151
Edit Masks for Numeric Fields 152
Edit Masks for Alphanumeric Fields 155
Edit Masks for Binary Fields - Format B 157
Hexadecimal Edit Masks 157
Edit Masks for Date and Time Fields - Formats D and T 159
Edit Masks for Logical Fields - Format L 163
Edit Masks with MOVE EDITED (Edit Mask Specified with operand2) 164
67 EMFM - Edit Mask Free Mode 165
68 EMU - Unicode Edit Mask 167
69 ENDIAN - Endian Mode for Compiled Objects 169
70 ENDMSG - Display Session-End Message 171
71 ES - Empty Line Suppression 173
72 ESCAPE - Ignore Terminal Commands %% and %. 175
73 ESXDB - Database ID Used for Entire System Server DDMs 177
74 ET - Execution of END/BACKOUT TRANSACTION Statements 179
75 ETA - Error Transaction Program 181
76 ETDB - Database for Transaction Data 183
77 ETEOP - Issue END TRANSACTION at End of Program 185
78 ETID - Adabas User Identification 187
79 ETIO - Issue END TRANSACTION upon Terminal I/O 189
80 FC - Filler Character for INPUT Statement 191
81 FC - Filler Character for DISPLAY Statement 193
82 FCDP - Filler Character for Dynamically Protected Input Fields 195
83 FDDM - Natural System File for DDMs 197
84 FDIC - Predict System File 199
85 FL - Floating Point Mantissa Length 201
86 FNAT - Natural System File for System Programs 203
87 FREEGDA - Release GDA in Utility Mode 205
88 FS - Default Format/Length Setting for User-Defined Variables 207
89 FSEC - Natural Security System File 209
90 FUSER - Natural System File for User Programs 211
91 GC - Filler Character for Group Headers 213
92 GFID - Global Format IDs 215
93 GPGEN - Generate GP Information 217
GPGEN Parameter Syntax 218
Examples of GPGEN Parameter 219
94 HC - Header Centering 221
95 HD - Header Definition 223
96 HE - Helproutine 225
HE Parameter Syntax 226

Execution of Helproutines 228
Examples 228
97 HI - Help Character 231
98 HW - Heading Width 233
99 IA - Input Assign Character 235
100 IC - Insertion Character 237
101 ICU - Unicode Insertion Character 239
102 ID - Input Delimiter Character 241
103 IKEY - Processing of PA and PF Keys 243
104 IM - Input Mode 245
105 INIT-LIB - Library for Automatic Logon 247
106 IP - INPUT Prompting Text 249
107 IS - Identical Suppress 251
108 ITERM - Session Termination in Case of Initialization Error 253
109 KC - Check for Statement Keywords 255
110 KCHECK - Check for Statement Keywords 257
111 KD - Key Definition 259
112 LC - Lower to Upper Case Translation 261
113 LC - Leading Characters 263
114 LCU - Unicode Leading Characters 265
115 LDB - Wait Time for Response of Local Adabas Database 267
116 LE - Reaction when Limit for Processing Loop Exceeded 269
117 LFILE - Logical System File Definition 271
LFILE Parameter Syntax 272
Example of LFILE Parameter 273
118 LOGONRQ - Logon for RPC Server Request Required 275
119 LS - Line Size 277
Profile Parameter LS 278
Session Parameter LS 278
Specification with Statements 279
120 LT - Limit for Processing Loops 281
121 MADIO - Maximum DBMS Calls between Screen I/O Operations 283
122 MAINPR - Override Default Output Report Number 285
123 MASKCME - MASK Compatible with MOVE EDITED 287
124 MAXBUFF - Default Buffer Size 289
125 MAXCL - Maximum Number of Program Calls 291
126 MAXPREC - Maximum Number of Digits after Decimal Point 293
127 MAXYEAR - Maximum Year for Date/Time Values 295
128 MC - Multiple-Value Field Count 297
129 MFBS - Multifetch Buffer Size 299
130 MFMR - Multifetch Max Records 301
131 MFSET - Multi-Fetch Setting 303
132 ML - Position of Message Line 305
133 MP - Maximum Number of Pages of a Report 307
134 MS - Manual Skip 309

135 MSGSF - Display System Error Messages in Short/Full Format 311
136 MT - Maximum CPU Time 313
137 NATLOG - Natural Log File 315
Examples 316
138 NC - Use of Natural System Commands 319
139 NCFVERS - NCF File Protocol Version 321
140 NL - Numeric Length for Output 323
141 NOAPPLERR - Suppress Message Number Prefix NAT 325
142 OPF - Overwriting of Protected Fields by Helproutines 327
143 OPRB - Database Open/Close Processing 329
OPRB String Syntax 330
144 PARM - Alternative Parameter File 333
145 PC - Periodic Group Count 335
146 PCHECK - Parameter Checking for Object Calling Statements 337
147 PECK - PCHECK/ECHECK Error Processing 339
148 PERSIST - Flag a Parameter File as Persistent 341
149 PM - Print Mode 343
Profile Parameter PM 344
Session Parameter PM 344
150 PRGPAR - Data to be Passed to Program Receiving Control at Termination 347
151 PROFILER - Profile a Natural Session 349
PROFILER Parameter Syntax 350
Examples of PROFILER Parameter 353
152 PROGRAM - Non-Natural Program Receiving Control after Termination 355
153 PS - Page Size for Natural Reports 357
154 PSIGNF - Internal Representation of Positive Sign of Packed Numbers 359
155 RCFIND - Handling of Response Code 113 for FIND Statement 361
156 RCGET - Handling of Response Code 113 for GET Statement 363
157 RDACTIVE - Activate Remote Debugger 365
158 RDNODE - Remote Debugger Node Name 367
159 RDPORT - Remote Debugger Port 369
160 RDS - Define Remote Directory Server 371
161 RECAT - Dynamic Recataloging 373
162 REINP - Issue Internal REINPUT Statement for Invalid Data 375
163 RI - Release ISNs 377
164 RNCONST - Renumber Line Numbers in Constants 379
165 ROSY - Read-Only Access to System Files 381
166 RPCSDIR - Library for Service Directory 383
167 RTINT - Allow Runtime Interrupt 385
168 RQTOUT – REQUEST DOCUMENT Timeout 387
169 SA - Sound Terminal Alarm 389
170 SB - Selection Box 391
Syntactical Considerations 392
Runtime Considerations 393
171 SD - Time Delay between Two Screens 395

172 SERVER - Start Natural Session as an RPC Server Session 397
173 SF - Spacing Factor 399
174 SG - Sign Position 401
175 SM - Programming in Structured Mode 403
176 SNAT - Sound a Bell at Syntax Error 405
177 SORTSZE - Size of Sort Buffer 407
178 SPODDEBUGPORT - Debugger Port for Debugging in the Context of SPoD 409
179 SRETAIN - Retain Source Format 411
180 SRVCMIT - Server Commit Time 413
181 SRVNAME - Name of RPC Server 415
182 SRVNODE - Name of Node 417
183 SRVRTRY - Number of Connect/Reconnect Attempts 419
184 SRVTERM - Server Termination Event 421
185 SRVUSER - User ID for RPC Server Registry 423
186 SRVWAIT - Wait Time of RPC Server 425
187 SSIZE - Size of Source Area Allocated by the Editors 427
188 STACK - Place Data/Commands on the Stack 429
189 STARTUP - Program Name for System Variable *STARTUP 431
190 STEPLIB - Initial Setting for *STEPLIB System Variable 433
191 SUBCHAR - Substitution Character for Default Code Page 435
192 SUTF8 - UTF-8 Format for Sources 437
193 SYMGEN - Generate Symbol Table 439
194 SYNERR - Control of Syntax Errors 441
195 TC - Trailing Characters 443
196 TCU - Unicode Trailing Characters 445
197 TD - Time Differential 447
198 TF - Translation of Database ID/File Number 449
TF Parameter Syntax 451
199 THSEP - Dynamic Thousands Separator 453
200 THSEPCH - Thousands Separator Character 455
201 TIMEOUT - Wait Time for RPC Server Response 457
202 TMPSORTUNIQ - Unique Names for Temporary Sort Work Files 459
203 TQ - Translate Quotation Marks 461
204 TQMARK - Translate Quotation Marks 463
205 TRACE - Define Trace Level for Natural RPC Servers 465
206 TRANSP - Server Transport Protocol 467
207 TRYALT - Try Alternative Server Address 469
208 UC - Underlining Character 471
209 UDB - User Database ID 473
210 ULANG - User Language 475
211 USEDIC - Common Logical Name for Dictionary Servers 477
212 USER - User ID 479
Settings 480
213 USEREP - Repository Usage 481
214 USIZE - Size of User Buffer 483

215 WEBIO - Use Natural Web I/O Interface 485
216 WFOPFA - Opening of Work Files 487
217 WH - Wait for Record in Hold Status 489
218 WORK - Work-File Assignments 491
Settings 492
219 XREF - Creation of XRef Data for Natural 493
Possibilities of Setting the XREF Parameter 494
XRef Data Generation 495
Extended XRef Data Generation (For Internal Use Only) 495
220 YD - Year Differential 497
221 YSLW - Year Sliding or Fixed Window 499
Examples of YSLW Parameter 501
222 ZD - Zero-Division Check 503
223 ZP - Zero Printing 505

Preface

This documentation contains detailed descriptions of all Natural profile and session parameters provided to configure your Natural environment.

If a Natural session parameter with the same name and functionality as a Natural profile parameter exists, the descriptions of both parameters are combined in a single document.

Introduction to Profile Parameters	References to documents providing detailed information on profile parameter usage.
Introduction to Session Parameters	General information on session parameter usage and evaluation.
Parameters in Alphabetical Order	Descriptions of all profile parameters and session parameters in alphabetical order.

1 About this Documentation

- Document Conventions 2
- Online Information and Support 2
- Data Protection 3

Document Conventions

Convention	Description
Bold	Identifies elements on a screen.
`Monospace font`	Identifies service names and locations in the format *`folder.subfolder.service`*, APIs, Java classes, methods, properties.
`Italic`	Identifies: Variables for which you must supply values specific to your own situation or environment. New terms the first time they occur in the text. References to other documentation sources.
`Monospace font`	Identifies: Text you must type in. Messages displayed by the system. Program code.
{ }	Indicates a set of choices from which you must choose one. Type only the information inside the curly braces. Do not type the { } symbols.
|	Separates two mutually exclusive choices in a syntax line. Type one of these choices. Do not type the | symbol.
[ ]	Indicates one or more options. Type only the information inside the square brackets. Do not type the [ ] symbols.
...	Indicates that you can type multiple options of the same type. Type only the information. Do not type the ellipsis (...).

Online Information and Support

Product Documentation

You can find the product documentation on our documentation website at **https://documentation.softwareag.com**.

Product Training

You can find helpful product training material on our Learning Portal at **https://learn.softwareag.com**.

Tech Community

You can collaborate with Software GmbH experts on our Tech Community website at **https://tech-community.softwareag.com**. From here you can, for example:

- Browse through our vast knowledge base.
- Ask questions and find answers in our discussion forums.
- Get the latest Software GmbH news and announcements.
- Explore our communities.
- Go to our public GitHub and Docker repositories at **https://github.com/softwareag** and **https://hub.docker.com/publishers/softwareag** and discover additional Software GmbH resources.

Product Support

Support for Software GmbH products is provided to licensed customers via our Empower Portal at **https://empower.softwareag.com**. Many services on this portal require that you have an account. If you do not yet have one, you can request it at **https://empower.softwareag.com/register**. Once you have an account, you can, for example:

- Download products, updates and fixes.
- Search the Knowledge Center for technical information and tips.
- Subscribe to early warnings and critical alerts.
- Open and update support incidents.
- Add product feature requests.

Data Protection

Software AG products provide functionality with respect to processing of personal data according to the EU General Data Protection Regulation (GDPR). Where applicable, appropriate steps are documented in the respective administration documentation.

2 Introduction to Profile Parameters

For detailed information on using a profile parameter, see the following documents:

- Profile Parameter Usage in the *Operations* documentation
- Creating a New Parameter File in the *Configuration Utility* documentation
- Overview of Profile Parameters in the *Configuration Utility* documentation

3 Introduction to Session Parameters

- Session Parameter Usage 8
- How to Set Session Parameters 8
- Session Parameter Evaluation 10

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.

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` (described in the *System Commands* documentation).

How to Set Session Parameters

Natural session parameters can be set in several ways:

- via the default Natural parameter 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 the **Session Parameters** command in the **Tools** menu of Natural Studio;
- 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 Session 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 Session 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 Session 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 Session 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).

4 ACIVERS - Define API Version for Use with EntireX Broker

ACI

This keyword subparameter has been deprecated and is ignored. The highest possible API version is negotiated dynamically by the RPC nucleus and EntireX.

5 ACTPOLICY - Defines Default Activation Policy for DCOM Classes

This Natural profile parameter 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	

6 AD - Attribute Definition

- AD Parameter Syntax 16
- Field Representation 17
- Field Alignment 18
- Field Input/Output Characteristics 18
- Interpretation of Alphanumeric Fields 20
- Mandatory Input 20
- Length of Input Value 20
- Field Upper/Lower Case Characteristics 21
- Filler Character 21

With this session parameter, you specify field attributes at field/element or statement level.

Related session parameter: CD - Color Definition

Possible settings	See *AD Parameter Syntax*.	You can specify multiple attributes in any sequence.
Default setting	See below.	
Applicable statements	FORMAT	
	DISPLAY INPUT NEWPAGE WITH TITLE PRINT REINPUT WRITE WRITE TITLE WRITE TRAILER	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	

Note: The AD parameter may be also specified in function calls, however, only the attributes specified in the section *Function Call* (*Programming Guide*) can be used.

The following topics are covered below:

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:

```
AD= { B } { L } { A } { E } { G } { T } { 'c' }
    { C }
    { D } { R } { M } { F } { H } { W }
    { I } { Z } { O }
    { N }       { P }
    { U }
    { V }
    { Y }
```

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 FORMAT	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 Function Call	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 FORMAT	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	By default, the passed value of a parameter can be changed in the called object (subprogram, stored procedure, subroutine, dialog,

Value	Meaning	Statements	Explanation
		OPEN DIALOG PERFORM SEND EVENT SEND METHOD Function Call	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 FORMAT	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 Function Call	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 a call-by-value (see BY VALUE in 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 method's parameter data area (see the *PARAMETER clause* 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 call 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.

Note: The Field Input/Output Characteristics A, M and O of the AD parameter may be also specified in function calls.

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 z/OS 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 FORMAT	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 FORMAT	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 FORMAT	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 FORMAT	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 FORMAT	The value entered is to be translated to upper case.
W	accept lower case	INPUT FORMAT	Lower case values are to be accepted. AD=W is the default value. **Note:** To make AD=W effective, you have to specify the value ON for the Natural profile parameter LC.

Filler Character

Value	Meaning	Statements	Explanation
'*c*'	filler character	INPUT FORMAT	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.

Before the value is displayed for a modifiable field (AD=M), field positions that are not occupied by the value are filled with the specified filler character as follows:

- Leading or trailing positions (depending on the field alignment) are filled for format I, N and P fields.
- Trailing positions are filled for format A fields.

If the user enters a value in response to the INPUT statement, before the value has been assigned to the field,

- both leading and trailing filler characters are removed for format I, N and P fields,
- trailing filler characters are removed for format A fields.

Caution: Filler characters that may occur as part of the value in either leading or trailing position should be avoided to prevent undesired results. For example, if the filler character "0" (zero) is defined for a field of format N5 and the value 00100 is entered as input data, leading and trailing zeroes are removed so that only the value 1 remains, and will be assigned to the field. For the same reason, the minus sign "-" should be avoided as a filler character for numeric fields if negative values are to be entered.

7 ADAPRM - Adabas Review Support

This Natural profile parameter is used to pass Natural session data to Adabas Review within the seventh Adabas buffer.

Possible settings	ON	Natural session data is passed. **Note:** Set ADAPRM to ON if Adabas Review is installed.
	OFF	No Natural session data is passed.
Default setting	OFF	
Dynamic specification	yes	
Specification within session	no	

8 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	
	DISPLAY INPUT PRINT WRITE	Parameter may be specified at statement level and/or at element level.
Applicable command	none	

Notes:

1. It is not recommended to use the AL session parameter for input fields (**attribute definition** AD=A or AD=M) in an INPUT statement.
2. Any edit mask specified for a field (see session parameter EM) will override the AL session parameter for this field.

Example:

```
FORMAT AL=20
```

See also *Parameters to Influence the Output of Fields* in the *Programming Guide*.

9 AUTO - Automatic Logon

This Natural profile parameter 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	
Application programming interface	`USR1005N`	See *SYSEXT - Natural Application Programming Interfaces* in the *Utilities* documentation.

Notes:

1. The setting of the profile parameter `INIT-LIB` is used as library ID for the logon.
2. If used with Natural Security, `AUTO=ON` disables logons with another user ID, and the `INIT-LIB` parameter is not evaluated (see the *Natural Security* documentation for further information).

10 AUTOREGISTER - Automatic Registry Update

This Natural profile parameter is used by NaturalX. It determines whether Natural classes are to be registered as DCOM classes each time they are cataloged.

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	

Note: Alternatively, Natural classes can be registered manually with the system command `REGISTER`.

11 AUTORPC - Automatic Natural RPC Execution

This Natural profile 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).

Possible settings	`ON`	Natural RPC will automatically try to execute it remotely.
	`OFF`	Natural RPC will not automatically try to execute it remotely. **Note:** With `AUTORPC=OFF`, you can execute `CALLNAT`s remotely using interface objects.
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.

Notes:

1. If you want to use a remote `CALLNAT` statement to execute a subprogram on an EntireX RPC server, we strongly recommend that you set `AUTORPC=OFF` and use an interface object. For details, see *Interface Objects and Automatic RPC Execution* in the *Natural RPC (Remote Procedure Call)* documentation.
2. `AUTORPC` is specified on the client side only.
3. For details see *Interface Objects and Automatic RPC Execution* in the *Natural RPC (Remote Procedure Call)* documentation.
4. For information on Natural RPC, see the *Natural RPC (Remote Procedure Call)* documentation.

12 BATCH - Batch Mode Simulation

This Natural profile parameter 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. **Note:** The log file is named *natbatch.log* and is located in the Natural binary directory.
	`OFF`	Error messages are displayed but not written to a log file.
Default setting	`OFF`	
Dynamic specification	yes	This parameter can only be specified dynamically.
Specification within session	no	

Note: To run Natural in batch mode, use the parameter `BATCHMODE` instead of `BATCH`.

13 BATCHMODE - Batch Mode

This Natural profile parameter applies to batch mode only. It enables batch mode and sets the system variable `*DEVICE` to `BATCH` when Natural is started.

For information on batch mode operation, see *Natural in Batch Mode* in the *Operations* documentation.

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	

14 BMBLANK - Display Trailing Blanks

This Natural profile parameter is used to control the display of trailing blanks in the batch output file **CMPRINT**.

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	

Notes:

1. This Natural profile parameter applies to batch mode only.
2. This parameter applies only if the parameter BMSIM is set to MF.
3. Trailing blanks are generated automatically if BMSIM is set to MF.
4. BMBLANK has no effect if BMSIM is set to OS or VM.

15 BMCONTROL - Display Control Characters

This Natural profile parameter controls the output of control characters (such as form feed and line feed) in the batch output file CMPRINT.

Note: This Natural profile parameter applies to batch mode only.

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	

16 BMFRAME - Window Frame Characters

With this parameter you can define window frame characters that will be written to the batch output file `CMPRINT`.

Note: This Natural profile parameter applies to batch mode only.

Possible settings	6 characters	Specify a sequence of 6 characters (see *Example*).	
		Character Position:	Displays:
		1	Horizontal bar
		2	Vertical bar
		3	Upper-left corner
		4	Upper-right corner
		5	Lower-left corner
		6	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
```

17 BMSIM - Similar Batch Mode Output

This Natural profile parameter is used for the general appearance description of the batch mode output file `CMPRINT`.

Note: This Natural profile parameter applies to batch mode only.

<table>
<tr><td rowspan="2">Possible settings</td><td>MF</td><td>Forces output similar to Natural for z/OS: 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:
<table>
<tr><th>Control Code</th><th>Interpretation</th></tr>
<tr><td>blank</td><td>Normal output line without control characters</td></tr>
<tr><td>0</td><td>Insert one empty line</td></tr>
<tr><td>-</td><td>Insert two empty lines</td></tr>
<tr><td>+</td><td>Print this line twice (bold printing)</td></tr>
<tr><td>1</td><td>Form feed before printing this line</td></tr>
</table></td></tr>
<tr><td>OS</td><td>Forces output similar to Natural for Linux and Cloud.

The following control character codes are used:
<table>
<tr><th>Control Code</th><th>Interpretation</th></tr>
<tr><td>\n</td><td>Line feed before printing this line</td></tr>
<tr><td>\f</td><td>Form feed before printing this line</td></tr>
</table></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>

18 BMTIME - Display Process Time

This Natural profile parameter 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`.

Note: This Natural profile parameter applies to batch mode only.

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
```

19 BMTITLE - Display Window Title

This Natural profile parameter is used to control the displaying of window titles in the batch output file CMPRINT.

Note: This Natural profile parameter applies to batch mode only.

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	

20 BMVERSION - Display Natural Version

This Natural profile parameter 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	

21 BPID - Specify Buffer Pool ID

This Natural profile parameter 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	The parameter can only be specified dynamically.
Specification within session	no	

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

22 BPID2 - Specify Secondary Buffer Pool

This Natural profile parameter specifies the name (ID) of a secondary buffer pool.

When Natural runs with a read-only buffer pool as the primary buffer pool, objects missing in the read-only buffer pool cannot be loaded. To avoid this, Natural can attach during execution to a secondary standard buffer pool (which allows read/write access) and activate the missing objects there. For further information, see *Secondary Read/Write Buffer Pool* in the *Operations* documentation.

Possible settings	1 - 8 characters	Name of the secondary buffer pool.
Default setting	none	
Dynamic specification	yes	
Specification within session	no	

23 BPSFI - Object Search First in Buffer Pool

This Natural profile parameter determines 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. **Caution:** If you set `BPFSI=ON`, make sure that object names are unique across all libraries that are involved in the search. If objects with the same name exist in different libraries being searched, unpredictable results may occur.
	`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	

For further information, see *Steplibs* and *Search Sequence for Object Execution* in the *Using Natural Studio* documentation.

24 CC - Error Processing in Batch Mode

This Natural profile parameter specifies the action to be taken if an error is detected during the compilation/execution of a Natural program in batch mode.

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. If all input is processed, Natural terminates with Return Code 61 and writes the Natural error 9987 (Error occurred during execution/compilation.) to the batch output file (if the profile parameter ENDMSG is set to ON).	
Default setting	OFF		
Dynamic specification	yes		
Specification within session	yes	Applicable Statements:	
		Applicable command:	
Application programming interface	USR1005N	See *SYSEXT - Natural Application Programming Interfaces* in the *Utilities* documentation.	

Notes:

1. This Natural profile parameter only applies in batch mode.
2. It does not apply if user-written error-handling routines are used.

