

Natural for Mainframes

Parameter Reference

Version 4.2.6 for Mainframes

October 2009

This document applies to Natural Version 4.2.6 for Mainframes and to all subsequent releases.

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

Copyright © Software AG 1979-2009. All rights reserved.

The name Software AG, webMethods and all Software AG product names are either trademarks or registered trademarks of Software AG and/or Software AG USA, Inc. Other company and product names mentioned herein may be trademarks of their respective owners.

Table of Contents

1 Parameter Reference	1
2 Introduction to Profile Parameters	3
Profile Parameter Usage	4
Common Profile and Session Parameter Descriptions	4
3 Introduction to Session Parameters	5
Session Parameter Usage	6
How to Set Session Parameters	6
Session Parameter Evaluation	8
4 ACIVERS - Define ACI Version for Use with EntireX Broker ACI	9
5 AD - Attribute Definition	11
AD Parameter Syntax	12
Field Representation	13
Field Alignment	13
Field Input/Output Characteristics	14
Interpretation of Alphanumeric Fields	15
Mandatory Input	16
Length of Input Value	16
Field Upper/Lower Case Characteristics	16
Filler Character	17
6 ADAMODE - Adabas Call Interface Mode	19
7 ADANAME - Name of Adabas Link Routine	23
8 ADAPRM - REVIEW/DB Support	25
9 ADASBV - Adabas Security by Setting	27
10 AL - Alphanumeric Length for Output	29
11 ASIZE - Entire System Server Auxiliary Buffer	31
12 ASPSIZE (Internal Use)	33
13 ASYNNAM - Output System ID for Asynchronous Processing	35
14 ATTN - Attention Key Interrupt Support	37
15 AUTO - Automatic Logon	39
16 AUTORPC - Automatic Natural RPC Execution	41
17 BPC64 - Buffer Pool Cache Storage Type	43
18 BPCSIZE - Cache Size for Natural Buffer Pool	45
19 BPI - Buffer Pool Initialization	47
BPI Parameter Syntax	48
NTBPI Macro Syntax	49
Keyword Subparameters	49
Examples of NTBPI Macros	54
Examples of BPI Parameter	55
20 BPLIST - Name of Preload List for Natural Buffer Pool	57
21 BPMETH - Buffer Pool Space Search Algorithm	59
22 BPNAME - Name of Natural Global Buffer Pool	61
23 BPPROP - Global Buffer Pool Propagation	63
24 BPSFI - Object Search First in Buffer Pool	65

25 BPSIZE - Size of Natural Local Buffer Pool	67
26 BPTXT - Size of Text Segments in Natural Buffer Pool	69
27 BSIZE - Size of EntireX Broker Buffer	71
28 BX - Box Definition	73
29 CANCEL - Session Cancellation with Dump	75
30 CC - Error Processing in Batch Mode	77
31 CCTAB - Printer Escape Sequence Definition	79
CCTAB Parameter Syntax	80
NTCCTAB Macro Syntax	82
String Syntax for OPN, CLS, CODE, CS, CSS or CSE	82
Proportional Fonts	83
Examples of NTCCTAB Macros	83
Examples of CCTAB Parameter	83
32 CD - Color Definition	85
33 CDYNAM - Dynamic Loading of Non-Natural Programs	87
34 CF - Character for Terminal Commands	89
35 CFICU - Unicode and Code Page Support	91
CFICU Parameter Syntax	92
NTCFICU Macro Syntax	93
Keyword Subparameters	93
Example of NTCFICU Macro	96
Example of Dynamic Parameter CFICU	96
36 CFWSIZE (Internal Use)	97
37 CLEAR - Processing of CLEAR Key in NEXT Mode	99
38 CM - Command Mode	101
39 CMPO - Compilation Options	103
CMPO Parameter Syntax	104
NTCMPO Macro Syntax	104
Keyword Subparameters	104
Example of CMPO Parameter	105
Example of NTCMPO Macro	105
40 CMPR - General Default Compression Optimization Algorithm	107
41 COMPR - Set RPC Buffer Compression	109
42 CP - Default Code Page Name	111
43 CPCVERR - Code Page Conversion Error	113
44 CPOBJIN - Code Page of Batch Input File	115
45 CPPRINT - Code Page of Batch Output File	117
46 CPRPC - Define Code Page Name	119
47 CPSYNIN - Code Page of Batch Input File for Commands	121
48 CSIZE - Size of Con-nect/Con-form Buffer Area	123
49 CSTATIC - Programs Statically Linked to Natural	125
Example of CSTATIC Parameter	126
Example of NTCSTAT Macro	127
50 CV - Attribute Control Variable	129
51 CVMIN - Control Variable Modified at Input	131

52 DATSIZE - Minimum Size of Buffer for Local Data	133
53 DB - Database Types and Options	135
DB Parameter Syntax	136
NTDB Macro Syntax	137
Possible Database Options	138
Examples of NTDB Macro	139
Examples of DB Parameter	139
54 DB2SIZE - Natural Buffer Area for DB2 or SQL/DS	141
55 DBCLOSE - Database Close at Session End	143
56 DBGERR - Automatic Start of Debugger at Runtime Error	145
57 DBID - Default Database ID of Natural System Files	147
58 DBOPEN - Database Open without ETID	149
59 DBROLL - Database Calls before Session Suspension	151
60 DBUPD - Database Updating	153
61 DC - Character for Decimal Point Notation	155
62 DD - Day Differential	157
63 DELETE - Deletion of Dynamically Loaded Programs	159
64 DF - Date Format	161
65 DFOUT - Date Format for Output	163
66 DFS - Specify RPC Client's Default Server Address	165
67 DFSTACK - Date Format for Stack	167
68 DFTITLE - Output Format of Date in Standard Report Title	169
69 DL - Display Length for Output	171
70 DLISIZE - Size of Natural Buffer Area for DL/I	173
71 DO - Display Order of Output Data	175
72 DS - Define Size of Storage Buffer	177
DS Parameter Syntax	179
NTDS Macro Syntax	179
Table of Buffer Sizes	179
Examples	181
73 DSC - Data-Stream Compression (for 3270-Type Terminals)	183
74 DSIZE - Size of DBLOG Buffer	185
75 DTFORM - Date Format	187
76 DU - Dump Generation	189
77 DUE - Dump for Specific Errors	191
78 DY - Dynamic Attributes	193
DY Parameter Syntax	194
Examples	196
79 DYNPARM - Control Use of Dynamic Parameters	199
DYNPARM Parameter Syntax	201
NTDYNP Macro Syntax	201
Examples	201
80 ECHO - Control Printing of Batch Input Data	203
81 EDBP - Software AG Editor Buffer Pool Definitions	205
EDBP Parameter Syntax	207

NTEDBP Macro Syntax	207
Keyword Subparameters	207
82 EDPSIZE - Size of Software AG Editor Auxiliary Buffer Pool	213
83 EJ - Page Eject	215
84 EM - Edit Mask	217
Syntax	218
Blanks in Edit Masks	219
Default Edit Masks	219
Edit Masks for Numeric Fields	220
Edit Masks for Alphanumeric Fields	223
Edit Masks for Binary Fields - Format B	225
Hexadecimal Edit Masks	225
Edit Masks for Date and Time Fields - Formats D and T	227
Edit Masks for Logical Fields - Format L	232
85 EMFM - Edit Mask Free Mode	235
86 ENDBT - BACKOUT TRANSACTION at Session End	237
87 ENDMSG - Display Session-End Message	239
88 ES - Empty Line Suppression	241
89 ESCAPE - Ignore Terminal Commands %% and %	243
90 ESIZE - Size of User-Buffer Extension Area	245
91 ET - Execution of END/BACKOUT TRANSACTION Statements	247
92 ETA - Error Transaction Program	249
93 ETDB - Database for Transaction Data	251
94 ETEOP - Issue END TRANSACTION at End of Program	253
95 ETID - Adabas User Identification	255
96 ETIO - Issue END TRANSACTION upon Terminal I/O	257
97 ETPSIZE - Size of Entire Transaction Propagator Buffer	259
98 ETRACE - External Trace Function	261
99 ETSYNC - Issue Syncpoint upon End of Transaction/Backout Transaction	263
100 EXCSIZE - Size of Buffer for Natural Expert C Interface	265
101 EXRSIZE - Size of Buffer for Natural Expert Rule Tables	267
102 FAMSTD - Overwriting of Print and Work File Access Method Assignments	269
103 FC - Filler Character for INPUT Statement	271
104 FC - Filler Character for DISPLAY Statement	273
105 FCDP - Filler Character for Dynamically Protected Input Fields	275
106 FDIC - Predict System File	277
107 FL - Floating Point Mantissa Length	279
108 FNAT - Natural System File for System Programs	281
109 FNR - Default File Number of Natural System Files	283
110 FREEGDA - Release GDA in Utility Mode	285
111 FS - Default Format/Length Setting for User-Defined Variables	287
112 FSEC - Natural Security System File	289
113 FSPOOL - Natural Advanced Facilities Spool File	291
114 FUSER - Natural System File for User Programs	293
115 GC - Filler Character for Group Headers	295