25 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	
	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
```

26 CDYNAM - Dynamic Loading of Non-Natural Programs

This Natural profile parameter determines whether or not non-Natural programs are to be loaded dynamically by Natural.

Possible settings	ON	Any number of non-Natural programs can be loaded dynamically during the execution of a Natural program.
	OFF	Dynamic loading of non-Natural programs is not performed by Natural.
Default setting	ON	
Dynamic specification	yes	
Specification within session	no	

27 CF - Character for Terminal Commands

This Natural profile and session parameter 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.

Possible settings	any special character	A terminal command must begin with the character specified here. 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`
Application programming interface	`USR1005N`	See *SYSEXT - Natural Application Programming Interfaces* in the *Utilities* documentation.	

Notes:

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

2. Under Natural Security:, the setting of this parameter can be overridden by the *Session Parameters* option of the Library Profile.

28 CLEAR - Processing of CLEAR Key in NEXT Mode

This Natural profile parameter 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.
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	

Note: Under Natural Security:, the setting of this parameter can be overridden by the Session Parameters option of the Library Profile.

Example:

```
CF=%
CLEAR=R
```

Natural executes the terminal command %R when the CLEAR key is pressed in NEXT mode.

29 CM - Command Mode

This Natural profile parameter 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	
Specification within session	no	

30 CMOBJIN - Batch Input File for Natural INPUT Data

This Natural profile parameter 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	
Default setting	none	
Dynamic specification	yes	
Specification within session	no	

Notes:

1. This Natural profile parameter applies to batch mode only.
2. If the file name or path assigned to this parameter contains special characters (e.g. backslash) or spaces, the entire string must be enclosed in double quotes, see example below.
3. If the setting for the profile parameter `CMSYNIN` is equal to the setting of `CMOBJIN`, Natural reads input from `CMSYNIN`.
4. If an error occurs, Natural reacts in accordance with the setting of the profile/session parameter `CC`.

Example:

```
CMOBJIN="C:\tmp\data.txt"
```

31 CMPRINT - Batch Output File

This Natural profile parameter applies to batch mode only.

It is used to specify the batch output file 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`.

Note: If the file name or path assigned to this parameter contains special characters (for example, backslash) or spaces, the entire string must be enclosed in double quotes, see example below.

Possible settings	any string	`CMPRINT="C:\tmp\out.txt"`
Default setting	none	
Dynamic specification	yes	
Specification within session	no	

32 CMPRTnn - Additional Report

This Natural profile parameter 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 LPT device used by a report in a `DISPLAY`, `PRINT` and `WRITE` statement.

Possible settings	any string	If the file name or path assigned to this parameter contains special characters, e.g. backslash (\) or spaces, the entire string must be enclosed in double quotes. **Example:** `CMPRT07="C:\tmp\report7.txt"`
Default setting	none	
Dynamic specification	yes	
Specification within session	no	

In order to allow the user to specify variable print file names, alpha-format system variables and numeric counter markers may be embedded in the file name specification for `CMPRT`*nn*.

The supported alpha-format system variables are:

```
*APPLIC-ID
*APPLIC-NAME
*DEVICE
*ETID
*INIT-USER
*LIBRARY-ID
*NET-USER
*PID
*PROGRAM
```

```
*USER
*USER-NAME
```

If any of these strings (in upper case only) is encountered within the print file specification, it will be replaced at run-time with the contents of the appropriate system variable. Additionally, a counter marker (#) may be used. This will be replaced by a 2-digit counter which will automatically be incremented for each print file.

Example:

The specification `CMPRT01=abc_*PID_*ETID_*PROGRAM_#.dat` in a Natural session with process ID `1234`, ETID `XYZ` running a program with the name `PRINT` which produces print file output to File 01 would produce print files with the following names (assuming the program runs 3 times):

```
abc_1234_XYZ_PRINT_01.dat
abc_1234_XYZ_PRINT_02.dat
abc_1234_XYZ_PRINT_03.dat
```

See also *Using Natural in Batch Mode* in the *Operations* documentation.

33 CMSYNIN - Batch Input File for Natural Commands and INPUT Data

This Natural profile parameter 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.

Note: If the file name or path assigned to this parameter contains special characters, e.g. backslash (`\`) or spaces, the entire string must be enclosed in double quotes, see example below.

Possible settings	any string	`CMSYNIN="C:\tmp\cmd.txt"`
Default setting	none	
Dynamic specification	yes	
Specification within session	no	

See also *Using Natural in Batch Mode* in the *Operations* documentation.

34 CMWRKnn - Natural Work Files

This Natural profile parameter 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.

Note: If the file name or path assigned to this parameter contains special characters, e.g. backslash (`\`) or spaces, the entire string must be enclosed in double quotes, see example below.

Possible settings	any string	`CMWRK05="C:\tmp\workfile5.sag"`
Default setting	none	
Dynamic specification	yes	
Specification within session	no	

See also *Using Natural in Batch Mode* in the *Operations* documentation.

35 COMPR - Set RPC Buffer Compression

This Natural profile and session parameter can be used to set the RPC buffer compression.

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	The setting of this parameter can be changed using the GLOBALS system command.

Notes:

1. COMPR is specified on the client side only.
2. COMPR is effective only, if the automatic Natural RPC execution is used (AUTORPC=ON) and the CALLNAT is executed without an interface object. If an interface object is used, the compression has already been set during interface object generation. For details, see *Using Compression* in the *Natural RPC (Remote Procedure Call)* documentation.
3. For further information, see the *Natural RPC (Remote Procedure Call)* documentation.

36 COMSERVERID - Determine DCOM Server ID

This Natural profile parameter 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	

37 COVERAGE - Code Coverage of a Natural Session

- COVERAGE Parameter Syntax 84
- Examples of COVERAGE Parameter 85

This profile parameter is used to perform a code coverage of a Natural session. The coverage data is written to an NCVF (Natural coverage file) resource file you can analyze with NaturalONE or with the Natural Profiler utility. For more information, see the *NaturalONE* documentation or the *Natural Profiler* documentation.

Possible settings	See ***COVERAGE Parameter Syntax***.	
Default setting	none	See the default settings of the subparameters in ***COVERAGE Parameter Syntax***.
Dynamic specification	yes	
Specification within session	no	

COVERAGE Parameter Syntax

The `COVERAGE` parameter is specified as follows:

COVERAGE={(*subparameter=value*[,*subparameter=value*]...)}

Important: Blank spaces are not allowed in the syntax. Use commas to separate the syntax elements.

Where:

Subparameter	Value	Description
`ACTIVE`	`ACTIVE=`*`value`* determines whether the coverage infrastructure is activated. Default: `OFF`	
	`ON`	The coverage infrastructure is activated and coverage data is written to the resource file.
	`OFF`	The coverage infrastructure is deactivated.
`RESLIB`	1 - 8 characters	`RESLIB=`*`value`* specifies the name of the Natural library that contains the resource file. It is possible to specify an environment variable for *`value`*. Default: `SYSTEM`
`RESNAME`	1 - 253 characters	`RESNAME=`*`value`* specifies the name of the resource file (without path and extension) into which the data is written. It is possible to specify an environment variable for *`value`*. Default: A file name is automatically generated containing the current user ID and timestamp.

Examples of COVERAGE Parameter

Example 1: Code coverage using the default resource

```
COVERAGE=(ACTIVE=ON)
```

The coverage data is written to an NCVF resource file with a name containing the current user ID and timestamp in the library SYSTEM on the FUSER.

Example 2: Code coverage using a specific resource

```
COVERAGE=(ACTIVE=ON,RESNAME=MYAPP,RESLIB=MYLIB)
```

The coverage data is written to the resource file *MYAPP.ncvf* in the library *MYLIB*.

Example 3: Code coverage using a variable

```
$COVNAME='123'
                $COVLIB='ABC'
                COVERAGE=(ACTIVE=ON,RESNAME=$COVNAME,RESLIB=$COVLIB)
```

The coverage data is written to the resource file *123.ncvf* in the library *ABC*.

Note: In a Windows environment, you could also use `%covname%` or `%covlib%` instead of `$COVNAME` or `$COVLIB`.

38 CP - Default Code Page Name

This Natural profile parameter defines the default code page for Natural data and Natural sources.

Possible settings	1 - 64 characters	The name of the desired code page.
	' ' (blank)	Reset to system code page.
Default setting	' ' (blank)	System code page.
Dynamic specification	yes	
Specification within session	no	

Note: The system code page is detected via ICU (International Components for Unicode).

39 CPCVERR - Code Page Conversion Error

This Natural profile and session parameter specifies whether a conversion error that occurs when converting

- from Unicode to code page or
- from code page to Unicode or
- from one code page to another code page

results in a Natural error or not. Anyway, after the conversion, the target operand will contain the conversion result where all characters which can not be converted will be replaced by a substitution character which is defined by ICU for the affected code page.

Notes:

1. This parameter is not regarded for the conversion of Natural sources when loading them into the source area or during catalog.
2. On z/OS, it is not regarded whether a Unicode field is converted into the code page before an I/O on a terminal emulation. In this case, the substitution character is replaced by the placeholder character which is defined in `NATCONFG`.
3. If `CPCVERR` is `ON`, the Natural output window will not accept characters which are not contained in the current code page for A format fields. If `CPCVERR` is `OFF`, all characters will be accepted for A format fields. The runtime will then replace the characters which are not contained in the current code page with the substitution character of this code page (or parameter `SUBCHAR`, if defined).

Possible settings	ON	A Natural error NAT3413 is issued, if at least one code point could not be translated correctly during ICU conversion.
	OFF	No error is generated if one or more code points could not be translated correctly during ICU conversion.
Default setting	ON	
Dynamic specification	yes	
Specification within session	yes	
Applicable statements	SET GLOBALS	
Applicable command	GLOBALS	

See also:

- *Profile Parameters* in the *Unicode and Code Page Support* documentation.
- *Using an Error Transaction Program* in the *Programming Guide*

40 CPOBJIN - Code Page of Batch Input File

This Natural profile parameter specifies the code page of the batch input file which is defined by the Natural profile parameter CMOBJIN.

Possible settings	1 -64 characters	ICU code page name (IANA name recommended).
	' ' (blank)	The code page resulting from the evaluation of the profile parameter CP is used.
Default setting	' ' (blank)	
Dynamic specification	yes	
Specification within session	no	

See also *Profile Parameters* in the *Unicode and Code Page Support* documentation.

41 CPPRINT - Code Page of Batch Output File

This Natural profile parameter specifies the code page of the batch output file which is defined by the Natural profile parameter `CMPRINT`.

Possible settings	1 - 64 characters	ICU code page name (IANA name recommended).
	' ' (blank)	The code page resulting from the evaluation of the profile parameter `CP` is used.
Default setting	' ' (blank)	
Dynamic specification	yes	
Specification within session	no	

See also *Profile Parameters* in the *Unicode and Code Page Support* documentation.

42 CPRPC - Define Code Page Name

This parameter specifies the name of the code page used by the EntireX Broker.

Note: Currently, it applies only to the Natural RPC facility when the transport protocol ACI (that is EntireX Broker) is used.

Possible settings	1 - 40 characters	Valid code page name of EntireX Broker.
Default setting	none	
Dynamic specification	yes	
Specification within session	no	

Notes:

1. `CPRPC` can be specified on both the client and the server side.
2. For information on the EntireX Broker, refer to the section about Internationalization in the EntireX Broker documentation.
3. See also *Unicode and Code Page Support*, *Configuration and Administration of the Unicode/Code Page Environment*, *Profile Parameters*.

43 CPSYNIN - Code Page of Batch Input File for Commands

This Natural profile parameter specifies the code page of the batch input file for commands which is defined by the Natural profile parameter CMSYNIN.

Possible settings	1 - 64 characters	ICU code page name (IANA name recommended).
	' ' (blank)	The code page resulting from the evaluation of the profile parameter CP is used.
Default setting	' ' (blank)	
Dynamic specification	yes	
Specification within session	no	

See also *Profile Parameters* in the *Unicode and Code Page Support* documentation.

44 CV - Attribute Control Variable

This session parameter is used to reference an attribute control variable.

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 PROCESS PAGE REINPUT WRITE	Parameter may be specified at statement level and/or at element level.
Applicable command	none	

Notes:

1. An attribute control variable is defined with Format C (see *Special Formats* in the *Programming Guide*) and is used to assign field attributes dynamically and/or check the "modified" status of a field in conjunction with an INPUT or PROCESS PAGE statement; see also *Logical Condition Criteria, MODIFIED Option - Check whether Field Content has been Modified* in the *Programming Guide*.
2. By specifying the MODIFIED option of the IF statement, the attribute control variable can be used to check whether the contents of a field has been modified during the execution of an INPUT or PROCESS PAGE statement: IF #ATTR MODIFIED ...
3. A single attribute control variable can be applied to several input fields by specifying it once at statement level or multiple times at element level, in which case the "modified" status indication is set if any of the fields referencing the control variable has been modified. If the CV parameter is specified both 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.

4. The attribute control variable can be expanded up to three dimensions, for example, `CONTR(*)`, `CONTR(*,*)`, `CONTR(*,*,*)`, depending on the rank of the corresponding array.

Example:

```
DEFINE DATA LOCAL
1 #ATTR(C)
1 #A   (N5)
END-DEFINE
...
MOVE (AD=I CD=RE) TO #ATTR
INPUT #A (CV=#ATTR)
...
```

45 CVMIN - Control Variable Modified at Input

This Natural profile parameter 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.

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	

Note: 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.

46 DBGAT - Debug Attach Server for NaturalONE

- DBGAT Parameter Syntax 104
- Example of DBGAT Parameter 105

This Natural profile parameter allows debugging of an external Natural application with NaturalONE.

Possible settings	See *DBGAT Parameter Syntax*.	
Default setting	none	See also the default settings of the subparameters in *DBGAT Parameter Syntax*.
Dynamic specification	yes	This parameter can only be specified dynamically.
Specification within session	no	
Application programming interface	no	

For detailed information on how to debug external Natural applications, see the *NaturalONE* documentation.

This section covers the following topics:

DBGAT Parameter Syntax

The `DBGAT` parameter is specified as follows:

```
DBGAT=(subparameter=value,...)
```

Important: Spaces are not allowed in the syntax. The optional syntax elements are separated from each other using commas.

Where:

Subparameter	Value	Description
`ACTIVE`	`ON` `OFF`	`ON` means that the debug attach mechanism is active. The Natural runtime is ready for debugging. If `ACTIVE` is not specified, the corresponding value defined in the Configuration Utility is used as the default.
`HOST`	1 - 64 characters	Name of the debug attach server that is to be connected. If `HOST` is not specified, the corresponding value defined in the Configuration Utility is used as the default.
`PORT`	0 - 65535	Number of the port to which the debug attach server listens. If `PORT` is not specified, the corresponding value defined in the Configuration Utility is used (2500 is the default).
`CLID`	1 - 64 characters	Client ID of the NaturalONE project that is to be debugged.

Subparameter	Value	Description
		If CLID is not specified, the corresponding value defined in the Configuration Utility is used as the default.
SSL	ON	The communication to the debug attach server uses SSL/TLS encryption.
	OFF	The communication to the debug attach server does not use SSL/TLS encryption.
TLS	ON	See description for SSL.
	OFF	
VERIFY	ON	Natural will verify the SSL/TLS connection to the debug attach server. Natural will first search the CA certificate in the das.client.ca.crt file stored in the location of the Natural binary. If the file does not exist, Natural will try to search in the SSL_CERT_FILE environment variable or SSL_CERT_DIR for the CA certificate location (OpenSSL standard).
	OFF	Natural will not verify the SSL/TLS connection.

Example of DBGAT Parameter

```
DBGAT=(ACTIVE=ON,HOST=MYHOST,PORT=9999,CLID=MYCLIENTID,SSL=ON,VERIFY=ON)
```

47 DBSHORT - Interpretation of Database Field Short Names

This Natural profile and session parameter can be used to determine the interpretation of database field short names. It corresponds to the option `DBSHORT` of the Natural system command `COMPOPT`.

A database field defined in a DDM is described by two names:

- the short name with a length of 2 characters, used by Natural to communicate with the database (especially with Adabas);
- the long name with a length of 3-32 characters (1-32 characters, if the underlying database type accessed is Db2/SQL), which is supposed to be used to reference the field in the Natural programming code.

Under special conditions, you may reference a database field in a Natural program with its short name instead of the long name. This applies if running in Reporting Mode without Natural Security and if the database access statement contains a reference to a DDM instead of a view.

The decision if a field name is regarded as a short-name reference depends on the name length. When the field identifier consists of two characters, a short-name reference is assumed; a field name with another length is considered as a long-name reference. This standard interpretation rule for database fields can additionally be influenced and controlled by setting the compiler option `DBSHORT` to `ON` or `OFF`:

Possible settings	`ON`	Using a short name is allowed for referencing a database field. However, a data base short name is *not permitted* in general (even if `DBSHORT=ON`) ■ for the definition of a field when a view is created; ■ when a `DEFINE DATA LOCAL` statement was specified; ■ when running under Natural Security.

	OFF	A database field may only be referenced via its long name. Every database field identifier is considered as a long-name reference, regardless of its length. If a two character name is supplied which can only be found as a short name but not as a long name, syntax error NAT0981 is raised at compile time. This makes it possible to use long names defined in a DDM with 2-byte identifier length. This option is essential if the underlying database you access with this DDM is SQL (Db2) and table columns with a two character name exist. For all other database types (for example, Adabas), however, any attempt to define a long field with a 2-byte name length will be rejected at DDM generation. Moreover, if no short-name references are used (what can be enforced via DBSHORT=OFF), the program becomes independent of being compiled without Natural Security.
Default setting	ON	
Dynamic specification	yes	
Specification within session	yes	
Applicable statements	OPTIONS	
Applicable command	Either GLOBALS or DBSHORT option of COMPOPT	

Examples:

Assume the following data base field definition in the DDM EMPLOYEES:

Short Name	Long Name
AA	PERSONNEL-ID

Example 1:

```
OPTIONS DBSHORT=ON
READ EMPLOYEES
  DISPLAY AA      /* data base short name AA is allowed
END
```

Example 2:

```
OPTIONS DBSHORT=OFF
READ EMPLOYEES
  DISPLAY AA      /* syntax error NAT0981, because DBSHORT=OFF
END
```

Example 3:

```
OPTIONS DBSHORT=ON
DEFINE DATA LOCAL
1 V1 VIEW OF EMPLOYEES
  2  PERSONNEL-ID
END-DEFINE
READ V1 BY PERSONNEL-ID
  DISPLAY AA     /* syntax error NAT0981, because PERSONNEL-ID is defined in view;
                 /* (even if DBSHORT=ON)
END-READ
END
```

48 DBUPD - Database Updating

This Natural profile parameter 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. When compiling a program (`CHECK`, `CATALOG` or `STOW` command), a NAT0105 error message (`Database updating not permitted`) is issued if the program contains one of the following statements: `UPDATE`, `STORE`, `DELETE` or `INSERT`. A database update is not performed when a program with an `UPDATE`, `STORE` or `DELETE` statement executes. Instead, a NAT1010 warning message is issued during the next screen I/O. In addition, a database loop that contains an `UPDATE` or `DELETE` statement does not place the records in hold status (no `read with hold`).
Default setting	`ON`	
Dynamic specification	yes	
Specification within session	no	
Application programming interface	`USR1005N`	See *SYSEXT - Natural Application Programming Interfaces* in the *Utilities* documentation. * Recommended.
	`USR1042N` *	

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

49 DC - Character for Decimal Point Notation

This Natural profile and session parameter determines the character to be used as decimal separator, that is, a point or a comma.

Possible settings	any special character	The character assigned to DC will be in effect for all notations where a decimal separator is possible; that is, variables, constants and edit masks. The character specified with the DC parameter must not be the same as the one specified with the IA (input assign character) or ID (input delimiter character) parameter. In addition, we recommend that this character is not the same as the one specified with the CF (control character for terminal commands) or HI (help character) parameter.		
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.
Application programming interface	USR1005N	See *SYSEXT - Natural Application Programming Interfaces* in the *Utilities* documentation.		

Notes:

1. Within a Natural session, the profile parameter DC can be overridden by the session parameter DC.
2. If you change DC in your parameter file, you must resave the DDM you are using in your Natural program which stores a new *.NSD* file on disk.

3. Under Natural Security, the setting of this parameter can be overridden by the Session Parameters option of the *Library Profile*.

50 DD - Day Differential

This Natural profile parameter is used to adjust 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	`-10953` to `+10953`	Machine date is adjusted. Specification of "+" is optional.
	`0`	No adjustment is made.
Default setting	`0`	
Dynamic specification	yes	
Specification within session	no	
Application programming interface	`USR1005N`	See *SYSEXT - Natural Application Programming Interfaces* in the *Utilities* documentation.

See also the profile parameter `TD`.

51 DF - Date Format

With the DF session parameter, you determine the length of a date when converted into alphanumeric representation without an edit mask being specified.

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	
	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	

Notes:

1. The DF parameter is evaluated at compilation time.

2. The sequence of the day, month and year components and the delimiter characters used are determined by the profile parameter `DTFORM`.
3. When the value of a date field is converted into 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.
4. 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.
5. 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`.
6. See also *Date Format for Alphanumeric Representation - DF Parameter* in the *Programming Guide*.

52 DFOUT - Date Format for Output

This Natural profile and session parameter determines the format in which the settings of date variables are displayed by `INPUT`, `DISPLAY`, `PRINT` and `WRITE` statements.

Possible settings	`S`	Date variables are displayed with a 2-digit year component, and delimiters as determined by the profile parameter `DTFORM`. Example: *yy-mm-dd*	
	`I`	Date variables are displayed with a full 4-digit year component and no delimiters. Example: *yyyymmdd*	
Default setting	`S`		
Dynamic specification	yes		
Specification within session	yes	Applicable statements:	`SET GLOBALS`
		Applicable command:	`GLOBALS`
Application programming interface	`USR1005N`	See *SYSEXT - Natural Application Programming Interfaces* in the *Utilities* documentation.	

Notes:

1. Within a Natural session, the profile parameter `DFOUT` can be overridden by the session parameter `DFOUT`.
2. The *profile parameter* `DFOUT` is evaluated at runtime.
3. 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.

4. The sequence of the day, month and year components in the date settings is determined by the `DTFORM` profile parameter.
5. See also *Processing of Date Information* in the *Programming Guide*.

53 DFS - Specify RPC Client's Default Server Address

This Natural profile parameter can be used to define an RPC default server address by specifying up to 5 positional subparameters.

Possible settings	See *DFS Parameter Syntax*.	
Default setting	none	Subparameter defaults, see *DFS Parameter Syntax*.
Dynamic specification	yes	See below.
Specification within session	yes	At runtime, this value can be overwritten using the Natural application programming interface `USR2007N`.
Application programming interface	`USR2007N`	See *Application Programming Interfaces for Use with Natural RPC* in the *Natural RPC (Remote Procedure Call)* documentation and *SYSEXT - Natural Application Programming Interfaces* in the *Utilities* documentation.

Notes:

1. `DFS` is specified on the client side only.
2. `DFS` 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 Addresses* in the *Natural RPC (Remote Procedure Call)* documentation.

DFS Parameter Syntax

The parameter syntax is as follows:

DFS=(*server-name*,*server-node*,*logon-indicator*,*transport-protocol-name*,*service-directory-indicator*)

Where:

Syntax Element	Value	Explanation
server-name	1 - 32 characters	Valid server name. See also parameter SRVNAME. There is no default, the value must be specified.
server-node	1 - 192 characters	Node name. See also parameter SRVNODE. 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. On Windows platforms: Instead of specifying L, check the selection box.
	(blank)	Blank means that no server logon will be executed. If nothing is specified, this is the default.
transport-protocol-name	ACI	The transport protocol to be used. ACI is the only possible value and the default.
service-directory-indicator	SERVDIR	A service directory must be present before the DFS profile parameter is evaluated.
	NOSERVDIR	No service directory is used before the DFS profile parameter is evaluated; that is, a service directory needs not be available on the client side. If nothing is specified, SERVDIR is the default.

54 DFSTACK - Date Format for Stack

This Natural profile and session parameter determines the format in which the settings of date variables are placed on the stack via a STACK, RUN or FETCH statement.

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. 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. Example: *yyyymmdd*	
Default setting	S		
Dynamic specification	yes		
Specification within session	yes	Applicable statement:	SET GLOBALS
		Applicable command:	GLOBALS
Application programming interface	USR1005N	See *SYSEXT - Natural Application Programming Interfaces* in the *Utilities* documentation.	

Notes:

1. Within a Natural session, the profile parameter `DFSTACK` can be overridden by the session parameter `DFSTACK`.
2. The profile parameter `DFSTACK` does not apply to `STACK`, `RUN` or `FETCH` statements for which the session parameter `DF` is set.
3. See also *Processing of Date Information* in the *Programming Guide*.

55 DFTITLE - Output Format of Date in Standard Report Title

This Natural profile and session parameter 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).

Possible settings	`S`	The date is output with a 2-digit year component and delimiters. Example: *`yy-mm-dd`*	
	`L`	The date is output with a 4-digit year component and delimiters. Example: *`yyyy-mm-dd`*	
	`I`	The date is output with a 4-digit year component and no delimiters. Example: *`yyyymmdd`*	
Default setting	`S`		
Dynamic specification	yes		
Specification within session	yes	Applicable statement:	`SET GLOBALS`
		Applicable command:	`GLOBALS`
Application programming interface	`USR1005N`	See *SYSEXT Utility - Natural Application Programming Interfaces* in the *Utilities* documentation.	

Notes:

1. Within a Natural session, the profile parameter `DFTITLE` can be overridden by the session parameter `DFTITLE`.
2. `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.

3. It has no effect on a user-defined page title (as specified with a `WRITE TITLE` statement).
4. The sequence of the day, month and year components and the delimiter characters used are determined by the profile parameter `DTFORM`.
5. See also *Processing of Date Information* and *Date Format for Default Page Title - DFTITLE Parameter* in the *Programming Guide*.

56 DL - Display Length for Output

With this session parameter, you specify the display length for a field of format A or U. The default display length is the length of the field.

Possible settings	1 to *n*	*n* = value of LS (line size) parameter minus 1
Default setting	none	
Applicable statements	FORMAT	
	DISPLAY INPUT PRINT WRITE	Parameter may be specified at statement level and/or at element level.
Applicable command	none	

Example:

```
FORMAT DL=20
```

For further information and an example of the DL session parameter usage, see the following sections in the *Programming Guide*:

- *Parameters to Influence the Output of Fields*
- *Output Length - AL and NL Parameters*
- *Display Length for Output - DL Parameter*

57 DTFORM - Date Format

This Natural profile parameter 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		
Application programming interface	USR1005N	See *SYSEXT - Natural Application Programming Interfaces* in the *Utilities* documentation.	

Notes:

1. The first day of a week is assumed to be Monday - unless DTFORM=U is specified, in which case Sunday is used.
2. 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.
3. The output format of the date in a default report page title is also specified by the profile parameter DFTITLE.
4. See also *Processing of Date Information* and *Default Edit Mask for Date - DTFORM Parameter* in the *Programming Guide*.

58 DU - Dump Generation

This Natural profile and session parameter determines whether a disassembled object code dump is to be generated.

Possible settings	ON	When one of the system commands CHECK, STOW, CATALOG, RUN or CATALL is executed, a disassembled object code file is produced. This dump file is written into the directory which is defined in the Natural TMP directory option in the Natural Configuration Utility; see *Local Configuration File*, *Installation Assignments*. The name of this dump file consists of the source file name and the extension *.DIA*. If the source file has not been saved, the name of the dump file is *GEN.DIA*. If the program contains database access statements, dump files with the extension *.ADA* (for Adabas) or *.SQL* (for SQL databases) are also created. If XREF data are generated, a dump file *.XRF* is created. **Note:** DU=ON may create a large dump file (depending on the size of the source file), which can cause significant degradation in system performance.
	OFF	No dump file is generated.
Default setting	OFF	
Dynamic specification	yes	
Specification within session	yes	
Applicable statements	SET GLOBALS	
Applicable command	GLOBALS	

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

59 DY - Dynamic Attributes

- DY Parameter Syntax 134
- Examples 136

This session parameter is used to assign attributes for dynamic attribute field display.

Possible settings	See *DY Parameter Syntax*.	
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 applies, the current field presentation and color attributes remain in effect, unless new settings are defined in the DY entry. This means, the field color is only changed by a DY attribute if the DY parameter itself defines a new color. The same applies to the **field representation attributes**, such as (AD=B,C,D,I,N,U,V).

The following topics are covered below:

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 (*)

* The field representation attributes marked with an asterisk (*) require corresponding hardware features, and will be ignored at runtime if these features are not available.

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 eight 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)

60 DYNPARM - Control Use of Dynamic Parameters

- Settings 138

This Natural profile parameter enables/disables the use of Natural dynamic parameters.

Settings

Possible settings	ON	Dynamic parameters supplied during Natural startup are processed.
	OFF	Dynamic parameters supplied during Natural startup are not processed. **Note:** If DYNPARM is set to OFF in the Natural default parameter file NATPARM, no alternative user-defined parameter files can be used when starting Natural.
Default setting	ON	
Dynamic specification	no	
Specification within session	no	

See *Dynamic Assignment of Parameter Values* for additional information on the use of dynamic parameters.

61 ECHECK - Existence Check for Object Calling Statements

This Natural profile parameter is used to control Natural's compiler.

Possible settings	ON	The compiler checks for the existence of an object that is specified in an object calling statement, such as FETCH [RETURN/REPEAT], RUN [REPEAT], CALLNAT, PERFORM, INPUT USING MAP, PROCESS PAGE USING, function call and helproutine call. The existence check is based on a search for the cataloged object or for the source of the object when it is invoked by a RUN [REPEAT] statement. It requires that the name of the object to be called/run is defined as an alphanumeric constant (not as an alphanumeric variable). Otherwise, ECHECK=ON will have no effect. **Problems in Using the CATALL Command with ECHECK=ON** When a CATALL system command is used in conjunction with ECHECK=ON, you should consider the following: If a CATALL process is invoked, the order in which the objects are compiled depends primarily on the type of the object and secondarily on the alphabetical name of the object. The object type sequence used is: GDAs, LDAs, PDAs, functions, subprograms, external subroutines, help routines, maps, adapters, programs, classes. Within objects of the same type, the alphabetical order of the name determines the sequence in which they are cataloged. As mentioned above, the success of the object calling statement is checked against the compiled form of the called object. If the calling object (the one which is being compiled and includes the object calling statement) is cataloged before the invoked object, the ECHECK result may be wrong if the called object was not cataloged beforehand. In this case, the object image of the called object has not yet been produced by the CATALL command.

		Solution: ■ Set compiler option ECHECK to OFF. ■ Perform a general compile with CATALL on the complete library, or if just one or a few objects were changed, perform a separate compile on these objects. ■ Set compiler option ECHECK=ON. ■ On the complete library, perform a general compile with CATALL, selecting function CHECK.
	OFF	No existence check is performed.
Default setting	OFF	
Dynamic specification	yes	
Specification within session	yes	
Applicable statements	OPTIONS	
Applicable command	ECHECK	Option of system command COMPOPT

62 ECHO - Control Printing of Batch Input Data

This Natural profile parameter is used to enable or disable the printing of input data 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	

Notes:

1. This Natural profile parameter only applies in batch mode.
2. 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 `%*`.
3. Input read from `CMSYNIN` in command (`NEXT`) mode is echoed to the batch output file `CMPRINT` always.

63 EDTBPSIZE - Software AG Editor Buffer Pool Size

This Natural profile parameter is used to set the size of the Software AG Editor buffer pool.

Possible settings	0 - 4000	Size of the Software AG Editor buffer pool in KB.
Default setting	400	
Dynamic specification	no	
Specification within session	no	

64 EDTLFILES - Number of Software AG Editor Logical Files

This Natural profile parameter is used to set 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	

65 EJ - Page Eject

This Natural profile and session parameter is used to 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.

Possible settings	`ON`	A page eject is performed.		
	`OFF`	No page eject is performed. **Note:** 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 statement:	`SET GLOBALS`	Parameter is evaluated at runtime.
		Applicable command:	`GLOBALS`	Parameter may be specified dynamically with the `GLOBALS` system command.
Application programming interface	`USR1005N`	See *SYSEXT - Natural Application Programming Interfaces* in the *Utilities* documentation.		

Notes:

1. Within a Natural session, the profile parameter `EJ` can be overridden by the session parameter `EJ`.
2. The `EJ` setting can in turn be overridden by an `EJECT` statement.
3. This parameter only applies to the first report (Report 0). For additional reports, the statement `EJECT` with report specification (*rep*) has to be used.

4. Under Natural Security, the setting of this parameter can be overridden by the Session Parameters option of the Library Profile.

66 EM - Edit Mask

- EM Parameter Syntax 150
- Examples 151
- Blanks in Edit Masks 151
- Default Edit Masks 151
- Edit Masks for Numeric Fields 152
- Edit Masks for Alphanumeric Fields 155
- Edit Masks for Binary Fields - Format B 157
- Hexadecimal Edit Masks 157
- Edit Masks for Date and Time Fields - Formats D and T 159
- Edit Masks for Logical Fields - Format L 163
- Edit Masks with MOVE EDITED (Edit Mask Specified with operand2) 164

With this session parameter, you can specify an edit mask for an input and/or output field that is used in one of the statements listed in the following table under *Applicable statements*.

Possible settings	See ***EM Parameter Syntax***.	
Default setting	none	
Applicable statements	FORMAT	Parameter may be specified dynamically with the FORMAT statement.
	DEFINE DATA DISPLAY INPUT PRINT PROCESS PAGE/PROCESS PAGE UPDATE WRITE	Parameter may be specified at statement level and/or at element level.
	COMPRESS MOVE EDITED WRITE WORK FILE	Parameter may be specified at element level.
Applicable command	none	

Notes:

1. For information on Unicode edit masks, see session parameter EMU.
2. The parameter EM can also be used with U format fields. For information on Unicode format, see *Unicode and Code Page Support in the Natural Programming Language*, *Session Parameters*, EMU, ICU, LCU, TCU versus EM, IC, LC, TC.
3. See also *Edit Masks - EM Parameter* in the *Programming Guide*.

The following topics are covered below:

EM Parameter 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)
```

You may replace a sequence of the same significant characters with a numeric notation, such as x(8) for xxxxxxxx. The following examples demonstrate the abbreviated notation which may be used for the significant characters of numeric (Z,9), hexadecimal (H), alphanumeric (X) and date (N,L) 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
EM=YYYY-L(8)-DD-N(8) is equivalent to: EM=YYYY-LLLLLLLL-DD-NNNNNNNN
```

Blanks in Edit Masks

Blanks behind the equal sign (=) of the EM parameter are not allowed (for example: EM=<blank>XXX).

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 separator and/or low-order digits after the decimal separator will be truncated.

The following topics are covered below:

- Characters for the Definition of Numeric Edit Masks
- Sign Characters
- Literal Leading Characters
- Literal Insertion and Trailing Characters
- Trailing Sign Characters
- Examples of Numeric Edit Masks

Characters for the Definition of Numeric Edit Masks

Character	Function
9	Position to be displayed (one digit of the field value).
. (period)	The first period inserted is used as a decimal separator. Subsequent periods are treated as literal characters. **Note:** At this point, the period represents the sign currently defined as decimal separator character. If another character is chosen (for example, a comma) with the session or profile parameter DC, this character is to be used instead.
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 separator character. A zero value may be displayed as blanks using all Zs in the edit mask (see also 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 format N 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=*EURZZ9.9	EUR**0.0	EUR*54.	EUR*87.	EUR962.	EUR830.
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 closing 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 of an alphanumeric edit mask, the first of these leading characters will not be displayed, but is used as filler character and replaces all leading blanks in the alphanumeric output field.

Example:

```
DEFINE DATA LOCAL
1 #X (A4)  INIT <'  34'>
END-DEFINE
WRITE #X (EM=*A:X:)
   6X #X (EM=*A:XX:)
   6X #X (EM=*A:XXX:)
   6X #X (EM=*A:XXXX:)
   6X #X (EM=1234XXXX5678)
END
```

Output Produced:

```
A:*:      A:**:      A:**3:      A:**34:      23411345678
```

Trailing characters which immediately follow the last permissible print position will be displayed.

If the number of positions specified with the mask is smaller than the field length, the overhanging field content is not displayed.

If the number of positions specified with the mask is higher than the field length, the mask is truncated on the first overhanging position.

Example:

```
DEFINE DATA LOCAL
1 #TEXT  (A4) INIT <'BLUE'>
END-DEFINE
WRITE #TEXT (EM=X-X-X)        /* 'B-L-U', 3 bytes of field only.
WRITE #TEXT (EM=X-X-X-X-X)    /* 'B-L-U-E-', with truncated mask.
END
```

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': Edit mask
************************************************************************
DEFINE DATA LOCAL
1 #TEXT  (A4)
END-DEFINE
*
ASSIGN #TEXT = 'BLUE'
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
```

Output of Program `EMMASK1`:

```
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 separator 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

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

Example Program Using Hexadecimal Edit Masks:

```
** Example 'EMMASK2': Edit mask
************************************************************************
DEFINE DATA LOCAL
1 #TEXT1 (A2)
1 #TEXT2 (N2)
END-DEFINE
*
ASSIGN #TEXT1 = 'AB'
ASSIGN #TEXT2 =  10
*
WRITE NOTITLE
        'MASK (EM=HH)  :' 18T #TEXT1 (EM=HH)    30T #TEXT2 (EM=HH)
      / 'MASK (EM=H^H) :' 18T #TEXT1 (EM=H^H)   30T #TEXT2 (EM=H^H)
      / 'MASK (EM=HH^H):' 18T #TEXT1 (EM=HH^H)  30T #TEXT2 (EM=HH^H)
      / 'MASK (EM=H-H) :' 18T #TEXT1 (EM=H-H)   30T #TEXT2 (EM=H-H)
      / 'MASK (EM=H)   :' 18T #TEXT1 (EM=H)     30T #TEXT2 (EM=H)
END
```

Output of Program `EMMASK2` (ASCII):

```
MASK (EM=HH)  :  4142        3130
MASK (EM=H^H) :  41 42       31 30
MASK (EM=HH^H):  4142        3130
MASK (EM=H-H) :  41-42       31-30
MASK (EM=H)   :  41          31
```

Output of Program `EMMASK2` (EBCDIC):

```
MASK (EM=HH)  :  C1C2        F1F0
MASK (EM=H^H) :  C1 C2       F1 F0
MASK (EM=HH^H):  C1C2        F1F0
MASK (EM=H-H) :  C1-C2       F1-F0
MASK (EM=H)   :  C1          F1
```

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 characters described in the following sections can be specified.

- Date - Format D, and Time - Format T
- Syntactical Restrictions for Date Characters
- Hints for Input Edit Mask
- Hints for Week Display (WW or ZW) in Output Edit Mask
- Time - Format T - only
- Examples of Date and Time Edit Masks

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.

Character	Usage
NN... or N(*n*)	Name of day (language-dependent). The maximum length is determined by the number of Ns 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 characters 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

The range of valid year values (YYYY) is 1582 - 2699.

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, refer to profile parameter YSLW in the *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)
END
```

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 mask for date and time variables
************************************************************************
*
WRITE NOTITLE
  'DATE INTERNAL :' *DATX (DF=L) /
  '              :' *DATX (EM=N(9)' 'ZW.'WEEK 'YYYY)  /
  '              :' *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
```

Output of Program `EMDATI`:

```
DATE INTERNAL : 2005-01-12
              : Wednesday  2.WEEK 2005
              :  12.DAY 2005
     ROMAN    : MMV
     AMERICAN : 01/12/2005           OR   01/12/2005
     JULIAN   : 2005012              OR   2005012
     GREGORIAN: 12.January2005       OR   12January  2005


TIME INTERNAL : 16:04:14             OR   16:04:14.8
              : 16.04.14.8
```

```
               : 04.04.14 PM
               : 16
```

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': Edit mask for logical variables
************************************************************************
DEFINE DATA LOCAL
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
```

Output of Program EMLOGV:

```
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
```

Edit Masks with MOVE EDITED (Edit Mask Specified with operand2)

In a `MOVE EDITED` statement, the edit mask specifies the location of the significant input data for *operand2* within the contents of *operand1*.

If *operand1* contains extra characters in the input contents, the edit mask must include matching characters to disregard them.

These additional characters can also be ignored by including the wildcard character `?` in the edit mask.

If the edit mask does not include the additional characters or the wildcard `?` where extra input characters occur, the `MOVE EDITED` operation will not interpret the input correctly.

Note: To disregard the first character of the input, you must specify a filler character at the beginning of the edit mask for both alphanumeric and numeric edit masks to ensure the expected results.

See also *Statements > MOVE > Example 6 - MOVE EDITED with Edit Mask and Extra Characters* and *Example 7 - MOVE EDITED with Edit Mask and Wildcard Character '?'*.

67 EMFM - Edit Mask Free Mode

This Natural profile parameter is used to activate/deactivate the Edit Mask Free mode at session startup.

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-`.

Notes:

1. The Edit Mask Free mode allows you to omit literals during input into a field with a numeric edit mask.
2. For additional information, see *Numeric Edit Mask Free Mode* in the `INPUT` statement description in the *Statements* documentation.

68 EMU - Unicode Edit Mask

With this session parameter, you can specify a Unicode edit mask for an input and/or output field that is used in one of the statements listed in the following table under *Applicable statements*.

Possible settings	The syntax of the session parameter `EMU` is identical to that of the session parameter `EM` (see ***EM Parameter Syntax***). **Note:** See also *Unicode Edit Masks - EMU Parameter* in the *Programming Guide*.	
Default setting	none	

Applicable statements	`DEFINE DATA`
	`DISPLAY`
	`INPUT`
	`PRINT`
	`WRITE`
	`PROCESS PAGE`
		Parameter may be specified at statement level and/or at element level.

	`COMPRESS`
	`MOVE EDITED`
	`WRITE WORK FILE`
		Parameter may be specified at element level.

Applicable command	none	

Notes:

1. Edit masks which are defined with `EMU` are kept in Unicode format so that the content is independent of the installed system code page.
2. For further information and an example, see also *Unicode and Code Page Support in the Natural Programming Language*, *Session Parameters*, section *EMU, ICU, LCU, TCU versus EM, IC, LC, TC*.

69 ENDIAN - Endian Mode for Compiled Objects

This Natural profile and session parameter specifies the architecture for which the compiler should generate GP. See also *Portable Natural Generated Programs* in the *Programming Guide*.

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`	

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

70 ENDMSG - Display Session-End Message

This Natural profile parameter is used to suppress the display the default message NAT9995 that is displayed at the end of the Natural session to indicate that the Natural session has been ended normally.

Possible settings	ON	Message NAT9995, NAT9978 or NAT9987 is written to the batch output file CMPRINT at the end of the session.
	OFF	Message NAT9995, NAT9978 or NAT9987 does not appear in CMPRINT.
Default setting	ON	
Dynamic specification	yes	
Specification within session	no	

Notes:

1. If Natural terminates with a startup error, then Natural message NAT9987 will be used instead of NAT9995.
2. If Natural terminates with a runtime error, then Natural message NAT9978 will be used instead of NAT9995.

71 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. **Note:** 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	Empty line suppression is disabled.
Default setting	OFF	
Specification within session	yes	
Applicable statements	FORMAT	
	DISPLAY WRITE	Parameter may be specified at statement level and/or at element level.
Applicable command	none	

Notes:

1. 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.
2. See also *Parameters to Influence the Output of Fields* in the *Programming Guide*.

Example:

```
DISPLAY (ES=ON) NAME CITY
```

72 ESCAPE - Ignore Terminal Commands %% and %.

This Natural profile parameter can be used to disable the terminal commands `%%` and `%.`.

Possible settings	`ON`	Enables the use of terminal commands `%%` and `%.`.
	`OFF`	The terminal commands `%%` 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	

73 ESXDB - Database ID Used for Entire System Server DDMs

This Natural profile parameter specifies the database ID used for Entire System Server's DDMs.

Possible settings	`1 - 254`	Database ID. To activate this parameter, a database ID in the range of 1 to 254 must be specified.
	`0`	With `ESXDB=0`, the Entire System Server Interface is not active.
Default setting	`0`	
Dynamic specification	no	
Specification within session	no	

Notes:

1. This Natural profile parameter applies to the Entire System Server Interface.
2. 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, refer to *Setting up the Entire System Server Interface* in the *Operations* documentation.

74 ET - Execution of END/BACKOUT TRANSACTION Statements

This Natural profile parameter 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.

75 ETA - Error Transaction Program

This Natural profile parameter 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='  '`, no error transaction program is called.
Default setting	`'  '` (blank)	
Dynamic specification	yes	
Specification within session	yes	
Application programming interface	`USR1041N`	`USR1041N`is a sample error transaction program delivered in source form. See *SYSEXT - Natural Application Programming Interfaces* in the *Utilities* documentation.

Notes:

1. The setting of this parameter can be modified by a user program by way of assignment to the system variable `*ERROR-TA` or, if Natural Security is installed, within the Natural Security library profile; see *Components of a Library Profile* in the *Natural Security* documentation.
2. For further information, see *Using an Error Transaction Program* in the *Programming Guide*.

76 ETDB - Database for Transaction Data

This Natural profile parameter 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. **Note:** Database ID `255` is reserved for logical system files, see profile parameter `LFILE`.
	`0`	The transaction data is written to the database specified with the profile parameter `UDB`.
Default setting	`0`	
Dynamic specification	yes	
Specification within session	no	

Other transaction processing related parameters: `ET` | `ETEOP` | `ETIO`

77 ETEOP - Issue END TRANSACTION at End of Program

This Natural profile parameter 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	

78 ETID - Adabas User Identification

This Natural profile parameter 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 - 8 characters	This setting is used as the user ID setting in an Adabas open call. **Note:** The Adabas user ID has its own syntax. Consult your Adabas *Command Reference* documentation if you want to use special characters for the setting of `ETID`.
	`OFF`	The `ETID` is set to `'  '` (blanks), but Natural Security is allowed to set the `ETID`.
	`'  '` (blank)	The `ETID` is set to `'  '` (blanks). This value is passed to Adabas on an open call without being modified by Natural Security.
	`$$`	The `ETID` is replaced by the process ID.
Default setting	`'  '` (blank)	
Dynamic specification	yes	
Specification within session	no	

Notes:

1. If the `ETID` is not specified neither in `NATPARM` nor dynamically, Natural uses the setting of `*INIT-USER` to fill the `ETID`.
2. $$ will be evaluated during the start-up of Natural.
3. The `ETID` might also be used by Natural for work-file name generation. This might cause problems when running multiple instances on one machine or again, if special characters are used for the setting of `ETID`. Refer to the Natural parameter `TMPSORTUNIQ` for a solution.

79 ETIO - Issue END TRANSACTION upon Terminal I/O

This Natural profile parameter 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. **Note:** Natural add-on products (except for Natural Security) 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	

80 FC - Filler Character for INPUT Statement

This Natural profile parameter is used to specify the default filler character to be used for fields displayed by an `INPUT` statement.