116 HC - Header Centering	297
117 HCAM - Hardcopy Access Method	299
118 HCDEST - Hardcopy Output Destination	301
119 HD - Header Definition	303
120 HE - Helproutine	305
HE Parameter Syntax	306
Execution of Helproutines	308
Examples	308
121 HI - Help Character	311
122 HW - Heading Width	313
123 IA - Input Assign Character	315
124 IC - Insertion Character	319
125 ID - Input Delimiter Character	321
126 IKEY - Processing of PA and PF Keys	323
127 IM - Input Mode	325
128 IMSG - Session Initialization Error Messages	327
129 INTENS - Printing of Intensified Fields	329
130 IP - INPUT Prompting Text	331
131 IS - Identical Suppress	333
132 ISIZE - Size of Initialization Buffer	335
133 ITERM - Session Termination in Case of Initialization Error	337
134 ITRACE - Internal Trace Function	339
135 KD - Key Definition	341
136 KEY - Setting Assignments to PA, PF and CLEAR Keys	343
137 LC - Lower to Upper Case Translation	345
138 LC - Leading Characters	347
139 LE - Reaction when Limit for Processing Loop Exceeded	349
140 LFILE - Logical System File Definition	351
LFILE Parameter Syntax	353
NTLFILE Macro Syntax	353
Old NTFIL Macro Syntax	353
Example of LFILE Parameter	353
Example of NTLFILE Macro	354
141 LIBNAM - Name of External Program Load Library	355
142 LOG (Internal Use)	357
143 LOGONRQ - Logon for RPC Server Request Required	359
144 LS - Line Size	361
Profile Parameter LS	362
Session Parameter LS	362
Specification with Statements	363
145 LT - Limit for Processing Loops	365
146 MADIO - Maximum DBMS Calls between Screen I/O Operations	367
147 MAINPR - Override Default Output Report Number	369
148 MAXBUFF - Maximum Buffer Size	371
149 MAXCL - Maximum Number of Program Calls	373