Possible settings	any character	Default filler character.
Default setting	blank	
Dynamic specification	yes	
Specification within session	no	

Notes:

1. The default filler character is used to pre-fill non-protected input fields (field attribute specification `AD=A`) when fields are written to a terminal by an `INPUT` statement.
2. For modifiable input fields (field attribute specification `AD=M`), it is used to fill the rest of the field.

81 FC - Filler Character for DISPLAY Statement

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.

Possible settings	any character	Filler character for individual headings.
Default setting	blank	
Specification within session	yes	
Applicable statements	DISPLAY FORMAT	
Applicable command	none	

Notes:

1. 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.
2. Unlike the GC parameter, which applies to headings across a group of columns, the FC parameter applies to individual columns.

Example:

```
DISPLAY (FC=*)
```

82 FCDP - Filler Character for Dynamically Protected Input Fields

This Natural profile and session parameter can be used 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).

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`	
Application programming interface	`USR1005N`	See *SYSEXT - Natural Application Programming Interfaces* in the *Utilities* documentation.

Notes:

1. Depending on the setting of the `FCDP` parameter, dynamically protected input fields are displayed filled either with blanks or with the defined filler characters.
2. Within a Natural session, the profile parameter `FCDP` can be overridden by the session parameter `FCDP`.

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 _____
```

83 FDDM - Natural System File for DDMs

This Natural profile parameter is used to specify five subparameters for the Natural system file for DDMs.

Possible settings	See *FDDM Parameter Syntax*.	
Default setting	none	
Dynamic specification	yes	
Specification within session	no	

Notes:

1. 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 z/OS, where all DDMs are stored in the system file `FDIC`.
2. 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.
3. For information on system files, refer to: System Files in the Natural Operations documentation.

FDDM Parameter Syntax

The parameter syntax is as follows:

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

Where:

Syntax Element	Value	Explanation
database-ID	1 - 65535	Database identification of the database in which the Natural system file for DDMs is located. **Note:** Database ID 255 is reserved for logical system files, see Natural profile parameter LFILE.
file-number	1 - 65535	File number of the database file in which the Natural system file for DDMs is located.
password	1 - 8 characters	The password is only required if the Natural user-program system file has been password-protected using the Adabas security feature. **Note:** The password is reserved for future use; currently, it is ignored.
cipher-key	1 - 8 numeric characters	The cipher key is only required if the Natural user-program system file has been ciphered using the Adabas security feature. **Note:** The cipher key is reserved for future use; currently, it is ignored.
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.

Examples:

```
FDDM=(22,5)
FDDM=(22,5,,12345)
FDDM=(22,5,,,RO)
```

84 FDIC - Predict System File

This Natural profile parameter defines the database ID, file number, password and cipher key for the Predict system file (FDIC), which Predict uses to retrieve and/or store data.

Possible settings	See *FDIC Parameter Syntax*.	
Default setting	none	
Dynamic specification	yes	
Specification within session	no	

Notes:

1. In a remote development environment, a Development Server File is used instead, see the *SPoD - Natural's Single Point of Development* and the *Natural Development Server* documentation.
2. For information on system files, refer to *System Files* in the *Operations* documentation.

FDIC Parameter Syntax

The parameter syntax is as follows:

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

Where:

Syntax Element	Value	Explanation
database-ID	1 - 65535, except 255	Database identification of the database in which the Predict system file is located. **Note:** 1. Database ID 255 is reserved for logical system files, see Natural profile parameter LFILE.

Syntax Element	Value	Explanation
		2. If no `FDIC` is available, do not enter anything in the DBID field.
file-number	`1 - 65535`	File number of the database file in which the Predict system file is located. **Note:** If no `FDIC` is available, do not enter anything in the DBID field.
password	1 - 8 characters	Password for the Predict system file. **Note:** 1. A password is only required if the Predict system file has been password-protected using the Adabas security feature. 2. The password feature is reserved for future use; currently, it is ignored.
cipher-key	1 - 8 numeric characters	Cipher key for the Predict system file. **Note:** 1. A cipher key is only required if the Predict system file has been ciphered using the Adabas security feature. 2. The cipher key feature is reserved for future use; currently, it is ignored.
`RO`	-	Read only option - not supported on this platform.

Examples:

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

85 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 - 16`	Mantissa length. **Note:** 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
```

86 FNAT - Natural System File for System Programs

This Natural profile parameter defines the database ID, file number, password, cipher key and read-only flag for the Natural system file for Natural system programs (FNAT).

Possible settings	See *FNAT Parameter Syntax*.	
Default setting	none	
Dynamic specification	yes	
Specification within session	no	

Notes:

1. The Natural system file is the file from which all Natural system programs are retrieved and upon which all system commands operate. Error texts and Natural help information related to the Natural system libraries are also contained in this system file.
2. For information on system files, refer to *System Files* in the *Operations* documentation.

FNAT Parameter Syntax

The parameter syntax is as follows:

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

Where:

Syntax Element	Value	Explanation
database-ID	1-254	Database identification of the database in which the Natural system file is located. **Note:** Database ID 255 is reserved for logical system files, see Natural profile parameter LFILE.
file-number	1 - 255	File number of the database file in which the Natural system file is located.
password	1 - 8 characters	Password for the Natural system file. **Note:** 1. A password is only required if the Natural system file has been password-protected using the Adabas security feature. 2. The password feature is reserved for future use; currently, it is ignored.
cipher-key	1 - 8 numeric characters	Cipher key for the Natural system file. **Note:** 1. A cipher key is only required if the Natural system file has been ciphered using the Adabas security feature. 2. The cipher key feature is reserved for future use; currently, it is ignored.
RO	-	Read-only option. **Note:** 1. RO indicates that the Natural system file is “read-only”. 2. RO is only specified if modifications on the file are to be disabled.

Examples:

```
FNAT=(,102)
FNAT=(99,102,,,RO)
FNAT=(99,102,PASSW2)
```

87 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 *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. **Note:** This behavior corresponds to the previous situation when the utility was invoked using the system command `LOGON library-name`.
	`OFF`	The current user GDA and AIV variables are preserved when a utility is started. **Note:** 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	

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

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

Note: See also *Format and Length of User-Defined Variables* in the *Programming Guide*.

Possible settings	ON	No default format/length is assigned by Natural for a newly introduced variable in reporting mode. **Note:** 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	
Application programming interface	USR1005N	See *SYSEXT - Natural Application Programming Interfaces* in the *Utilities* documentation.

Notes:

1. This Natural profile and session parameter only applies to reporting mode; it has no effect in structured mode.
2. Within a Natural session, the profile parameter FS can be overridden by the session parameter FS.

3. Under Natural Security, the setting of this parameter can be overridden by the Session Parameters option of the Library Profile.

89 FSEC - Natural Security System File

This Natural profile parameter defines the database ID, file number, password, and cipher key for the Natural Security system file (FSEC), which is used by Natural Security to retrieve/store its security information.

Possible settings	See *FSEC Parameter Syntax*.	
Default setting	none	
Dynamic specification	yes	
Specification within session	no	

Notes:

1. This Natural profile parameter only applies if Natural Security is used.
2. For information on system files, refer to *System Files* in the *Operations* documentation.

FSEC Parameter Syntax

The `FSEC` parameter syntax is as follows:

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

Where:

Syntax Element	Value	Explanation
database-ID	`1-65535`, except `255`	Database identification of the database in which the Natural Security system file is located. **Note:** 1. Database ID `255` is reserved for logical system files, see Natural profile parameter `LFILE`.

Syntax Element	Value	Explanation
		2. If no FSEC system file is available, do not enter anything in the file number field.
file-number	1-65535	File number of the database file in which the Natural Security system file is located. **Note:** If no FSEC system file is available, do not enter anything in the file number field.
password	1 - 8 characters	Password for the Natural Security system file. **Note:** 1. A password is only required if the Natural Security system file has been password-protected using the Adabas security feature. 2. The password feature is reserved for future use; currently, it is ignored.
cipher-key	1 - 8 numeric characters	Cipher key for the Natural Security system file.
RO	-	Read-only option. **Note:** The RO option is not supported on this platform.

Examples:

```
FSEC=(10,8)
FSEC=10,5,PASSW1,12345678
FSEC=1,200,,12345678
```

90 FUSER - Natural System File for User Programs

This Natural profile parameter defines the database ID, file number, password, and cipher key for the Natural user-program system file (FUSER).

Possible settings	See ***FUSER Parameter Syntax***.	
Default setting	none	
Dynamic specification	yes	
Specification within session	no	

Notes:

1. The Natural user-program system file (FUSER) is the database file from which all user-written Natural programs are retrieved.
2. For information on system files, refer to: *System Files* in the *Operations* documentation.

FUSER Parameter Syntax

The FUSER parameter syntax is as follows:

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

Where:

Syntax Element	Value	Explanation
database-ID	1-254	Database identification of the database in which the Natural user-program system file is located. **Note:** Database ID 255 is reserved for logical system files, see Natural profile parameter LFILE.

Syntax Element	Value	Explanation
file-number	1-255	File number of the database file in which the Natural user-program system file is located.
password	1 to 8 characters	Password for the Natural user-program system file. **Note:** 1. A password is only required if the Natural user-program system file has been password-protected using the Adabas security feature. 2. The password feature is reserved for future use; currently, it is ignored.
cipher-key	1 to 8 numeric characters	Cipher key for the Natural user-program system file. **Note:** 1. A cipher key is only required if the Natural user-program system file has been ciphered using the Adabas security feature. 2. The cipher key feature is reserved for future use; currently, it is ignored.
RO	-	Read-only option. **Note:** 1. RO indicates that the Natural user-program system file is “read-only”. 2. RO is only specified if modifications on the Natural user-program system file are to be disabled.

Examples:

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

91 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.

Possible settings	any character	Filler character for group headers.
Default setting	blank	
Specification within session	yes	
Applicable statements	`DISPLAY` `FORMAT`	
Applicable command	none	

Note: Unlike the `FC` parameter, which applies to individual columns, the `GC` parameter applies to headings across a group of columns.

Example:

```
DISPLAY (GC=*)
```

92 GFID - Global Format IDs

This Natural profile and session parameter is used 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	yes	
Applicable statements	`OPTIONS`	
Applicable Commands:	Either `GLOBALS` or `GFID` option of `COMPOPT`.	

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

93 GPGEN - Generate GP Information

- GPGEN Parameter Syntax 218
- Examples of GPGEN Parameter 219

This profile parameter is used to enable/disable the generation of GP information which is used when debugging or profiling applications or when using the code coverage feature.

Possible settings	See *GPGEN Parameter Syntax*	
Default setting	none	See the default settings of the subparameters in *GPGEN Parameter Syntax*.
Dynamic specification	yes	
Specification within session	no	

GPGEN Parameter Syntax

The `GPGEN` parameter is specified as follows:

GPGEN={(*subparameter=value*[,*subparameter=value*]...)}

Important: Blank spaces are not allowed in the syntax. Use commas to separate the syntax elements.

Where:

Subparameter	Value	Description
`DEBUGGER`	`DEBUGGER=`*`value`* determines whether GP information for the Debugger is generated. Default: `OFF`	
	`ON`	GP information for the Debugger is generated.
	`OFF`	GP information for the Debugger is not generated.
`PROFILER`	`PROFILER=`*`value`* determines whether GP information for the Natural Profiler is generated. Default: `OFF`	
	`ON`	GP information for the Profiler is generated.
	`OFF`	GP information for the Profiler is not generated.
`COVERAGE`	`COVERAGE=`*`value`* determines whether GP information for **code coverage** is generated. Default: `OFF`	
	`ON`	GP information for code coverage is generated.
	`OFF`	GP information for code coverage is not generated.

Examples of GPGEN Parameter

```
GPGEN=(DEBUGGER=ON,PROFILER=ON,COVERAGE=ON)
```

94 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)
```

95 HD - Header Definition

With this session parameter, you define which default text is to be used when

- the field is output with a `DISPLAY` statement;
- an equal sign (=) is placed immediately before the field in a `WRITE` or `INPUT` statement.

Possible settings	'*text*'	120 alphanumeric or Unicode characters at maximum.
Default setting	none	
Applicable statements	`DEFINE DATA`	Parameter may be specified at field level and/or element level.
Applicable command	none	

96 HE - Helproutine

- HE Parameter Syntax 226
- Execution of Helproutines 228
- Examples 228

With this session parameter, you assign a helproutine or a help map to a field.

Possible settings		See *HE Parameter Syntax* 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 or help map may then be invoked during processing of an `INPUT` statement or a map by choosing either of the following methods:

- In the field for which to invoke the help request, enter the help character in the leftmost position of the field and press ENTER. The default help character is a question mark (?).

  If you enter the help character at a different position of the field or if you enter more than one character, the string is taken as user input and no help is invoked. If the field contains hexadecimal zeroes, it depends on the terminal emulation whether Natural can interpret the values as a help request.
- Or:

  Place the cursor in the field for which to invoke the help request and press the PF key defined as help function key with the `SET KEY` statement.

The following topics are covered below:

HE Parameter Syntax

The syntax of this parameter is:

```
HE=operand1 [ , { operand2 } ] ...20
                { =        }
                { nX       }
```

Operand Definition Table:

Operand	Possible Structure						Possible Formats													Referencing Permitted	Dynamic Definition
operand1	C	S					A													no	no
operand2	C	S	A				A	U	N	P	I	F	B	D	T	L	C	G	O	no	no

Syntax Element Description:

Syntax Element	Description
operand1	*operand1* is the name of the helproutine or help 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 one-character code corresponding to the current value of the Natural system variable `*LANGUAGE`. This feature allows the use of multi-lingual helproutines or help maps.
operand2	You may specify 1 to 20 parameters (*operand2*) which are passed to the helproutine or help map. They may be specified as constants or as user-defined variables which contain the values of the parameters.
=	The equals sign (=) is used to pass an object or a field name to a helproutine or help map: ■ If the equals sign is entered in the `HE=` specification at statement level, the name of the object (as contained in the system variable `*PROGRAM`) being executed is passed to the helproutine or help map. In **Example 3**, the object name passed is `PROGRAM1`. ■ If the equals sign is entered in the `HE=` specification at field level, the name of the field is passed to the helproutine or help map. In **Example 3**, the field name passed is `#PARM1`. If the equals sign is used as a parameter, the corresponding parameter in the helproutine or help map must be specified with format/length A65.
*n*X	The notation *n*`X` can be used to specify parameters to be omitted, that is, for which no values are to be passed. The corresponding receiving parameters in the called helproutine's `DEFINE DATA PARAMETER` statement must be defined as `OPTIONAL`.

Notes:

1. The operands must be separated either by the input delimiter character (as specified with the session parameter `ID`) or by a comma. However, a comma must not be used for this purpose if the comma is defined as decimal character (with the session parameter `DC`).
2. If parameters are specified, the helproutine must begin with a `DEFINE DATA PARAMETER` statement which defines fields that correspond with the parameters in format and length.
3. 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.

4. 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 I (integer), N (numeric unpacked), or P (packed numeric) 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 or help map 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
...
```

Example 3:

```
* Program 'PROGRAM1'
*
DEFINE DATA LOCAL
1 #PARM1 (A65) INIT <'valueparm1'>
END-DEFINE
SET KEY PF1 = HELP
FORMAT KD=ON
*
INPUT (AD=M HE='HELP1',=)
  'Enter ? for name of executed object:'
  / #PARM1
*
INPUT (AD=M)
  'Enter ? for field name:'
  / #PARM1 (HE='HELP1',=)
*
END
```

Parameter Data Area in Example Helproutine `HELP1`:

```
* Helproutine 'HELP1'
*
DEFINE DATA PARAMETER
1 #FLD1 (A65)
END-DEFINE
...
```

97 HI - Help Character

This Natural profile parameter 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. **Note:** The character specified with the `HI` parameter must not be the same as the one specified with the `CF` (control character for z/OS terminal commands) parameter. In addition, we recommend that this character is not the same as the one specified with the `DC` (decimal character), `IA` (input assign character) or `ID` (input delimiter character) parameter.
	blank	**Note:** 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	
Application programming interface	`USR0350N`	See *SYSEXT - Natural Application Programming Interfaces* in the *Utilities* documentation.

98 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. **Note:** 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. **Note:** 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)
```

99 IA - Input Assign Character

This Natural profile and session parameter 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.

Possible settings	any special character	Assignment character for the input parameter processing in `INPUT` statements.
Default setting	=	Equals sign.
Dynamic specification	yes	
Specification within session	yes	
Applicable statements	`SET GLOBALS`	
Applicable command	`GLOBALS`	
Application programming interface	`USR1005N`	See *SYSEXT - Natural Application Programming Interfaces* in the *Utilities* documentation.

Notes:

1. The character specified with the `IA` parameter must not be the same as the character specified with the `DC` (decimal character) or `ID` (input delimiter character) parameter. In addition, we recommend that this character is not the same as the one specified with the `CF` (control character for z/OS terminal commands) or `HI` (help character) parameter.
2. Within a Natural session, the profile parameter `IA` can be overridden by the session parameter `IA`.
3. Under Natural Security, the setting of this parameter can be overridden by the *Session Parameters* option of the Library Profile.

Example:

In the following example, it is assumed that, for the beginning, the default input assign character (=) applies.

```
** Example 'IACHAR': Input Assign character
************************************************************************
DEFINE DATA LOCAL
1 #A (A1)
1 #B (A1)
END-DEFINE
*
INPUT #A #B
*
WRITE 'Field #A:' #A  / 'Field #B:' #B
*
END
```

1. Enter the command

   ```
   IACHAR #A=Y,#B=X
   ```

   The program produces the following output:

   ```
   Page      1                                                    05-01-19  11:05:51

   Field #A: Y
   Field #B: X
   ```

2. Enter the command

   ```
   GLOBALS IA=:
   ```

   This sets the input assign character to colon (:).

3. Then enter the command

   ```
   IACHAR #B:X,#A:Y
   ```

   The program produces the following output:

   ```
   Page      1                                                    06-11-13  12:12:24

   Field #A: Y
   Field #B: X
   ```

100 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 increased accordingly.

Possible settings	any character	Character string to be inserted. You can specify a string of one to ten characters. **Note:** 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 a circumflex accent (^).
Default setting	none	
Specification within session	yes	
Applicable statements	`FORMAT`	Parameter may be specified dynamically with the `FORMAT` statement.
	`DISPLAY`	Parameter may be specified at statement level and/or at element level.
Applicable command	none	

Notes:

1. The insertion character is inserted between leading spaces and the field value whereas the leading character is output in front of the leading space.
2. For numeric values, the insertion characters will be placed before the first significant digit printed.
3. The `IC` and `LC` parameters are mutually exclusive.
4. The parameter `IC` can also be used with U format fields.

5. For information on Unicode format, see also *Unicode and Code Page Support in the Natural Programming Language*, *Session Parameters*, *EMU, ICU, LCU, TCU versus EM, IC, LC, TC* in the *Unicode and Code Page Support* documentation.
6. The difference between the session parameters `LC`, `LCU` and `IC`, `ICU` will be evident, if the corresponding field is output right justified (session parameter `AD=R`).
7. See also *Parameters to Influence the Output of Fields* in the *Programming Guide*.

Examples:

```
DISPLAY AA(IC=*)
DISPLAY SALARY(IC='$')
```

101 ICU - Unicode 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.

Possible settings	any character	Character string to be inserted. You can specify a string of one to ten characters. **Note:** 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 a circumflex (^).
Default setting	none	
Specification within session	yes	
Applicable statements	`FORMAT`	Parameter may be specified dynamically with the `FORMAT` statement.
	`DISPLAY`	Parameter may be specified at statement level and/or at element level.
Applicable command	none	

Notes:

1. The session parameter `ICU` is identical to the session parameter `IC`. The difference is that the insertion characters are always stored in Unicode format. This allows you to specify insertion characters with mixed characters from different code pages, and assures that always the correct character is displayed independent of the installed system code page.
2. For numeric values, the insertion characters will be placed before the first significant digit printed.
3. The parameters `ICU` and `LCU` are mutually exclusive.

See also:

- *Parameters to Influence the Output of Fields* in the *Programming Guide*
- *Unicode and Code Page Support in the Natural Programming Language*, *Session Parameters*, *EMU, ICU, LCU, TCU versus EM, IC, LC, TC* in the *Unicode and Code Page Support* documentation.

102 ID - Input Delimiter Character

This Natural profile and session parameter defines the character to be used as a delimiter character for `INPUT` statements in keyword/delimiter mode.

Possible settings	any special character	Input delimiter character.
Default setting	,	Comma (,). **Note:** If the input delimiter character is to be a comma (,), it must be specified as `ID=','` when using the dynamic parameter facility, because the comma character separates individual parameters.
Dynamic specification	yes	
Specification within session	yes	
Applicable statements	`SET GLOBALS`	
Applicable command	`GLOBALS`	
Application programming interface	`USR1005N`	See *SYSEXT - Natural Application Programming Interfaces* in the *Utilities* documentation.

Notes:

1. The character specified with this parameter must not be the same as the one specified with the `DC` (decimal character) or `IA` (input assign character) parameter. In addition, we recommend that this character is not the same as the one specified with the `CF` (control character for z/OS terminal commands) or `HI` (help character) parameter.
2. 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.
3. Within a Natural session, the profile parameter `ID` can be overridden by the session parameter `ID`.

4. Under Natural Security, the setting of this parameter can be overridden by the Session Parameters option of the Library Profile.

103 IKEY - Processing of PA and PF Keys

This Natural profile parameter specifies 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`	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	

104 IM - Input Mode

This Natural profile and session parameter determines the default mode for video-terminal input.

Possible settings	F	Forms mode.
	D	Delimiter mode.
Default setting	D	
Dynamic specification	yes	
Specification within session	yes	
Applicable statements	SET GLOBALS	
Applicable command	GLOBALS	
Application programming interface	USR1005N	See *SYSEXT - Natural Application Programming Interfaces* in the *Utilities* documentation.

Notes:

1. Within a Natural session, the setting of the profile parameter IM can be overridden by the session parameter IM.
2. Under Natural Security, the setting of this parameter can be overridden by the Session Parameters option of the Library Profile.
3. For information on delimiter mode and forms mode, see the INPUT statement.

105 INIT-LIB - Library for Automatic Logon

This Natural profile parameter 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	

Note: 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).

106 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
```

107 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.

Possible settings	`ON`	A value which is identical to the previous value for the field will not be displayed. **Note:** If a `DISPLAY` or `WRITE` statement is used to create multiple output lines using the `VERT` or slash (/) 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	

Notes:

1. The `IS` parameter setting can be suspended for one record by issuing the `SUSPEND IDENTICAL SUPPRESS` statement.
2. The `IS` parameter may be used in combination with the parameters `ES` and `ZP` to cause empty line suppression.
3. See also *Parameters to Influence the Output of Fields* in the *Programming Guide*.

Example:

```
FORMAT IS=ON
```

108 ITERM - Session Termination in Case of Initialization Error

This Natural profile parameter specifies whether or not the Natural session is to continue in the case of a session initialization error.

If this profile parameter is not correctly specified, the session is terminated immediately with message:

Natural Startup Error: 105

Value of dynamic parameter `ITERM` must be `ON` or `OFF`.

Possible settings	`ON`	If a session initialization error occurs, the session is terminated immediately with message: Natural Startup Error: 106 Terminate on error during initialization. Followed by the error message of the initialization error.
	`OFF`	If an error occurs during initialization, the following happens: ■ In online mode, the initialization error is displayed, and you can choose to either continue or terminate the session. ■ In batch mode, the initialization error is reported in the batch output file, and the session is continued - possibly leading to errors or undesired results later in the session.
Default setting	`OFF`	
Dynamic specification	yes	
Specification within session	no	

109 KC - Check for Statement Keywords

This parameter corresponds to the Natural profile parameter `KCHECK`.

110 KCHECK - Check for Statement Keywords

This profile parameter checks field declarations in a Natural object against a set of critical Natural keywords.

Possible settings	`ON`	The check for keywords is performed. If a variable name defined matches one of these keywords, a syntax error is reported when the Natural object is checked or cataloged.
	`OFF`	No check for keywords is performed.
Default setting	`OFF`	
Dynamic specification	yes	
Specification within session	yes	
Applicable statements	`OPTIONS`	
Applicable command	`KCHECK`	Option of system command `COMPOPT`

Notes:

1. The section *Performing a Check for Natural Reserved Keywords* contains a list of the keywords that are checked by the `KCHECK` parameter.
2. The document *Natural Reserved Keywords* in the *Programming Guide* contains an overview of Natural keywords and reserved words.

111 KD - Key Definition

This session parameter is used to display the names assigned to the PF keys (see the SET KEY statement).

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	

Notes:

1. The PF key assignment 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.
2. As the key assignment display requires two lines, the logical page size (see the session parameter PS) must be reduced by two.
3. In case of 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
```

112 LC - Lower to Upper Case Translation

This Natural profile parameter controls lower-case to upper-case translation of input characters.

Possible settings	ON	No translation of lower-case characters to upper case is performed.
	OFF	Natural translates all lower-case characters, except input from the Natural stack which was placed there by the STACK statement, to upper case.
Default setting	ON	
Dynamic specification	yes	
Specification within session	no	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
Application programming interface	USR1005N	See *SYSEXT - Natural Application Programming Interfaces* in the *Utilities* documentation.

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

113 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 increased accordingly.

Possible settings	any character	Up to 10 characters may be specified. **Note:** 1. Leading characters may optionally be specified enclosed within apostrophes, in which case, any characters can be specified. 2. Any character string specified which contains a closing parenthesis or a quotation mark must be enclosed within apostrophes. 3. A circumflex (^) is used to represent a blank in a character string not enclosed within apostrophes.
Default setting	none	
Specification within session	yes	
Applicable statements	FORMAT	Parameter may be specified dynamically with the FORMAT statement.
	DISPLAY	Parameter may be specified at statement level and/or at element level.
Applicable command	none	

Notes:

1. The session parameters LC and IC are mutually exclusive.
2. The parameter LC can also be used with U format fields.
3. For information on Unicode format, see also *Unicode and Code Page Support in the Natural Programming Language*, *Session Parameters*, *EMU, ICU, LCU, TCU versus EM, IC, LC, TC*.

4. See also *Parameters to Influence the Output of Fields* in the *Programming Guide*.

Example:

```
DISPLAY {LC=*}
```

114 LCU - Unicode 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.

Possible settings	any character	Up to 10 characters may be specified. **Note:** 1. Leading characters may optionally be specified enclosed within apostrophes, in which case, any characters can be specified. 2. Any character string specified which contains a closing parenthesis or a quotation mark must be enclosed within apostrophes. 3. A circumflex (^) is used to represent a blank in a character string not enclosed within apostrophes.
Default setting	none	
Specification within session	yes	
Applicable statements	`FORMAT`	Parameter may be specified dynamically with the `FORMAT` statement.
	`DISPLAY`	Parameter may be specified at statement level and/or at element level.
Applicable command	none	

Notes:

1. The session parameter `LCU` is identical to the session parameter `LC`. The difference is that the leading characters are always stored in Unicode format. This allows you to specify leading characters with mixed characters from different code pages, and assures that always the correct character is displayed independent of the installed system code page.

2. The session parameters `LCU` and `ICU` are mutually exclusive.

See also:

- *Parameters to Influence the Output of Fields* in the *Programming Guide*
- *Unicode and Code Page Support in the Natural Programming Language*, *Session Parameters*, *EMU, ICU, LCU, TCU versus EM, IC, LC, TC* in the *Unicode and Code Page Support* documentation.

115 LDB - Wait Time for Response of Local Adabas Database

This Natural profile parameter 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	

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

116 LE - Reaction when Limit for Processing Loop Exceeded

This Natural profile and session parameter controls the action to be taken if the limit of retrieved records was exceeded in a `READ`, `FIND` or `HISTOGRAM` processing loop.

Possible settings	`ON`	The database loop will be terminated when the limit is reached. The program flow will continue normally with the statement following the terminated database loop. When the execution of the Natural object is complete, error NAT0957 (`Database loop limit reached with 'LE=ON'.`) is raised. **Note:** `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`	The database loop will be terminated when the limit is reached. The program flow will continue normally with the statement following the terminated database loop. When the execution of the Natural object is complete, no error message appears.
Default setting	`OFF`	
Dynamic specification	yes	
Specification within session	yes	
Applicable statements	`SET GLOBALS`	
Applicable command	`GLOBALS`	
Application programming interface	`USR1005N`	See *SYSEXT - Natural Application Programming Interfaces* in the *Utilities* documentation.

Notes:

1. The `LE` parameter applies to `READ`, `FIND` and `HISTOGRAM` statements with a limit specified (see *Example*).
2. The limit may be specified either globally for a Natural object by using the `LIMIT` statement or by specifying an explicit limit value supplied in the database processing loop.
3. Within a Natural session, the profile parameter `LE` can be overridden by using the session parameter `LE`.

Example:

```
DEFINE DATA LOCAL
1 EMPL-VIEW VIEW OF EMPLOYEES
  2 NAME
END-DEFINE
READ (10) EMPL-VIEW BY NAME
  WRITE NAME
END-READ
END
```

`LE=OFF`: after 10 records the loop ends without a message.

`LE=ON`: after 10 records the loop ends with an error message NAT0957 (`Database loop limit reached with 'LE=ON'`).

117 LFILE - Logical System File Definition

- LFILE Parameter Syntax 272
- Example of LFILE Parameter 273

This Natural profile parameter specifies information concerning the physical database file to be associated with a logical system file.

Possible settings	See *LFILE Parameter Syntax*.	
Default setting	none	
Dynamic specification	yes	
Specification within session	no	
Application programming interface	USR0011N	See *SYSEXT - Natural Application Programming Interfaces* in the *Utilities* documentation.
	USR2004N (recommended)	

Note:

- LFILE can be used for products which have their own system files (for example, Con-nect, Natural Review) to specify where such a system file is to be located. Such products use the database ID 255 and a **logical file number** in their data definition modules (DDMs). With the LFILE parameter, you specify which **physical file number** and **database ID** (and, if applicable, password and cipher key) are associated with that logical file number. Natural maps the logical file number to the physical file number and database ID and uses it for any database calls.
-

LFILE Parameter Syntax

The LFILE parameter is specified as follows:

```
LFILE=(logical-fnr,physical-dbid,physical-fnr,password,cipher-key)
```

Where:

Syntax Element	Value	Explanation
logical-fnr	1 - 251	Logical file number. This parameter is mandatory.
physical-dbid	0 - 65535, except 255	Physical database ID. Database ID 255 is reserved for logical system files.
physical-fnr	1 - 65535	Physical file number.
password	1 - 8 characters.	Password and cipher key are only required if the database file has been password-protected and/or ciphered using the Adabas security feature. With FDDM, FNAT and FUSER, the password and the cipher key are reserved for future use; currently they are ignored.
cipher-key	1 - 8 numerical digits.	

Note: To define different logical files, the LFILE parameter must be specified multiple times (separated by a comma or a blank); see *Example of LFILE Parameter*.

Example of LFILE Parameter

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

118 LOGONRQ - Logon for RPC Server Request Required

This Natural profile parameter determines whether or not logon data are required for an RPC server request.

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. **Note:** If the Natural RPC server runs under Natural Security, you are strongly recommended to set `LOGONRQ=ON`. For further information, see *Using Natural RPC with Natural Security* in the *Natural RPC (Remote Procedure Call)* documentation.
	`OFF`	A logon is *not required*. Logon data will be processed nevertheless.
Default setting	`OFF`	
Dynamic specification	yes	
Specification within session	no	

Notes:

1. `LOGONRQ` is specified on the server side only.
2. For Natural clients, the logon data can be requested either by setting the `LOGON` option of the `SYSRPC` Service Directory Maintenance or by using the **logon indicator** of parameter `DFS`.
3. For information on Natural RPC, see the *Natural RPC (Remote Procedure Call)* documentation.

119 LS - Line Size

- Profile Parameter LS 278
- Session Parameter LS 278
- Specification with Statements 279

This Natural profile and session parameter specifies the maximum number of characters permitted per line for `DISPLAY`, `INPUT` and `WRITE` statements.

The following topics are covered below:

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.
	`0`	Only permitted with the statement `SET GLOBALS` or with the system command `GLOBALS`. The value `0` will be replaced by the physical line size.
Default setting	Physical line size.	
Applicable command	`GLOBALS`	
Applicable statements	`FORMAT` `SET GLOBALS`	
Application programming interface	`USR1005N`	See *SYSEXT - Natural Application Programming Interfaces* in the *Utilities* documentation.

Notes:

1. At logon to a library, `LS` is reset to the physical line size.
2. Under Natural Security, the setting of this parameter can be overridden by the Session Parameters option of the Library Profile.

Specification with Statements

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

Applicable statements	`DISPLAY`
	`INPUT`
	`WRITE`
		Parameter may be specified at statement level.

120 LT - Limit for Processing Loops

This Natural profile and session parameter is used to limit the number of database records which can be retrieved within Natural applications.

Possible settings	1 - 2147483647 0	Maximum number of records that can be retrieved. All retrieved records (including records rejected by means of a WHERE clause) are counted and compared with this limit. LT=0 defines that no limit is in effect for the number of retrieved records. **Note:** Within a session, you can specify a value in the range of 0 to *n*, where *n* is the value of profile parameter LT at session start.
Default setting	99999999	
Dynamic specification	yes	
Specification within session	yes	
Applicable statements	SET GLOBALS	**Note:** When the LT parameter is used in conjunction with the statement SET GLOBALS or the system command GLOBALS, the limit value that can be set may not exceed the LT value defined in the Natural parameter file NATPARM.
Applicable command	GLOBALS	
Application programming interface	USR1005N	See *SYSEXT - Natural Application Programming Interfaces* in the *Utilities* documentation.

Notes:

1. The limit set with the LT parameter applies to all statements retrieving records from the database; that is, statements that initiate processing loops, such as READ, FIND, HISTOGRAM or SELECT, and statements that retrieve only a single record, such as FIND UNIQUE, FIND NUMBER, FIND FIRST, GET (SAME) and SELECT SINGLE.
2. All retrieved records are counted and the result of the count is compared with the LT limit. The count also includes those records which were rejected by a WHERE clause of a FIND, READ or

HISTOGRAM statement. The LT limit does not affect the statements STORE, UPDATE, DELETE, END TRANSACTION and BACKOUT TRANSACTION.

3. When a record is retrieved from the database, the count of retrieved records is incremented before it is compared with the current value of the LT parameter. If the incremented count exceeds the current LT value, Natural error NAT1003 (Global limit for database calls reached) is raised. The count of retrieved records is reset to zero whenever a Natural program is started on Level 1. The count is not reset if the program on Level 1 invokes another Natural object (for further information, see *Multiple Levels of Invoked Objects* in the *Programming Guide*). Therefore, the LT parameter limits the number of records retrieved from the database by a Level 1 program and objects invoked by that program on a level other than 1.
4. If the value of the LT parameter is dynamically changed within a program by using a SET GLOBALS LT=*n* statement, the new limit value becomes effective for the next statement that retrieves a record from the database.
5. Within a Natural session, the profile parameter LT can be overridden by using the session parameter LT.

121 MADIO - Maximum DBMS Calls between Screen I/O

Operations

This Natural profile parameter is used to specify 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	
Application programming interface	`USR1005N`	See *SYSEXT - Natural Application Programming Interfaces* in the *Utilities* documentation. * Recommended.
	`USR8207N*`	

Note: If the specified limit is exceeded, the Natural program is interrupted and the user is notified with an appropriate Natural error message.

122 MAINPR - Override Default Output Report Number

This Natural profile parameter is used to override the default output report number for all Natural reports. It must be set to a valid printer number (0 - 31).

Possible settings	`0 - 31`	Valid printer number.
Default setting	`0`	
Dynamic specification	yes	
Specification within session	no	
Application programming interface	`USR6002N`	See *SYSEXT - Natural Application Programming Interfaces* in the *Utilities* documentation.

Notes:

1. Specifying a `MAINPR` setting is the same as changing all of the `DISPLAY`, `PRINT`, `WRITE` or `INPUT` statements' printer references from the default setting (`0`) to the desired printer number.
2. A physical printer corresponding to the report number specified must be defined to Natural as described in the *Configuration Utility* documentation, section *Device/Report Assignments*.

123 MASKCME - MASK Compatible with MOVE EDITED

This Natural profile parameter is used to control Natural's compiler.

Possible settings	ON	The range of valid year values that match the YYYY mask characters is 1582 - 2699 to make the MASK option compatible to MOVE EDITED.
	OFF	The range of valid year values that match the YYYY mask characters is 0000 - 2699.
Default setting	OFF	
Dynamic specification	yes	
Specification within session	yes	
Applicable statements	OPTIONS	
Applicable command	MASKCME	Option of COMPOPT

124 MAXBUFF - Default Buffer Size

This Natural profile parameter is used in a Natural RPC environment to specify the default buffer sizes.

Possible settings	`1 - 2097147`, but less than or equal to *`value`* = `RPCSIZE - 4`	Default buffer size in KB.
	`0`	No buffer is allocated.
Default setting	`0`	
Dynamic specification	yes	
Specification within session	no	

Notes:

1. `MAXBUFF` can be specified on both the client and the server side.
2. On the server side, it determines the size of the buffer provided by the server to receive the client request and to send back the result. The buffer must be large enough to hold the largest data area received by all client requests and all results sent back to the client. If the size of the buffer is too small for a request, a temporary buffer with the required size is allocated and used for this request. For further information, see *Interface Objects and Automatic RPC Execution* in the *Natural RPC (Remote Procedure Call)* documentation.
3. On the client side, it determines the size of the buffer provided for the execution of Natural RPC calls. This buffer is used to build the client request and to receive the result from the server. The buffer must be large enough to hold the largest data area received by all client requests and all results sent back to the client. If the size of the buffer is too small for a request, a temporary buffer with the required size is allocated and used for this request.
4. On the client side, you need not specify `MAXBUFF` if you use an interface object generated with the SYSRPC utility and `COMPAT NONE`, and if the parameters neither contain dynamic fields, nor X-arrays or group structures.

5. The size of the data exchanged between the client and server is provided by the **Interface Object Generation** function of the SYSRPC utility. To calculate the size for RPC execution, you may use the `SYSRPC CSMASS` command; see *Calculating Size Requirements* in the *SYSRPC Utility* documentation.

125 MAXCL - Maximum Number of Program Calls

This Natural profile parameter is used to specify the maximum number of program calls permitted between two screen I/O operations.

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	
Application programming interface	`USR1005N`	See *SYSEXT - Natural Application Programming Interfaces* in the *Utilities* documentation. * Recommended.
	`USR8207N` *	

Note: If the specified limit is exceeded, the Natural program is interrupted and the user is notified with an appropriate Natural error message.

126 MAXPREC - Maximum Number of Digits after Decimal Point

This Natural profile parameter is used to control Natural's compiler. This option determines the maximum number of digits after the decimal point that the Natural compiler generates for results of arithmetic operations.

Possible settings	7,...,29	The value denotes the maximum number of digits after the decimal point that the Natural compiler generates for results of arithmetic operations.
		
		The default value 7 provides upwards compatibility for existing applications. If such applications are cataloged with MAXPREC=7, they will deliver the same results as before. Objects cataloged with a Natural version that did not support the MAXPREC option are executed as if MAXPREC=7 had been set.
		
		If higher precision is desired for intermediate results, the value should be increased.
		
		The setting of MAXPREC does not limit the number of digits after the decimal point that can be specified for user defined fields and constants. However, the precision of such fields and constants influences the precision of results of arithmetic operations. This makes it possible to benefit from enhanced precision in selected computations without having the need to set the compiler option MAXPREC to a value that unintentionally affects other computations. So even if MAXPREC=7 is in effect, the following example program can be cataloged and executed:
		
		`DEFINE DATA LOCAL`
		`1 P (P1.15)`
		`END-DEFINE`
		`P := P + 0.1234567890123456`
		`END`
		
		See also *Precision of Results of Arithmetic Operations* in the *Programming Guide*.

Default setting	7	

Dynamic specification	yes	
Specification within session	yes	
Applicable statements	OPTIONS	
Applicable command	MAXPREC	Option of system command COMPOPT

Caution: Changing the value of the MAXPREC option that is being used to catalog a Natural object may lead to different results, even if the object source has not been changed. See example below.

Example:

```
DEFINE DATA LOCAL
1 #R (P1.7)
END-DEFINE
#R := 1.0008 * 1.0008 * 1.0008
IF #R = 1.0024018 THEN ... ELSE ... END-IF
```

The value of #R after the computation and the execution of the IF statement depend on the setting of MAXPREC:

Setting of MAXPREC Effective at Compile Time	Value of #R	Executed Clause of IF Statement
MAXPREC=7	1.0024018	THEN clause
MAXPREC=12	1.0024019	ELSE clause

127 MAXYEAR - Maximum Year for Date/Time Values

This Natural profile parameter sets the maximum value for the year part of date and time values that can be entered as constants or as terminal input.

Possible settings	2699	The maximum year that can be entered is 2699; that is, the maximum date value that can be entered is 2699-12-31.
	9999	The maximum year that can be entered is 9999; that is, the maximum date value that can be entered is 9999-12-31.
Default setting	2699	
Dynamic specification	yes	
Specification within session	no	

Notes:

1. MAXYEAR=9999 changes the maximum date value that can be entered from 2699-12-31 to 9999-12-31.
2. Before setting the value for MAXYEAR to 9999, you should carefully check your application for arithmetic operations or assignments of date or time values to fields that have data formats other than date or time, and perform the necessary changes. Otherwise, unexpected overflows leading to Natural errors at execution time may occur.

For example, you should check for

- redefinitions of date/time fields with P6/P12 fields
- assignments of date/time values to non-date/time fields such as P6 := D
- arithmetic operations with date/time values where the result is assigned to a non-date/time field, for example: P6 := D + 7
- input of date/time fields that is used in arithmetic operations with non-date/time fields later on, for example:

```
INPUT D(D)
P6 := D + 1
```

The use of the Natural Engineer is recommended to check your application.

The setting of `MAXYEAR` affects

- checking of date/time constants by the compiler, for example: `P6 := D'2699-12-31'`
- `INPUT` statements with input or modifiable date/time fields
- `MOVE EDITED` statements with source or target date/time fields
- `IS (D)` option in logical condition criteria
- `MASK` option in logical condition criteria with four-digit year check `(YYYY)`
- `VAL` system function with date field as target operand

You should ensure that the `MAXYEAR` settings are the same for

- cataloging and executing a Natural application
- Natural RPC servers and Natural RPC clients

See also:

- *Formats D - Date, and T - Time* in the *Programming Guide*
- *Date and Time Constants* in the *Programming Guide*
- Session parameter `EM` in the *Parameter Reference* documentation

128 MC - Multiple-Value Field Count

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 - 191`	Number of values.
Default setting	`1`	
Specification within session	yes	

Applicable statements	`DISPLAY`
	`FORMAT`
	`INPUT`
	`PRINT`
	`WRITE`	

Applicable command	none	

Note: This parameter may be used in reporting mode only.

Example:

```
FORMAT MC=5
```

129 MFBS - Multifetch Buffer Size

This Natural Adabas specific parameter is used if Natural is accessing Adabas data with *Multifetch*. The value defines the size of the *Multi Fetch Buffer*. The value limits the number of records that can be read by a single call.

The number of records can also be limited directly by **MFMR**. The value that is reached first is decisive.

Possible settings	1 - 65536	Default buffer size in KB.
Default setting	4	
Dynamic specification	yes	
Specification within session	no	

130 MFMR - Multifetch Max Records

This Natural Adabas specific parameter is used if Natural is accessing Adabas data with *Multifetch*. The value defines the maximum amount of records that can be read by a single multi fetch access.

The number of records can also be limited using the buffer size **MFBS**. The value that is reached first is decisive.

Possible settings	1 - 1024	
Default setting	32	
Dynamic specification	yes	
Specification within session	no	

131 MFSET - Multi-Fetch Setting

This Natural profile parameter specifies whether multi-fetch (see *MULTI-FETCH Clause*) 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	

132 ML - Position of Message Line

This profile and session parameter specifies 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.

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	`B`	
Dynamic specification	yes	
Specification within session	yes	
Applicable statements	`SET CONTROL 'M'`	
Applicable command	`GLOBALS`	
Application programming interface	`USR1005N`	See *SYSEXT - Natural Application Programming Interfaces* in the *Utilities* documentation.

Notes:

1. For information on the operand `'M'`, see the Natural terminal command `%M` (Control of Message Line).
2. The profile parameter `ML` does not exist in the Configuration Utility. Use session parameter `ML` instead.
3. Within a Natural session, the profile parameter `ML` can be overridden by the session parameter `ML`.

133 MP - Maximum Number of Pages of a Report

This Natural profile and session parameter specifies the maximum number of pages to be produced for a report.

In a Natural for Windows or Natural for Linux and Cloud environment, the `MP` profile parameter is set by using the **Max. Pages** option of the Configuration Utility described in *Device/Report Assignments* in the *Configuration Utility* documentation.

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	`32767`	
Dynamic specification	no	
Specification within session	no	

Applicable statements	`DISPLAY`
	`FORMAT`
	`PRINT`
	`WRITE`	

Applicable command	none	

Note: 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.

134 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.
	`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
```

135 MSGSF - Display System Error Messages in Short/Full Format

This Natural profile parameter 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	

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.

136 MT - Maximum CPU Time

This Natural profile and session parameter determines the maximum amount of CPU time which can be used by a Natural program.

Important: In server environments where the server itself runs without any operating system controlled CPU time limit, it is strongly recommended to set the profile parameter `MT` to a non-zero value to prevent the formation of endless loops caused e.g. by application errors. This recommendation applies to Natural RPC and Natural Development Server servers.

Possible settings	`1 - 9999999`	Maximum amount of CPU time in seconds. **Note:** 1. If Natural Security is installed, the profile parameter `MT` can be overridden within Natural Security. 2. With Natural Security, the maximum value for the profile parameter `MT` is `32767`. 3. 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`	
Application programming interface	`USR1005N`	See *SYSEXT - Natural Application Programming Interfaces* in the *Utilities* documentation.

Notes:

1. This Natural profile and session parameter only applies to programs executed in batch mode, under Natural Development Server (SPoD) or under Natural for TSO.applies only to mapped z/OS environments (SPoD).
2. This Natural profile and session parameter applies only to mapped z/OS environments (SPoD).
3. CPU time measurement starts when a Natural program is started from `NEXT` mode or by means of a `FETCH` statement, that is, on program level 1. In non-batch mode (Natural Development Server, Natural for TSO), CPU time measurement is restarted at every terminal I/O.
4. The limit for programs operating in interactive mode is controlled by the TP monitor in use.
5. 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.
6. 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.
7. Within a Natural session, the `MT` profile parameter can be overridden by the session parameter `MT`.
8. When running zIIP-enabled under z/OS, the `MT` profile parameter applies separately to both TCB (except under CICS) and SRB (zIIP) processing modes. If `MT=0` is set in SRB mode, Natural uses the existing z/OS TCB time limit to avoid endless loops because there is no z/OS CPU time limit (for example, the JCL TIME parameter) for SRBs.
9. The `MT` parameter is supported under CICS SRBs (zIIP), but not under CICS TCBs. If `MT=0` is set, Natural uses the CICS runaway time as CPU time limit in SRB mode. A CPU timeout abend that occurs when a CICS open TCB is used cannot be recovered. It causes an immediate AICA abend of the CICS task and the Natural session is aborted.

137 NATLOG - Natural Log File

- Examples 316

This Natural profile parameter 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	

Notes:

1. 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.
2. Natural tries to create the following file name, if the *`user-ID`*/*`et-ID`* information could be retrieved: *NATURAL_<`user-ID`>_<`et-ID`>.LOG*
3. If *`user-ID`* and *`et-ID`* could not be retrieved, then the following file name is used: *NATURAL.LOG*.
4. Example File Names: A Natural batch process is running with *`user-ID`*=`SYSTEM` and *`et-ID`*=`14`, then the resulting file name is *NATURAL_SYSTEM_14.LOG*.
5. 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*.

Examples

The following example shows 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 batch mode. The second entry shows an error message. Both messages are counted in the statistics summary.

Example `NATLOG` Output Contents for Windows

```
# ##################################################
#           N a t u r a l   L o g   F i l e
# ##################################################
#
# Logging started at : 06-Mar-2007 08:10:12.044
#                    :
# Natural Version    : V v.r.s 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.
```

138 NC - Use of Natural System Commands

This Natural profile and session parameter controls whether Natural system commands can be used during the Natural session or not.

Possible settings	ON	System commands cannot be used. **Exceptions:** FIN, LAST, LOGOFF, LOGON, RENUMBER, RETURN, SETUP and TECH. **Note:** 1. 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. 2. In a Natural Development Server environment on z/OS computers, the value OFF will be assumed for the Natural Development Server, even if NC=ON has been specified. 3. 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		
Applicable command	GLOBALS	
Application programming interface	USR1005N	See *SYSEXT - Natural Application Programming Interfaces* in the *Utilities* documentation.

Notes:

1. Within a Natural session, the profile parameter `NC` can be overridden by the session parameter `NC`.
2. Natural terminal commands and user-created commands (object module names) are not affected by the `NC` parameter.

139 NCFVERS - NCF File Protocol Version

This Natural profile 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 or 6.2.
	3	A format file of Entire Connection Protocol Version 3 is written, which is created by Natural Version 6.3.
	4	A format file of Entire Connection Protocol Version 4 is written, which is created by Natural Version 9.3.4.
Default setting	2	
Dynamic specification	no	
Specification within session	no	

Note: For information on the work file type `ENTIRE CONNECTION` and the Entire Connection format, refer to `ENTIRECONNECTION` in the `DEFINE WORK FILE` statement documentation and *Work File Formats* in the *Operations* documentation.

140 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.

Possible settings	*nn.m*	The length is specified as *nn.m*, where *nn* represents the number of positions before the decimal separator, and *m* represents the number of positions after the decimal separator. The *m* notation is optional. The value of *m* must not exceed 7. The total of *nn+m* must not exceed 29. **Note:** 1. If NL is set less than the field length, values are truncated. No error is produced when relevant digits are truncated. 2. 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	

Notes:

1. The NL parameter must not be specified for groups.
2. Any edit mask specified for a field will override the NL parameter for this field.

3. See also *Parameters to Influence the Output of Fields* in the *Programming Guide*.

Example:

```
DISPLAY #AA(NL=20) #AB(NL=3.2)
```

141 NOAPPLERR - Suppress Message Number Prefix NAT

This Natural profile parameter is used to suppress the message number prefix “NAT” with user-supplied error messages.

Possible settings	ON	The prefix “NAT” is not displayed in error messages.
	OFF	The prefix “NAT” is displayed in error messages.
Default setting	OFF	
Dynamic specification	yes	
Specification within session	no	

142 OPF - Overwriting of Protected Fields by Helproutines

This Natural profile and session parameter specifies whether the content of a write-protected field (attribute definition AD=P) can be overwritten by a helproutine assigned to the field.

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	
Application programming interface	USR1005N	See *SYSEXT - Natural Application Programming Interfaces* in the *Utilities* documentation.

Notes:

1. 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.
2. In addition, in reporting mode you can change the OPF setting using the statement SET GLOBALS.
3. Within a Natural session, the profile parameter OPF can be overridden by the session parameter OPF.

143 OPRB - Database Open/Close Processing

- OPRB String Syntax 330

This Natural profile parameter controls the use of the Adabas C open/close commands during a Natural session.

Possible settings	`OFF`	If the `OPRB` parameter is set to `OFF`, a Natural session starts with an Adabas `OP` 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).
	`OPRB=(string)`	You can specify an open request in accordance with the syntax described below. See also the examples listed.
Default setting	`OFF`	
Dynamic specification	no	
Specification within session	no	

This Natural profile parameter is required if any of the following conditions is true:

- 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.
- The Adabas record buffer to be used with the initial Adabas `OP` command can be explicitly provided. To access databases you have to specify the DBIDs and file numbers with their corresponding access rights at the `OPRB` string.
- The character set for Adabas format W is to be provided. To access databases you have to specify the DBIDs and the required encoding name.

Entries may not contain blanks, must be enclosed in parenthesis and must follow the rules defined in the relevant Adabas documentation.

In addition to the Adabas syntax, internal file numbers can be specified by using the *x-y* notation (that is, all numbers between *x* and *y*).

OPRB String Syntax

`DBID=(x)` specifies the database for the following access right entries:

`ACC=(file-list)`	Specifies access permission (read) for the files in the file list.
`UPD=(file-list)`	Specifies update permission (read/write) for the files in the file list.
`EXU=(file-list)`	Specifies exclusive update permission (exclusive read/write) for the files in the file list.
`WCODE=encoding`	Specifies the encoding for W fields in the Adabas user session (Adabas for z/OS). Required encoding code for Adabas on z/OS is 4095.

WCHARSET=*charset*	Specifies the default character set used for W fields in record and value buffers in the Adabas user session (Adabas for Linux and Windows). Required encoding names for Adabas on Linux and Windows are: UTF-16LE (for little-endian machines) UTF-16BE (for big-endian machines) UTF-16 (ADALNK decides whether to use LE or BE)

The trailing record buffer dot (.) can be omitted in the OPRB string because it is appended automatically.

DBID=0 specifies the default record buffer entry and can be omitted if it is the first DBID listed in the OPRB string. This default record buffer is taken if there is no specific entry for the requested database.

Combinations of the keywords ACC, UPD, EXU and WCODE or WCHARSET must follow the rules as defined in the relevant Adabas documentation. Natural issues an OP command at the start of a Natural session and a CL command at the end of the session. At the end of a Natural program, only the required RC commands are issued to release held ISN lists.

Example 1:

```
(ACC=2,3,4,DBID=15,UPD=3,4,ACC=5)
```

The following entries were defined:

```
'UPD=3,4,ACC=5.' for DB 15
'ACC=2,3,4.' for other databases (DB 0)
```

Example 2:

```
(DBID=15,ACC=2-7)
```

The following entry was defined:

```
'ACC=2,3,4,5,6,7.' for DB 15; access to other databases is not permitted
```

Example 3:

```
(DBID=0,ACC=2,3,4,5.)
```

The following entry was defined:

```
'ACC=2,3,4,5.' for all databases (DB 0)
```

Note: If you have Natural Security installed, open/close processing works the same way as without Natural Security; the `OPRB` parameter in the security profile is provided for future use only.

Example 4:

```
(DBID=0,ACC=2,3,4,5,DBID=12,WCHARSET='UTF-16LE',UPD=3-10)
```

The following entries were defined:

```
'ACC=2,3,4,5.' for all databases (DB 0)
```

```
WCHARSET='UTF-16LE',UPD=3,4,5,6,7,8,9,10. for DB 12
```

144 PARM - Alternative Parameter File

This Natural profile parameter can be used for Natural startup (Studio/Runtime/Server) in order to specify an alternative parameter file Natural is to run with.

Possible settings	1 - 8 characters	Any valid file name.
Default setting	none	
Dynamic specification	yes	This parameter can only be specified dynamically.
Specification within session	no	

Notes:

1. If no `PARM` parameter is specified, Natural will start using the default `NATPARM` parameter file containing all default settings for each single parameter. Use the Natural Configuration Utility in order to create your own new parameter file.
2. See *Creating a New Parameter File* in the *Configuration Utility* documentation.

145 PC - Periodic Group Count

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.

Possible settings	`1 - 191`	Number of values.
Default setting	`1`	
Specification within session	yes	
Applicable statements	`FORMAT`	

	`INPUT`
	`DISPLAY`
	`WRITE`
	`PRINT`
		Parameter may be specified at statement level and/or at element level.

Applicable command	none	

Note: This session parameter may be used in reporting mode only.

Example:

```
FORMAT PC=5
```

146 PCHECK - Parameter Checking for Object Calling Statements

This Natural profile parameter is used to control Natural's compiler.

Possible settings	ON	The compiler checks the number, format, length, and array index bounds of the parameters that are specified in an object calling statement, such as CALLNAT, PERFORM, INPUT USING MAP, PROCESS PAGE USING, OPEN DIALOG, SEND EVENT, helproutine calls. Also, the OPTIONAL feature of the DEFINE DATA PARAMETER statement is considered in the parameter check. The parameter check is based on a comparison of the parameters of the calling statement with the DEFINE DATA PARAMETER definitions for the object to be called. It requires that ■ the name of the object to be called is defined as an alphanumeric constant (not as an alphanumeric variable), ■ the object to be called is available as a cataloged object. Otherwise, PCHECK=ON will have no effect. **Error Control for PCHECK=ON** The parameter check is executed only when the object does not contain any syntax errors. The parameter check is executed for every object calling statement. The parameter check is controlled by the PECK profile parameter (see the Parameter Reference documentation). **Problems in Using the CATALL Command with PCHECK=ON** When a CATALL command is used in conjunction with PCHECK=ON, you should consider the following:

		If a CATALL process is invoked, the order in which the Natural objects are compiled depends primarily on the type of the object and secondarily on the alphabetical name of the object. The object type sequence used is: DDMs, GDAs, LDAs, PDAs, classes, maps, dialogs, external subroutines, subprograms, functions, adapters, helproutines, programs. Within objects of the same type, the alphabetical order of the name determines the sequence in which they are cataloged. As mentioned above, the parameters of the object calling statement are checked against the compiled form of the called object. If the calling object (the one which is being compiled and includes the object calling statement) is cataloged before the invoked object, the PCHECK result may be wrong if the parameters in the object calling statement and in the called object were changed. In this case, the new object image of the called object has not yet been produced by the CATALL command. This causes the *new* parameter layout in the object calling statement to be compared with the *old* parameter layout of the DEFINE DATA PARAMETER statement of the called subprogram. Solution: ■ Set compiler option PCHECK to OFF. ■ Perform a general compile with CATALL on the complete library, or if just one or a few objects were changed, perform a separate compile on these objects. ■ Set compiler option PCHECK=ON. ■ On the complete library, perform a general compile with the function CHECK of system command CATALL enabled.
	OFF	No parameter check is performed.
Default setting	OFF	
Dynamic specification	yes	
Specification within session	yes	
Applicable statements	OPTIONS	
Applicable command	PCHECK	Option of system command COMPOPT

147 PECK - PCHECK/ECHECK Error Processing

This Natural profile parameter controls whether a compilation check with the `ECHECK` or `PCHECK` option of the `COMPOPT` system command (see the *System Commands* documentation) terminates after a syntax error is detected in the object source. In addition, `PECK` determines how the syntax errors are reported.

Possible settings	`S`	Stops when the first syntax error is detected. The cursor is placed in the line that contains the error and the respective error (for example, NAT0935) is issued.
	`WS`	Same as `S` above, but additionally clears the message buffer when the compilation starts.
	`F`	Scans the entire object and places all errors on a stack. The cursor is placed in the line where the first error is detected. If several errors occur in the same line, a Natural error message appears in this line indicating that inconsistencies were found during the `PCHECK/ECHECK` validation. If several errors occur in different lines, the above Natural error message appears in the first line. All errors accumulated on the stack are listed after the scan is complete.
	`WF`	Same as `F` above, but additionally clears the message buffer when the compilation starts.
	`WL`	Same as `L` below, but additionally clears the message buffer when the compilation starts.
Default setting	`L`	Scans the entire object and places all errors on a stack. The cursor is placed in the line where the last error is detected. If several errors occur in the same line, a Natural error message appears in this line indicating that inconsistencies were found during the `PCHECK/ECHECK` validation. If several errors occur in different lines, the above Natural error message appears in the last line.

		All errors accumulated on the stack are listed after the scan is complete.
Dynamic specification	yes	
Specification within session	no	

148 PERSIST - Flag a Parameter File as Persistent

Caution: This Natural profile 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 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 operating system 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 file is not flagged as persistent.
Default setting	`OFF`	
Dynamic specification	no	
Specification within session	no	

149 PM - Print Mode

- Profile Parameter PM 344
- Session Parameter PM 344

The following topics are covered below:

Profile Parameter PM

The Natural profile parameter PM specifies how fields are to be printed or displayed.

Notes:

1. PM=I affects any system controlled output screen items, that is, system variables and PF key lines. Moreover, all non-alphanumeric fields, for example, numeric and date are affected. In addition, for Natural Web I/O Interface terminals the field sequence is changed from left to right into right to left. The field inversion routine is supplied as assembler module NATPM in the Natural source library and can be modified in case of need.
2. For detailed information on how to use the setting PM=I, see *Bidirectional Language Support* in the *Unicode and Code Page Support* documentation.

Possible settings	I	The default screen direction when running programs is right-to-left. The RTL attribute for new dialogs is checked.
	R	The default screen direction when running programs is left-to-right. The RTL attribute for new dialogs is not checked.
Default setting	R	
Dynamic specification	yes	
Specification within session	yes	
Application programming interface	USR1005N	See *SYSEXT - Natural Application Programming Interfaces* in the *Utilities* documentation.

Session Parameter PM

This session parameter PM is used to indicate how fields are to be displayed.

Possible settings	PM=I	The field direction is reversed.
	PM=N	The field is ignored (that is, not printed) for hardcopy output.
Default setting	none	The default field direction is used and it is regarded for hardcopy output.
Applicable statements	COMPRESS DEFINE DATA DISPLAY FORMAT INPUT	

	MOVE PRINT WRITE	

For detailed information on how to use the setting `PM=I`, see *Bidirectional Language Support* in the *Unicode and Code Page Support* documentation.

150 PRGPAR - Data to be Passed to Program Receiving Control at Termination

This Natural profile parameter 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	

151 PROFILER - Profile a Natural Session

- PROFILER Parameter Syntax 350
- Examples of PROFILER Parameter 353

This profile parameter is used to profile a Natural session. The profiling data is written to a resource file you can analyze with the Natural Profiler in NaturalONE. For more information, see the *NaturalONE* documentation.

Possible settings	See ***PROFILER Parameter Syntax***.	
Default setting	none	See the default settings of the subparameters in ***PROFILER Parameter Syntax***.
Dynamic specification	yes	
Specification within session	no	

PROFILER Parameter Syntax

The `PROFILER` parameter is specified as follows:

```
PROFILER={(subparameter=value[,subparameter=value]...)}
```

Important: Blank spaces are not allowed in the syntax. Use commas to separate the syntax elements.

Where:

Subparameter	Value	Explanation
`ACTIVE`		`ACTIVE=value` determines whether the profiling infrastructure is activated. Default: `OFF`
	`ON`	The profiling infrastructure is activated and events are written to the resource file.
	`OFF`	The profiling infrastructure is deactivated.
`EVENT`		`EVENT=value` specifies the types of events to be generated into the resource file. The following syntax applies: `event|(event[,event]...)` where: `event` is either the type of an event (`event-type`) or a group of event types (`event-group`). Default: All event types are processed.

Subparameter	Value	Explanation
	event-type: SI|ST|PL|PS| PT|PR|DA|DB|IA| IB|CA|CB|NS|E|U	*event-type* is one of the following: SI Session initialization event ST Session termination event PL Program load event PS Program start event PT Program termination event PR Program resume event DA After database call event DB Before database call event IA After terminal I/O event IB Before terminal I/O event CA After external program call event CB Before external program call event NS Natural statement event E Runtime error event U User-defined event **Note:** 1. Natural statement events (NS) are only generated if the corresponding Natural object was compiled with SYMGEN=ON (see the SYMGEN profile parameter)! 2. In addition to the events listed above, the Natural Profiler collects monitor pause events (MP) when the data collection is paused. The duration of the pause is not considered by the application performance analysis.
	event-group: S|P|D|I|C|N	*event-group* is one of the following: S Session event group (SI, ST) P Program event group (PL, PS, PT, PR) D Database call event group (DB, DA) I Terminal I/O event group (IB, IA) C External program call event group (CB, CA) N Natural event group (NS)
EVENTTRACE		EVENTTRACE=*value* determines whether individual events are written to the resource file. Default: OFF

Subparameter	Value	Explanation
	ON	Individual events are written to the resource file. You can examine the recorded events in the NaturalONE **Event Trace** page. The generated resource file has the extension `.nprf` (Natural Profiler resource file). **Caution:** The resulting file can become very large, especially when statement events are recorded.
	OFF	Individual events are not written to the resource file. Only consolidated hot-spot information is recorded in the file. This typically results in a much smaller file that is consequently much quicker to process. The generated resource file has the extension `.nprc` (Natural Profiler resource consolidated).
INTERVAL	1|2|4|5|8|10| 16|20|25|40|50| 80|100|125|200| 250|400|500|625| 1000|1250|2000| 2500|5000|10000	INTERVAL=*value* specifies the CPU time interval (in microseconds) to be used if SAMPLING is active. With sampling, only the last event of each sampling interval is recorded. Exception: Session events (event types SI and ST) are always recorded. Larger sampling intervals result in fewer events recorded and thus smaller resource files. However, larger sampling intervals also result in less accurate values. Default: 100
RESLIB	1 – 8 characters	RESLIB=*value* specifies the name of the FUSER system library that contains the resource file. Default: SYSTEM
RESNAME	1 – 253 characters	RESNAME=*value* specifies the name of the resource file (without path and extension) into which the data is written. Default: A file name is automatically generated containing the current user ID and timestamp.
SAMPLING		SAMPLING=*value* determines whether sampling is activated. The sampling method uses a statistical approach to collect data. Sampling significantly reduces the amount of data written to the resource file while approximately retaining the same CPU times as without sampling. **Caution:** Sampling gives an estimation of the consumed CPU time. Other values like the elapsed times or hit counts are not reliable when sampling is used. For details on sampling, see the *Sampling* in the *Natural Profiler Utility - Batch Mode* section of the Natural for z/OS documentation.

Subparameter	Value	Explanation
		Default: OFF
	ON	Sampling is activated.
	OFF	Sampling is deactivated. This is the default setting for this subparameter in the Natural parameter file.
FLAGS		FLAGS=*value* allows option flags to be passed to the profiler. Note that these are bitmask values. To combine option flags, you must specify the sum of their individual numerical values.
	2	Write a data collector trace file to the Natural temporary directory (see the TMP_PATH entry in the *NATURAL.INI* file).
	8	Trace resource file generation to the Natural temporary directory. Does not apply if profiling via NaturalONE, where the profiling data is sent directly to the client instead of being written to a resource file on the server.
	64	Do not create an elapsed time timer (elapsed times will all be 0).
	128	Use an elapsed time timer for the CPU times. The timer is paused while waiting for user input. Has the benefit that elapsed times are often much quicker to measure than CPU times, reducing the profiling overhead and thus improving the accuracy of the profiling results.
	512	If this flag is specified, the profiler attempts to deduct as much of the profiling overhead as possible from the measured CPU times, further improving the accuracy of the profiling results.

Examples of PROFILER Parameter

Example 1: Profile with all Events

```
PROFILER=(ACTIVE=ON,EVENTTRACE=ON,EVENT=(S,P,D,N,I,C,E,U))
```

All individual events are recorded.

Example 2: Profile with Program Load Event

```
PROFILER=(ACTIVE=ON,EVENT=PL,RESNAME=MYAPP,RESLIB=MYLIB)
```

All program load events (PL) are written to the resource file MYAPP in the library MYLIB.

Example 3: Profile with Sampling

```
PROFILER=(ACTIVE=ON,EVENT=(P,NS),SAMPLING=ON,INTERVAL=1000)
```

All programs of the event group `P` and the event `NS` are recorded using sampling with a sampling interval of 1000 microseconds.

Example 4: Sampling versus Non-Sampling

A Natural application is profiled twice. In a first run, without sampling:

```
PROFILER=(ACTIVE=ON)
```

The Natural Profiler generates 240,086 events and shows a CPU consumption of 30.2 percent for a called subprogram.

In the second run, the same application is profiled with sampling:

```
PROFILER=(ACTIVE=ON,SAMPLING=ON,INTERVAL=100)
```

The Natural Profiler now generates only 4,100 events and shows a CPU consumption of 30.1 percent for the same subprogram.

152 PROGRAM - Non-Natural Program Receiving Control after Termination

This Natural profile parameter specifies the non-Natural program which is to receive control after the termination of the Natural session.

Possible settings	1 - 12 characters	Non-Natural program
Default setting	none	
Dynamic specification	yes	
Specification within session	no	
Application programming interface	`USR6204N` (for all platforms)	See *SYSEXT - Natural Application Programming Interfaces* in the *Utilities* documentation.

Note: Data for the program specified with the profile parameter `PROGRAM` can be supplied with the `TERMINATE` statement.

153 PS - Page Size for Natural Reports

This Natural profile and session parameter specifies the maximum number of lines per page to be used for Natural reports created with the `DISPLAY` or `WRITE` statement.

Possible settings	`1 - 250`	Maximum number of lines per page.
	`0`	The physical page size is to be used. **Note:** 1. If `PS=0` is specified for the first report to be output (Report 0), the physical-device page-size minus 1 will be used. 2. 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`
	`SET GLOBALS`
	`WRITE`	

Applicable command	`GLOBALS`	
Application programming interface	`USR1005N`	See *SYSEXT - Natural Application Programming Interfaces* in the *Utilities* documentation.

Notes:

1. When used as a profile parameter, the `PS` parameter is honored in batch mode only and defines the physical page size.
2. In online mode, the physical page size is always set to the physical screen height.
3. See also *Page Size - PS Parameter* in the *Programming Guide*.
4. Under Natural Security, the setting of this parameter can be overridden by the Session Parameters option of the Library Profile.

154 PSIGNF - Internal Representation of Positive Sign of Packed Numbers

This Natural profile parameter can be used to define the internal representation of the positive sign of packed numbers.

Possible settings	ON	The positive sign of a packed number is represented internally as H'F'.
	OFF	The positive sign of a packed number is represented internally as H'C'.
Default setting	OFF	
Dynamic specification	yes	
Specification within session	yes	
Applicable statements	OPTIONS	
Applicable command	PSIGNF	Option of system command COMPOPT

155 RCFIND - Handling of Response Code 113 for FIND Statement

This Natural profile parameter specifies 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	

156 RCGET - Handling of Response Code 113 for GET Statement

This Natural profile parameter specifies 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	

157 RDACTIVE - Activate Remote Debugger

This Natural profile parameter specifies whether a remote debugger on a Windows 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 `RDPORT` (where the port number is specified).

Possible settings	`ON`	Remote debugging is enabled. For DCOM (Windows only) or RPC servers, the remote debugging session is opened automatically.
	`OFF`	Remote debugging is not possible. It is possible, however, to use the debugger integrated in Natural Studio.
Default setting	`OFF`	
Dynamic specification	yes	
Specification within session	no	

Note: See the *Debugger* documentation for further details.

158 RDNODE - Remote Debugger Node Name

This Natural profile parameter specifies the node name of the Windows computer on which a remote debugger is to be called.

Possible settings	Character string	Node name
Default setting	none	
Dynamic specification	yes	
Specification within session	no	

Note: See the *Debugger* documentation for further details.

159 RDPORT - Remote Debugger Port

This Natural profile parameter specifies the port number on which a remote debugger on a Windows computer has been installed.

Possible settings	0 or 1 - 9999	Port number. By default, the debugger service is installed on the Windows computer's Port 2600. You can therefore leave the UNIX computer's RDPORT setting at the default. If, however, Port 2600 on Windows has been reserved for another service, and a different port number has been specified, you must change RDPORT accordingly.
Default setting	2600	
Dynamic specification	yes	
Specification within session	no	

Note: See the *Debugger* documentation for further details.

160 RDS - Define Remote Directory Server

This Natural profile parameter allows you to define up to 10 remote directory servers in a Natural RPC environment. For each remote directory server, you specify up to 5 positional subparameters.

Possible settings	See *RDS Parameter Syntax*.	
Default setting	none	Subparameter defaults, see *RDS Parameter Syntax*.
Dynamic specification	yes	
Specification within session	no	

Notes:

1. RDS is specified on the client side only.
2. For information on Natural RPC, see the *Natural RPC (Remote Procedure Call)* documentation.

RDS Parameter Syntax

The parameter 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,ser
name,subprogram,logon-indicator,transport-protocol-name)...(server-name,server-node-name,subprogram,logo
```

Where:

Syntax Element	Value	Explanation
server-name	1 - 8 characters	The server name. There is no default, the value must be specified.
server-node-name	1 - 8 characters	The server node name. There is no default, the value must be specified.
subprogram	1 - 8 characters	The name of the subprogram titled `CALLNAT`, which is to be used as an interface. The default name is `RDSSCDIR`.
logon-indicator	`L`	The client initiates a Natural logon to the server with the library name of the current library on the client. On Windows platforms: Instead of specifying `L`, check the selection box.
	(blank)	Blank means that no server logon will be executed. If nothing is specified, this is the default.
transport-protocol-name	`ACI`	The name of the transport protocol to be used. `ACI` is the only possible value and the default.

161 RECAT - Dynamic Recataloging

This Natural profile parameter specifies 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	Natural issues an error message and disables the **system commands** CATALOG, PURGE and SAVE. **Note:** 1. If the RECAT parameter has been set to ON, and an object exists in both source and cataloged form, then the source and the cataloged object cannot be processed independently. In order to ensure consistency between the source and the cataloged object, Natural disables the system command CATALOG (also when invoked via CATALL). In addition, the system commands PURGE and SAVE are disabled for a source for which a corresponding cataloged object exists. 2. Only objects satisfying the criteria for a particular command (such as PURGE) will be displayed in the corresponding selection box.
	OFF	Natural issues an error message.
Default setting	OFF	
Dynamic specification	yes	
Specification within session	no	
Application programming interface	USR1005N	See *SYSEXT - Natural Application Programming Interfaces* in the *Utilities* documentation.

162 REINP - Issue Internal REINPUT Statement for Invalid Data

This Natural profile and session parameter can be used to prevent an internal `REINPUT` for invalid data.

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`	
Application programming interface	`USR1005N`	See *SYSEXT - Natural Application Programming Interfaces* in the *Utilities* documentation.

Notes:

1. 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.
2. Within a Natural session, the profile parameter `REINP` can be overridden by the session parameter `REINP`.

163 RI - Release ISNs

This Natural profile parameter specifies 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. **Note:** 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	
Application programming interface	`USR1005N`	See *SYSEXT - Natural Application Programming Interfaces* in the *Utilities* documentation.

Note: 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.

164 RNCONST - Renumber Line Numbers in Constants

This Natural profile parameter can be used to renumber the line number references in alphanumeric and Unicode constants within a Natural source. See also *Renumbering of Source-Code Line Number References* in the *Programming Guide*.

Possible settings	ON	The line number references within alphanumeric and Unicode constants are renumbered.
	OFF	The line number references within alphanumeric and Unicode constants are not renumbered. They remain as they are.
Default setting	OFF	
Dynamic specification	no	
Specification within session	no	
Applicable statements	none	
Applicable command	RENUMBER	

Note: The setting of RNCONST affects the execution behavior of the RENUMBER system command.

165 ROSY - Read-Only Access to System Files

This Natural profile parameter disables modifications on the Natural system files FDDM, FNAT, FUSER, FDIC* and FSEC*.

* Not supported on this platform.

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	

166 RPCSDIR - Library for Service Directory

This Natural profile parameter specifies the name of the Natural library (or one of its steplibs) used by the RPC client at runtime.

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

Notes:

1. The parameter `RPCSDIR` is specified on the client side only.
2. It is evaluated by the SYSRPC utility functions Service Directory Maintenance and Server Command Execution.
3. For information on Natural RPC, see the *Natural RPC (Remote Procedure Call)* documentation.

167 RTINT - Allow Runtime Interrupt

This Natural profile parameter determines whether it should be allowed to interrupt a running Natural application that does not respond anymore by using the interrupt key combination of the operating system (CTRL+BREAK).

Possible settings	ON	Interrupts are allowed.
	OFF	Interrupts are not allowed.
Default setting	OFF	
Dynamic specification	yes	
Specification within session	no	

Note: For further information, see *Interrupting a Running Natural Application* in the *Programming Guide*.

168 RQTOUT – REQUEST DOCUMENT Timeout

This Natural profile and session parameter specifies the timeouts used for HTTP requests issued internally by the `REQUEST DOCUMENT` statement. If this time is exceeded, the request (connect, data send or data receive) will be terminated with a corresponding error message.

Possible settings	`0` or `1 - 65535`	Seconds. A value of zero implies no timeout.
Default setting	`0`	
Dynamic specification	yes	
Specification within session	yes	The setting of this parameter can be changed using the `GLOBALS` system command.

Note: This parameter is not available on z/OS platforms.

169 SA - Sound Terminal Alarm

This Natural profile and session parameter specifies whether the terminal alarm feature is to be used.

Possible settings	ON	The terminal alarm sound is output each time the user is prompted for input by Natural.
	OFF	No terminal alarm is used for input prompting, however, the alarm may still be activated with the *ALARM Option* of the REINPUT statement.
Default setting	OFF	
Dynamic specification	yes	
Specification within session	yes	
Applicable statements	SET GLOBALS	
Applicable command	GLOBALS	
Application programming interface	USR1005N	See *SYSEXT - Natural Application Programming Interfaces* in the *Utilities* documentation.

Note: Within a Natural session, the profile parameter SA can be overridden by the session parameter SA.

170 SB - Selection Box

- Syntactical Considerations 392
- Runtime Considerations 393

Selection boxes in an INPUT statement are available on z/OS 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

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	For example: *PROGRAM, *COM
Named Constants (z/OS only)	defined with a CONST clause of DEFINE DATA statement.

In addition to the SB attribute, other attributes can be defined as well, for example: 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, such as 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; that is, 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.

```
DEFINE DATA
LOCAL
1 A(A4)
END-DEFINE
MOVE 'ABCD' TO A
*
SET KEY PF1 = HELP
FORMAT KD=ON
*
INPUT A (AD=M EM=X.X.X.X SB='1234','WXYZ')
WRITE A
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.

171 SD - Time Delay between Two Screens

This Natural profile parameter can be used to delay 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	`1 - 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.
	`0`	
Default setting	`0`	
Dynamic specification	no	
Specification within session	no	

172 SERVER - Start Natural Session as an RPC Server Session

This Natural profile parameter specifies whether or not the Natural session will be started as an RPC server session.

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	

Notes:

1. `SERVER` can be specified on both the client and the server side.
2. For information on Natural RPC, see the *Natural RPC (Remote Procedure Call)* documentation.

173 SF - Spacing Factor

This Natural profile and session parameter specifies the default number of spaces to be inserted between field settings of columns on Natural reports created using a `DISPLAY` statement.

Possible settings	`1 - 30`	Number of spaces. **Note:** 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`	
Application programming interface	`USR1005N`	See *SYSEXT - Natural Application Programming Interfaces* in the *Utilities* documentation.

Notes:

1. Within a Natural session, the profile parameter `SF` can be overridden by the session parameter `SF`.
2. Under Natural Security, the setting of this parameter can be overridden by the *Session Parameters* option of the Library Profile.
3. See also *Column Spacing - SF Parameter and nX Notation* in the *Programming Guide*.

174 SG - Sign Position

This session parameter determines whether or not a sign position is to be allocated for a numeric field.

Possible settings	ON	A sign position will be allocated.
	OFF	No sign position will be allocated. **Note:** 1. SG=OFF causes numeric fields with negative values to be output without a minus (-) sign. 2. SG=OFF does not prevent you from entering negative values in input fields.
Default setting	ON	
Specification within session	yes	
Applicable statements	DISPLAY FORMAT INPUT PRINT WRITE	
Applicable command	none	

Notes:

1. If the EM (edit mode) parameter is specified, it overrides the SG parameter.
2. See also *Parameters to Influence the Output of Fields* in the *Programming Guide*.

Example:

```
FORMAT SG=OFF
```

175 SM - Programming in Structured Mode

This Natural profile and session parameter specifies whether or not structured mode must be used.

Possible settings	`ON`	Forces the use of structured mode syntax.
	`OFF`	Programming can be done in either structured mode or reporting mode.
Default setting	`OFF`	
Dynamic specification	yes	
Specification within session	yes	
Applicable statements	none	
Applicable command	`GLOBALS`	
Application programming interface	`USR1005N`	See *SYSEXT - Natural Application Programming Interfaces* in the *Utilities* documentation.

Notes:

1. If structured mode (`SM=ON`) is specified by profile parameter `SM`, an attempt to change this setting with system command `GLOBALS` and session parameter `SM` will be rejected (Reporting mode not permitted).
2. Within a Natural session, the profile parameter setting `SM=OFF` can be overridden by the session parameter `SM=ON`.
3. Under Natural Security, 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.
4. Under Natural Security, this parameter may be disabled by Natural Security to the effect that structured mode is invariably in effect for a given library.

176 SNAT - Sound a Bell at Syntax Error

This Natural profile parameter 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	

177 SORTSZE - Size of Sort Buffer

This Natural profile parameter specifies the amount of storage to be reserved for use by the sort program.

Possible settings	500 - 2048	Buffer size in KB.
Default setting	500	
Dynamic specification	yes	
Specification within session	no	

Notes:

1. This sort buffer is only allocated when executing a Natural program which contains a SORT statement.
2. Increasing the buffer size setting leads to faster SORT processing, in particular when all data to be sorted fit into the sort buffer.

178 SPODDEBUGPORT - Debugger Port for Debugging in the Context of SPoD

This Natural profile parameter specifies the port number which will be used when debugging applications in a mapped environment.

Possible settings	0 or 1 - 65535	Port Number.
Default setting	0	
Dynamic specification	yes	
Specification within session	no	

Notes:

1. This Natural profile parameter is only relevant if you intend to use the old Natural debugger in conjunction with an earlier version (NDV2.1) of the Natural Development Server.
2. By default the port number is set to 0. Natural Studio then evaluates a free port on the PC. The evaluated port will be used as the communication channel between the Natural Studio Debugger and the remote application. The host where the Natural Development Server (NDV) is running, must be prepared to work with any port.
3. If a value not equal to zero is specified, Natural Studio will use that specified port as the debug port for the communication channel between the Natural Studio Debugger and the remote application. Use this option, if only a limited number of ports is allowed on the host where the NDV server is running.

179 SRETAIN - Retain Source Format

This Natural profile parameter specifies that all existing sources have to be saved in their original encoding format.

Possible settings	ON	The original code page of an existing Natural source is retained. If the profile parameter SUTF8 is defined as well, new sources will be saved in UTF-8 format.
	OFF	For existing Natural sources with format UTF-8 the encoding will not be changed. Existing sources with other encodings will be saved using the current code page.
Default setting	ON	
Dynamic specification	yes	
Specification within session	no	

Notes:

1. When new sources are created, they will be saved either in the current code page format or in UTF-8 format, depending on the setting of profile parameter SUTF8. This is independent of the setting of SRETAIN.
2. If a source can not be saved in the target code page format, because this code page does not define all characters contained in the source, a message is displayed which allows the user to choose whether he/she wants to remove the problematic characters or cancel the save process.
3. See also *Profile Parameters* in the *Unicode and Code Page Support* documentation.

180 SRVCMIT - Server Commit Time

This Natural profile parameter specifies the time at which a Natural RPC server automatically commits an RPC conversation or a non-conversational RPC request.

Possible settings	B	The Natural RPC server automatically commits a database transaction before the reply is sent to the client. **Note:** If the reply fails, the database transaction is already committed.
	A	The Natural RPC server automatically commits a database transaction after the reply has been successfully sent to the client. **Note:** If the reply fails, the database transaction is rolled back.
Default setting	B	
Dynamic specification	yes	
Specification within session	no	

Notes:

1. `SRVCMIT` is specified on the server side only.
2. This parameter is only evaluated if the profile parameter `ETEOP` is set to `ON`.
3. For information on Natural RPC, see the *Natural RPC (Remote Procedure Call)* documentation.

181 SRVNAME - Name of RPC Server

This Natural profile parameter specifies the name of the RPC server, with which it registers on the node specified with the profile parameter `SRVNODE`.

Possible settings	1 - 32 characters	Valid server name. In case of an EntireX Broker node, the value of `SRVNAME` corresponds to the value of the `SERVER` attribute of a service entry in the EntireX Broker attribute file, as shown below: `CLASS=RPC, SERVICE=CALLNAT, SERVER=srvname` See *Example*.
Default setting	none	
Dynamic specification	yes	
Specification within session	no	

Note: For information on Natural RPC, see the *Natural RPC (Remote Procedure Call)* documentation.

Example

```
SRVNAME='PRODUCTION_SERVER'
```

182 SRVNODE - Name of Node

This Natural profile parameter specifies the name of the node upon which an RPC server registers.

Possible settings	1 - 192 characters	Node name. In case of an EntireX Broker node, a node name refers to a TCP/IP address. For details about the structure of node names and their support by the EntireX Broker stubs, refer to the EntireX documentation. See *Examples*.
Default setting	none	
Dynamic specification	yes	
Specification within session	no	

Notes:

1. `SRVNODE` is specified on the server side only.
2. For information on Natural RPC, see the *Natural RPC (Remote Procedure Call)* documentation.

Examples

The examples below are based on the EntireX notation.

```
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 with port number */
```

Notes:

1. 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.

2. If the port number is omitted, 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.

183 SRVRTRY - Number of Connect/Reconnect Attempts

This Natural profile parameter specifies the number of attempts for an RPC server to connect/reconnect (`REGISTER`) to an EntireX Broker that is not active, and the wait time between two successive attempts.

Possible settings	See *SRVRTRY Parameter Syntax*.	
Default setting	`0,60`	No attempts.
Dynamic specification	yes	
Specification within session	no	

Note: `SRVRTRY` is specified on the server side only.

SRVRTRY Parameter Syntax

The `SRVRTRY` parameter syntax is as follows:

```
SRVRTRY=(attempts,wait-time)
```

Or:

```
SRVRTRY=attempts
```

Note: If only a value for *attempts* is specified, the parentheses may be omitted.

Where:

Syntax Element	Value	Explanation
attempts	0 or 1 - 2147483647	Number of attempts to connect/reconnect to an EntireX Broker that is not active (EntireX Broker message 02150148). **Note:** 1. The specification of *attempts* enables you to start a Natural RPC server before the required EntireX Broker has been started and to shutdown an EntireX Broker temporarily without implicitly terminating all Natural RPC servers. 2. If the EntireX Broker is still not active after the number of attempts specified in *attempts* or if *attempts* is zero, the RPC server terminates.
wait-time	0 or 1 - 3600	Wait time in seconds between two successive attempts.

Examples

1. `RPC=(SRVRTRY=(20,10))`

   Or:

   ```
   NTRPC SRVRTRY=(20,10)
   ```

   20 attempts with a wait time of 10 seconds between two successive attempts.

2. `RPC=(SRVRTRY=500)`

   Or:

   ```
   NTRPC SRVRTRY=500
   ```

   500 attempts with a wait time of 60 seconds between two successive attempts.

Note: For further information, see the Natural *Natural RPC (Remote Procedure Call)* documentation, and especially *Considerations for z/OS Natural RPC Servers with Replicas*.

184 SRVTERM - Server Termination Event

This Natural profile parameter specifies the event at which a Natural RPC server is automatically terminated.

Possible settings	`NEVER`	A Natural RPC server is never automatically terminated. **Note:** To terminate a Natural RPC server, refer to *Terminating a Natural RPC Server* and *Using Application Programming Interface USR2075N* (for the EntireX Broker Service) in the *Natural RPC (Remote Procedure Call)* documentation.
	`TIMEOUT`	A Natural RPC server is automatically terminated if the wait time for the next client request outside of an RPC conversation is exceeded. **Note:** `TIMEOUT` should only be set if you use an Attach Manager to dynamically start Natural RPC servers on request.
Default setting	`NEVER`	
Dynamic specification	yes	
Specification within session	no	

Notes:

1. `SRVTERM` is specified on the server side only.
2. For information on Natural RPC, see the *Natural RPC (Remote Procedure Call)* documentation.

185 SRVUSER - User ID for RPC Server Registry

This Natural profile parameter specifies the user ID needed to register a Natural RPC server on the node specified with the profile parameter SVRNODE.

Note: 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.

Possible settings	*user-ID*	Valid user ID. 1 to16 characters.
	*USER	If SRVUSER is set to *USER, the Natural server uses the current Natural user ID (see system variable *USER) to logon to the node.
	'*NSC'	If SRVUSER is set to '*NSC' and Natural Security is installed, the Natural server uses the current Natural user ID (see system variable *USER) and the password defined for this user ID in Natural Security to logon to the node.
Default setting	*timestamp*	If the user ID is omitted, the timestamp will be used.
Dynamic specification	yes	
Specification within session	no	

Notes:

1. SRVUSER is specified on the server side only.
2. For information on Natural RPC, see the *Natural RPC (Remote Procedure Call)* documentation.

186 SRVWAIT - Wait Time of RPC Server

This Natural profile parameter specifies the number of seconds the server is to wait for a Natural RPC client request.

Possible settings	`0`
	or
	`1 - 32767`
		Wait time in seconds.
		Note:
		1. If this time is exceeded, the RPC server is informed by the node to which the RPC server has registered. The RPC server writes a corresponding message to the Natural RPC server trace file, and continues to wait for an RPC client request.
		2. If TCP/IP is used to communicate with the node, a non-zero value will also avoid an indefinite wait in TCP/IP if the node cannot respond for any reason.

Default setting	`0`	Unlimited wait time. **Note:** In case of an EntireX Broker node, the wait time is set to the `SERVER-NONACT` value of the corresponding Entirex Broker attribute file.
Dynamic specification	yes	
Specification within session	no	

Notes:

1. `SRVWAIT` is specified on the server side only.
2. For information on Natural RPC, see the *Natural RPC (Remote Procedure Call)* documentation.

187 SSIZE - Size of Source Area Allocated by the Editors

This Natural profile parameter determines the maximum size of the Natural source area, which will be dynamically allocated by the Natural editors.

Possible settings	`1 - 100`	Maximum size of the Natural source area in MB.
Default setting	`1`	
Dynamic specification	no	
Specification within session	no	

Note: The maximum size for one Natural source member is 1 MB (independent of `SSIZE`).

188 STACK - Place Data/Commands on the Stack

This Natural profile parameter is used to place data/commands on the Natural stack.

Note:

The profile parameter `STACK` is used to place data/commands on the Natural command stack.

Possible settings	any character string
Default setting	none
Dynamic specification	yes
Specification within session	no

Notes:

1. The amount of data to be passed with this parameter is limited to 512 bytes. If this limit is exceeded, a corresponding error message is returned.
2. 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.
3. 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.
4. 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. 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 (:).
- A semicolon (;) is used to delimit multiple commands.

Examples:

```
LOGON:USER1;UCMD1 A,B;UCMD2 C,D:E;FIN
```

Logs on to the library `USER1`, executes the commands `UCMD1` and `UCMD2` providing the corresponding input data, and ends the Natural session.

```
CMD DATA:DATA;CMD
```

Places commands and 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.

If specified dynamically, the character string provided as data for the `STACK` parameter must be enclosed in apostrophes; if the character string contains multiple commands, it must also be enclosed in parentheses, for example:

```
STACK='(LOGON SYSTEM;UCMND)'
```

Logs on to the library `SYSTEM` and executes the command `UCMND`.

189 STARTUP - Program Name for System Variable

*STARTUP

This Natural profile parameter specifies a program name for the Natural system variable `*STARTUP`.

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

Notes:

1. 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`.
2. 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.

190 STEPLIB - Initial Setting for *STEPLIB System Variable

This Natural profile parameter specifies the initial setting for the system variable `*STEPLIB`.

Possible settings	1 - 8 characters	Any valid library name.
Default setting	`SYSTEM`	
Dynamic specification	yes	
Specification within session	no	
Application programming interface	`USR1005N`	See *SYSEXT - Natural Application Programming Interfaces* in the *Utilities* documentation.

Notes:

1. The content of the system variable `*STEPLIB` influences the order in which objects are searched in the system file.
2. Using the Configuration Utility, you can define additional steplibs that can be searched for objects which cannot be found in the current library. See *Steplibs* in the *Configuration Utility* documentation.
3. See also *Steplibs* and *Search Sequence for Object Execution* in the *Using Natural Studio* documentation.

191 SUBCHAR - Substitution Character for Default Code Page

This Natural profile parameter allows you to specify the substitution character for the default code page. The substitution character is automatically inserted whenever the conversion of a Unicode character into the current default code page (see profile parameter `CP`) fails and the profile parameter `CPCVERR` is set to `OFF`.

Possible settings	*n*	Substitution character.
	`OFF`	If `OFF` is specified, the ICU default substitution character is used. **Note:** For further information, see also *Profile Parameters* in the *`Unicode and Code Page Support`* documentation.
Default setting	`OFF`	
Dynamic specification	no	
Specification within session	no	

192 SUTF8 - UTF-8 Format for Sources

This Natural profile parameter specifies the default format to be used when Natural sources are saved.

Possible settings	ON	The default format for saving Natural sources is UTF-8. **Note:** 1. All sources will be saved in UTF-8 format, which assures that the source content does not depend on the installed system code page. 2. If the profile parameter SRETAIN is also set to ON, only newly created sources will be saved in UTF-8 format. Existing sources will then be saved in the original encoding, if possible.
	OFF	The default format for saving Natural sources is “code page”.
Default setting	OFF	
Dynamic specification	no	
Specification within session	no	

Note: See also *Profile Parameters* in the *Unicode and Code Page Support* documentation.

193 SYMGEN - Generate Symbol Table

This Natural profile and session parameter specifies whether a symbol table is to be generated or not.

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	none	
Applicable command	GLOBALS	

Notes:

1. 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.
2. Within a session, the profile parameter SYMGEN can be overridden by the session parameter SYMGEN.

194 SYNERR - Control of Syntax Errors

This Natural profile parameter specifies 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	
Application programming interface	USR4007N	See *SYSEXT - Natural Application Programming Interfaces* in the *Utilities* documentation.

Notes:

1. The error transaction program is defined either with the profile parameter ETA or by a user program by way of assignment to the system variable *ERROR-TA or, if Natural Security is installed, within the Natural Security library profile; see *Components of a Library Profile* in the *Natural Security* documentation.
2. For further information, see *Using an Error Transaction Program* in the *Programming Guide*.

195 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.
Default setting	none	
Specification within session	yes	
Applicable statements	`FORMAT`	Parameter may be specified dynamically with the `FORMAT` statement.
	`DISPLAY`	Parameter may be specified at statement level and/or at element level.
Applicable command	none	

Notes:

1. 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.
2. The parameter `TC` can also be used with U format fields. For information on Unicode format, see also *Unicode and Code Page Support in the Natural Programming Language*, *Session Parameters*, *EMU, ICU, LCU, TCU versus EM, IC, LC, TC*.
3. See also *Parameters to Influence the Output of Fields* in the *Programming Guide*.

Examples:

```
FORMAT TC=*
DISPLAY (TC='*B*')
```

196 TCU - Unicode 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.
Default setting	none	
Specification within session	yes	
Applicable statements	`FORMAT`	Parameter may be specified dynamically with the `FORMAT` statement.
	`DISPLAY`	Parameter may be specified at statement level and/or at element level.
Applicable command	none	

Notes:

1. 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.
2. The session parameter `TCU` is identical to the session parameter `TC`. The difference is that the trailing characters are always stored in Unicode format. This allows you to specify trailing characters with mixed characters from different code pages, and assures that always the correct character is displayed independent of the installed system code page.

See also:

- *Parameters to Influence the Output of Fields* in the *Programming Guide*
- *Unicode and Code Page Support in the Natural Programming Language*, *Session Parameters*, *EMU, ICU, LCU, TCU versus EM, IC, LC, TC*

197 TD - Time Differential

This Natural profile parameter specifies 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.

Possible settings	`-23,59` to `23,59`	The notation *`hours,minutes`* is used to add/subtract the specified time to/from the physical machine time to set the time/date to be used by Natural; *`minutes`* (if specified) must be `00-59`.
Default setting	`0,0`	
Dynamic specification	yes	
Specification within session	no	

Note: This parameter is applicable in an environment in which remote nodes are being used in a computer network.

Examples:

```
TD=6            (6 hours ahead)
TD=-11          (11 hours behind)
TD=(5,30)       (5 hours and 30 minutes ahead)
TD=(-6,30)      (6 hours and 30 minutes behind)
```

198 TF - Translation of Database ID/File Number

- TF Parameter Syntax 451

This Natural profile parameter is used to translate the database ID/file number of a production database into the database ID/file number of a test database.

Caution: This parameter applies to user files only. It does not apply to system files.

Possible settings	*production-DBID*	0 - 65535, except 255, or can be an asterisk (*) which stands for all DBIDs. **Note:** DBID 255 is reserved for logical system files, see profile parameter LFILE.
	production-FNR	1 - 65535, or an asterisk (*) which stands for all FNRs.
	test-DBID	0 - 65535, except 255.
	test-FNR	1 - 65535
Default setting	none	
Dynamic specification	yes	This parameter can be specified dynamically and in the Natural parameter file NATPARM.
Specification within session	no	

Notes:

1. The translation of file number 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.
2. The transfer is done by a translation of the application's database identification (DBID) and file number (FNR): with the TF parameter, you specify the production DBID/FNR and the test DBID/FNR. The Natural objects are then cataloged with the production DBID/FNR, but whenever a database access is executed, the production DBID/FNR is translated into the test DBID/FNR; that is, the test database is used. This means that testing can take place in the actual production environment, but not with production data.
3. The asterisk (*) notation for *production-DBID* and *production-FNR* is mutually exclusive.

TF Parameter Syntax

The parameter is specified as follows:

```
TF=(production-DBID,production-FNR,test-DBID,test-FNR)
```

The TF parameter can be specified several times. Existing specifications are displayed in a corresponding list box.

If multiple TF parameters are specified, the following applies:

- If one TF specification exists for a specific DBID/FNR and one with a wildcard also matches the specified FNR, the one with the wildcard is used.

  For example: The TF settings (5,2,3,7) and (5,*,7,8) translate 5/2 and give 7/8.
- If one TF specification exists for a specific DBID/FNR and one with a wildcard also matches the specified DBID, the one with the exact specification is used.

  For example: The TF settings (5,4,3,7) and (*,4,3,1) translate 5/4 and give 3/7.

Notes:

1. Production and test databases must be of the same type (Adabas/Adabas, for example).
2. If the database type is not specified in NATPARM, the Adabas is used as default type; that is, SQL must be specified explicitly.
3. For SQL, the file number must always be set to 1. The DBID must be in the range of 0-254.
4. The profile parameter UDB (User Database ID) is evaluated before the TF parameter is evaluated.

TF Parameter Syntax

The TF parameter is specified as follows:

```
TF=(production-DBID,production-FNR,test-DBID,test-FNR)
```

The TF parameter can be specified several times. Existing specifications are displayed in a corresponding list box.

If multiple TF parameters are specified, the following applies:

- If one TF specification exists for a specific DBID/FNR and one with a wildcard also matches the specified FNR, the one with the wildcard is used.

  For example: The TF settings (5,2,3,7) and (5,*,7,8) translate 5/2 and give 7/8.
- If one TF specification exists for a specific DBID/FNR and one with a wildcard also matches the specified DBID, the one with the exact specification is used.

  For example: The TF settings (5,4,3,7) and (*,4,3,1) translate 5/4 and give 3/7.

Notes:

1. Production and test databases must be of the same type (Adabas/Adabas, for example).
2. If the database type is not specified in `NATPARM`, the Adabas is used as default type; that is, SQL must be specified explicitly.
3. For SQL, the file number must always be set to `1`. The DBID must be in the range of `0-254`.
4. The profile parameter `UDB` (User Database ID) is evaluated before the `TF` parameter is evaluated.

199 THSEP - Dynamic Thousands Separator

This Natural profile parameter is used to enable or disable the use of dynamic thousands separators in edit masks at compilation time.

Possible settings	ON	Dynamic thousands separators are used. **Note:** Every dynamic thousands separator that is not part of a string literal is replaced at runtime with the thousands separator character defined with THSEPCH.
	OFF	Dynamic thousands separators are not used. **Note:** Thousands separators in the edit mask are treated as literal and displayed unchanged at runtime. This is the compatibility setting.
Default setting	OFF	
Dynamic specification	yes	
Specification within session	yes	
Applicable statements	OPTIONS	
Applicable command	THSEP	Option of system command COMPOPT.

Notes:

1. At runtime the dynamic thousands separators are replaced by the value (thousands separator character) of the profile and session parameter THSEPCH.
2. In the Natural source, the dynamic thousands separator is either a comma (,) or a period (.), depending on the current setting of the profile and session parameter DC (decimal character). If a comma is specified, then the dynamic thousands separator is a period, otherwise it is a comma.
3. Fields in Unicode format should not be redefined as alphanumeric (A) or numeric (N) fields.

See also:

- Profile parameter `THSEPCH` in the *Parameter Reference*.
- *Customizing Separator Character Displays* in the *Programming Guide*.

200 THSEPCH - Thousands Separator Character

This Natural profile and session parameter is used to specify the character to be used as thousands separator at runtime. Then the thousands separator character replaces the dynamic thousands separators in edit masks.

Possible settings	any character	At runtime, the dynamic thousands separator is replaced with this character. **Note:** 1. If the thousands separator character is to be a comma, it must be enclosed in quotes, that is, `THSEPCH=','` when using the dynamic parameter facility, because a comma is used to separate individual parameters. 2. If the thousands separator character is to be a quote, it must be specified as two quotes enclosed in quotes, that is, `THSEPCH='''''`.
Default setting	, (comma)	**Note:** By default, a comma is used as thousands separator.
Dynamic specification	yes	
Specification within session	yes	
Applicable statements	none	
Applicable command	`GLOBALS`	

Note: In the Natural source, the dynamic thousands separator is always represented by a comma (,) or a period (.).

See also:

- Profile parameter `THSEP` in the *Parameter Reference* documentation.
- Option `THSEP` of system command `COMPOPT` in the *System Commands* documentation.

- *Customizing Separator Character Displays* in the *Programming Guide*.

201 TIMEOUT - Wait Time for RPC Server Response

This Natural profile and session parameter specifies the number of seconds the client is to wait for an RPC server response.

Possible settings	`0 -32767`	Timeout in seconds. **Note:** If this time is exceeded, the remote procedure call will be terminated with a corresponding error message.
Default setting	`55`	
Dynamic specification	yes	
Specification within session	yes	The setting of this parameter can be changed using the `GLOBALS` system command.

Notes:

1. `TIMEOUT` is specified on the client side only.
2. For information on Natural RPC, see the *Natural RPC (Remote Procedure Call)* documentation.

202 TMPSORTUNIQ - Unique Names for Temporary Sort Work Files

If this profile parameter is specified, Natural assigns a unique file name to the temporary work file generated during the sort operation. Any values specified with the `ETID` profile parameter are not included in the file name.

By default, Natural creates work file names with the values of the `ETID` setting and the user ID embedded. This results in file names which are unique to a particular Natural session, but 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 `TMPSORTUNIQ` is specified, Natural generates a unique name for the temporary sort work file.
Default setting	not specified	
Dynamic specification	yes	
Specification within session	no	

203 TQ - Translate Quotation Marks

This parameter has been replaced by the Natural profile parameter TQMARK.

204 TQMARK - Translate Quotation Marks

This Natural profile parameter 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	yes	
Applicable statements	OPTIONS	
Applicable command	TQMARK	Option of system command COMPOPT

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 TQMARK parameter.

Example 1 (TQMARK=ON):

```
WRITE 'THERE"S A QUOTATION MARK'
```

is displayed as: THERE'S A QUOTATION MARK

Example 2 (TQMARK=OFF):

```
WRITE 'THERE"S A QUOTATION MARK'
```

is displayed as: `THERE"S A QUOTATION MARK`

Example 3 (TQMARK=ON or OFF):

```
WRITE 'DOUBLE APOSTROPHES'' OUTPUT IS A SINGLE APOSTROPHE'
```

is displayed as: `DOUBLE APOSTROPHES' OUTPUT IS A SINGLE APOSTROPHE`

205 TRACE - Define Trace Level for Natural RPC Servers

This Natural profile parameter activates the Natural RPC trace facility and determines the trace level to be used.

Possible settings	0	Nothing is traced.
	1	Only messages (inclusive Natural errors) are traced.
	(1,E)	Messages are traced, but only when an error occurs.
	2	All messages and data from/to client are traced.
	(2,E)	Messages and data from/to client are traced, but only when an error occurs.
	3 - 9	The values 3 - 9 displayed in the selection box of the Configuration Utility are also accepted. These values are for future use and behave like TRACE=2.
Default setting	0	
Dynamic specification	yes	
Specification within session	no	

Notes:

1. TRACE is specified on the server side only.
2. For further information, see *Using the Server Trace Facility* p.p. in the *Natural RPC (Remote Procedure Call)* documentation.
3. For information on Natural RPC, see the *Natural RPC (Remote Procedure Call)* documentation.

206 TRANSP - Server Transport Protocol

This Natural profile parameter specifies which server transport protocol is used. If ACI is used, you can additionally specify the transport method.

Note: 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.

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	

Notes:

1. `TRANSP` is specified on the server side only.
2. For information on Natural RPC, see the *Natural RPC (Remote Procedure Call)* documentation.

207 TRYALT - Try Alternative Server Address

This Natural profile and session parameter specifies whether an RPC client should try to execute an RPC request on an alternative server or not.

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	The setting of this parameter can be changed using the system command GLOBALS.

Notes:

1. TRYALT is specified on the client side only.
2. For further information, see *Specifying RPC Server Addresses* in the *Natural RPC (Remote Procedure Call)* documentation.
3. For information on Natural RPC, see the *Natural RPC (Remote Procedure Call)* documentation.

208 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.
	`OFF`	
Default setting	`-`	Hyphen (-).
Specification within session	yes	

Applicable statements	`DISPLAY`
	`FORMAT`
	`WRITE TITLE`
	`WRITE TRAILER`	

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=+)
```

Note: See also *Underlining Character for Titles and Headers - UC Parameter* in the *Programming Guide*.

209 UDB - User Database ID

This Natural profile parameter specifies the DBID to be used for a database access at runtime. This database ID specified with the `UDB` parameter replaces DBID 0 when Natural objects are executed.

Possible settings	`0` or `1 - 65535`, except `255`	Valid database ID. **Note:** Database ID 255 is reserved for logical system files, see profile parameter `LFILE`.
Default setting	1	
Dynamic specification	yes	
Specification within session	no	
Application programming interface	`USR1005N`	See *SYSEXT - Natural Application Programming Interfaces* in the *Utilities* documentation. * Recommended.
	`USR1040N` *	

Notes:

1. The database type of DBID 0, which is specified in the DBMS assignments table of the Configuration Utility, and the database types of the DBID specified with the `UDB` parameter must be the same: ADA/ADA, ADA2/ADA2, SQL/SQL. As an exception, the combination ADA/ADA2 is possible. The first type is the database type of DBID 0 and the second type is the database type of the DBID specified with the `UDB` parameter.
2. If the DBID in the DDM used is 0, then the database type is taken from the DBMS assignments table entry `DBID=0` at compilation time, whereas the database type of the DBID specified with the `UDB` parameter is used only at runtime.
3. If no DBID 0 is specified in the DBMS assignments table, then the default database type is set to `ADA`.

4. If no DBID is specified in the DDM used, the DBID specified with the `UDB` profile parameter determines which database is accessed. If so, the `UDB` profile parameter must be set to a valid DB number.
5. `UDB` is also used to specify the DBID for stowing transaction data if the profile parameter `ETDB` is not specified.

210 ULANG - User Language

This Natural profile parameter specifies the language to be used for date edit masks, system messages, user messages, help texts, helproutines, and multi-lingual maps. The setting is used to set the Natural system variable `*LANGUAGE`.

Possible settings	`1 - 60`	Natural language code. **Note:** 1. For example, 1 is assigned to English, 2 is assigned to German, 3 is assigned to French. 2. 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	
Application programming interface	`USR1005N`	See *SYSEXT - Natural Application Programming Interfaces* in the *Utilities* documentation.

Notes:

1. Within the session, the language code can be specified using the terminal command `%L=`.
2. *Screen Design*, *Skill-Sensitive User Interfaces* in the *Programming Guide*.

211 USEDIC - Common Logical Name for Dictionary Servers

This Natural profile parameter is used to specify a common logical name for dictionary servers defined with Natural RPC to enable remote dictionary access on a z/OS or Linux 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	

See also *Dictionary Server Assignments* in the *Configuration Utility* documentation.

212 USER - User ID

- Settings 480

This Natural profile parameter is used to enter a user ID.

Settings

Possible settings	Any valid user ID
Default setting	blank
Dynamic specification	no
Specification within session	no

Notes:

1. If the default setting is used, the login user ID from the operating system applies.
2. Under Natural Security, this profile parameter is ignored.

213 USEREP - Repository Usage

This Natural profile parameter 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	

214 USIZE - Size of User Buffer

This Natural profile parameter specifies the size of the user buffer in virtual memory. The user buffer contains all data dynamically allocated by Natural.

Possible settings	`20 - 1024`	Buffer size in MB.
	`0`	With `USIZE=0`, the memory capacity will be unrestricted.
Default setting	`20`	
Dynamic specification	no	
Specification within session	no	
Application programming interface	USR9301N	See SYSEXT - Natural Application Programming Interfaces in the Utilities documentation.

215 WEBIO - Use Natural Web I/O Interface

This Natural profile parameter defines whether the Natural input and output (I/O) remains unchanged (that is, terminal emulation in case of SPoD) or whether the Natural Web I/O Interface is used.

Possible settings	`ON`	I/O via Natural Web I/O Interface.
	`OFF`	I/O remains unchanged (terminal emulation). If the application is running on a Natural Development Server on Windows, this setting is ignored. In this case, the Natural Web I/O Interface is always used.
Default setting	`OFF`	
Dynamic specification	yes	
Specification within session	no	

Notes:

1. The Natural Web I/O Interface displays the input and output data in a web browser. On platforms other than Windows, the Natural Web I/O Interface can only be used to display and enter U format fields with characters which are not contained in the current code page.
2. The parameter `WEBIO` can only be used when Natural is running as a server, not in an interactive Natural session.
3. The parameter `WEBIO` is not compatible with the parameter `BATCHMODE`. If the parameter `BATCHMODE` is set, `WEBIO=OFF` is assumed.
4. See also *Profile Parameters* in the *`Unicode and Code Page Support`* documentation.

216 WFOPFA - Opening of Work Files

This Natural profile parameter 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	

Note: `WFOPFA=OFF` only affects main programs; for routines, `WFOPFA=ON` always applies.

217 WH - Wait for Record in Hold Status

This Natural profile and session parameter 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.

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`	
Application programming interface	`USR1005N`	See *SYSEXT - Natural Application Programming Interfaces* in the *Utilities* documentation.

Notes:

1. This Natural profile and session parameter applies to Adabas databases only.
2. Within a Natural session, the profile parameter `WH` can be overridden by the session parameter `WH`.
3. 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.
4. Under Natural Security, the setting of this parameter can be overridden by the Session Parameters option of the *Library Profile*.

5. For a `READ` or `FIND` statement using the `SKIP RECORDS IN HOLD` option, database access is always executed as if `WH=OFF` is set. If a user attempts to read a record that was placed into hold by another user, this record is skipped and processing continues with the next record in the read sequence. An error message is not returned in this case.

218 WORK - Work-File Assignments

- Settings 492

This Natural profile parameter defines the number of work files to be used during the session.

Settings

Possible settings	0 or 1 - 32	Number of work files.
Default setting	32	
Dynamic specification	no	
Specification within session	no	

WORK=*nn*

If WORK is set to *nn* (in the range 1 - 32), this setting defines the highest work file number which is available for the Natural session.

See also *Work Files* in the *Operations* documentation.

219 XREF - Creation of XRef Data for Natural

- Possibilities of Setting the XREF Parameter 494
- XRef Data Generation 495
- Extended XRef Data Generation (For Internal Use Only) 495

This Natural profile and session parameter is used to enable/disable the creation of XRef data for Natural. This parameter also determines how XRef data are treated when Natural members are processed with the Natural utilities SYSMAIN or INPL or with the Object Handler.

Possible settings	ON	XRef data are generated in the cases described above. Documentation premise is not checked.
	OFF	XRef data are not generated. Documentation premise is not checked.
	FORCE	A Natural object can only be cataloged if a documentation object already exists for this implementation object. XRef data are generated in the cases described above.
	DOC	A Natural object can only be cataloged if a documentation object already exists for this object. XRef data are not generated.
Default setting	OFF	
Dynamic specification	yes	
Specification within session	yes	
Applicable statements	none	
Applicable commands	GLOBALS, XREF	

The following topics are covered below:

Possibilities of Setting the XREF Parameter

There are different ways to set the Natural XREF parameter:

- In the Natural parameter file.
- As a dynamic parameter when starting a Natural session.
- In Natural Security. If Natural Security has been used to set the XREF parameter, the XREF command may only be used to enforce this setting (by changing from ON to FORCE, from OFF to ON or FORCE).
- With the Natural XREF command. If Natural Security is not installed, the XREF parameter is usually set with the Natural XREF command. The Natural command XREF ? displays the current setting of the XREF parameter.

XRef Data Generation

XRef data is generated in two cases:

- The Natural compiler writes XRef data for Natural programs and data areas when these are cataloged (provided that the `XREF` parameter has been set to either `ON` or `FORCE`, see below).
- Natural Security writes XRef data for programs that are used as Startup, Restart or Error-Transaction in an application or as a special link if the `XREF` parameter is set to `ON` or `FORCE` in the application's Natural Security definition and a user system file is defined for the application.

The `XREF` parameter controls the compilation in two aspects:

- generation of XRef data in the cases described above and
- fulfilment of premise to document implementation objects. The adherence to this premise can be ensured by allowing the completion of the catalog operation only for objects that are documented in the Predict `FDIC` system file or in the development server file used in Natural Single Point of Development (SPoD).

Note: In a NaturalONE development environment, XRef data is only generated for libraries in shared mode. If there is an existing private mode library with XRef data, it is recommended that you recatalog the library to avoid redundant XRef data.

Extended XRef Data Generation (For Internal Use Only)

The extended `XREF` parameter is reserved for internal use by Natural.

220 YD - Year Differential

This Natural profile parameter 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. If the profile parameter MAXYEAR is set to 9999, the upper value limit is 7999.
Default setting	0	
Dynamic specification	yes	
Specification within session	no	

Notes:

1. 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.
2. The year resulting from the sum of the profile parameters TD, DD and YD must be in the range of 1582 through 2699. If the profile parameter MAXYEAR is set to 9999, the upper year limit is 7999

221 YSLW - Year Sliding or Fixed Window

- Examples of YSLW Parameter 501

This Natural profile parameter specifies the range of years covered by the "year sliding window" or "year fixed window".

Note: The sliding-window or "year 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. See *Example of a Sliding Window*.
	Fixed Window	`1582-2600`	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. See *Example of a Fixed Window*.
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 date 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 *Programming Guide*.

Examples of YSLW Parameter

Example of a Sliding Window

If the current year is 2014 and you specify `YSLW=40`, the sliding window will cover the years 1974 to 2073. A 2-digit year setting *nn* from `74` to `99` is then interpreted accordingly as `19`*nn*, while a 2-digit year setting *nn* from `00` to `73` is interpreted as `20`*nn*.

See also the examples under *Year Sliding Window - YSLW Parameter* and *Combinations of DFSTACK and YSLW* in the *Programming Guide*.

Example of a Fixed Window

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*.

222 ZD - Zero-Division Check

This Natural profile and session parameter specifies the action to be taken when an attempt is made to perform a division operation in which the divisor is `0`.

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`	
Application programming interface	`USR1005N`	See *SYSEXT - Natural Application Programming Interfaces* in the *Utilities* documentation.

Notes:

1. Within a Natural session, the profile parameter `ZD` can be overridden by the session parameter `ZD`.
2. Under Natural Security, the setting of this parameter can be overridden by the *Session Parameters* option of the Library Profile.

223 ZP - Zero Printing

This Natural profile and session parameter specifies how a field which contains a setting of all zeros is to be output.

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	
Application programming interface	USR1005N	See *SYSEXT - Natural Application Programming Interfaces* in the *Utilities* documentation.

Notes:

1. This Natural profile and session parameter 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.
2. Within a Natural session, the profile parameter ZP can be overridden by the session parameter ZP.
3. See also *Parameters to Influence the Output of Fields* in the *Programming Guide*.