150 MAXROLL - Number of CMROLL Calls before Session Suspension	375
151 MAXYEAR - Maximum Year for Date/Time Values	377
152 MC - Multiple-Value Field Count	379
153 MENU - Menu Mode	381
154 ML - Position of Message Line	383
155 MONSIZE - Size of SYSTP Monitor Buffer	385
156 MP - Maximum Number of Pages of a Report	387
157 MS - Manual Skip	389
158 MSGSF - Display System Error Messages in Short/Full Format	391
159 MT - Maximum CPU Time	393
160 NAFSIZE - Size of Buffer for Natural Advanced Facilities	395
161 NAFUPF - Natural Advanced Facilities User Profile	397
162 NC - Use of Natural System Commands	399
163 NISN (Internal Use)	401
164 NL - Numeric Length for Output	403
165 NTASKS - Number of Server Tasks to be Started	405
166 NUCNAME - Name of Shared Nucleus	407
167 OBJIN - Use of CMOBJIN as Natural Input File	409
168 OPF - Overwriting of Protected Fields by Help routines	411
169 OPRB - Database Open/Close Processing	413
Dynamic OPRB with Natural Security	415
OPRB for VSAM	415
OPRB for Adabas	415
NTOPRB Macro Syntax	417
Examples of NTOPRB Macros	417
170 OPT - Control of Natural Optimizer Compiler	419
OPT Parameter Syntax	420
NTOPT Macro Syntax	420
171 OUTDEST - Output Destination for Asynchronous Processing	421
172 OVSIZE - Storage Thread Overflow Size	423
173 PARM - Alternative Parameter Module	425
174 PC - Control of Personal-Computer Access Method	427
175 PC - Periodic Group Count	429
176 PCNTRL - Print-Control Characters	431
177 PD - Limit of Pages for NATPAGE	433
178 PLOG - Logging of Dynamic Parameters	435
179 PLUGIN - Enable the Natural Plug-In Components	437
180 PM - Print Mode	439
Profile Parameter PM	440
Session Parameter PM	441
181 POS22 - Version 2.2 Algorithm for POS System Function	443
182 PRINT - Print File Assignments	445
PRINT Parameter Syntax	447
NTPRINT Macro Syntax	447
Keyword Subparameters for All Environments	448

Keyword Subparameters for AM=STD in All Environments	452
Keyword Subparameters for AM=STD in z/OS Environments	455
Keyword Subparameters for AM=STD in z/VSE Environments	457
Keyword Subparameters for AM=STD in BS2000/OSD Environments	458
Keyword Subparameters for AM=CICS	458
Keyword Subparameters for AM=COMP (Com-plete)	459
Keyword Subparameters for AM=SMARTS (Com-plete)	460
Keyword Subparameters for AM=IMS	460
Keyword Subparameters for DEFINE PRINTER Statement	461
183 PROFILE - Activate Dynamic Parameter Profile	463
184 PROGRAM - Non-Natural Program Receiving Control after Termination	465
185 PS - Page Size for Natural Reports	467
186 PSEUDO - CICS Pseudo-Conversational Mode	469
187 RCA - Resolve Addresses of Static Non-Natural Programs	471
188 RCALIAS - External Name Definition for Non-Natural Programs	473
RCALIAS Parameter Syntax	474
NTALIAS Macro Syntax	475
Examples of NTALIAS Macro	475
Example of RCALIAS Parameter	475
189 RCFIND - Handling of Response Code 113 for FIND Statement	477
190 RCGET - Handling of Response Code 113 for GET Statement	479
191 RDACT - (Internal Use)	481
192 RDCEXIT - Define Natural Data Collector User Exits	483
193 RDCSIZE - Size of Buffer for the Natural Data Collector	485
194 RDNODE (Internal Use)	487
195 RDPORT (Internal Use)	489
196 RDS - Define Remote Directory Server	491
197 READER - z/VSE System Logical Units for Input	493
198 RECAT - Dynamic Recataloging	495
199 REINP - Issue Internal REINPUT Statement for Invalid Data	497
200 RELO - Storage Thread Relocation	499
201 RFILE - File for Recordings	501
202 RI - Release ISNs	503
203 RJESIZE - Initial Size of NATRJE Buffer	505
204 RM - Retransmit Modified Fields	507
205 ROSY - Read-Only Access to System Files	509
206 RPC - Remote-Procedure-Call Settings	511
RPC Parameter Syntax	512
NTRPC Macro Syntax	512
Keyword Subparameters	512
RPC Parameter Example	513
NTRPC Macro Example	513
207 RPCSDIR - Library for Service Directory	515
208 RPCSIZE - Size of Buffer Used by Natural RPC	517
209 RPCUCT - Translate Subprogram Name into Upper Case	519

210 RUNSIZE - Size of Runtime Buffer	521
211 SA - Sound Terminal Alarm	523
212 SB - Selection Box	525
Syntactical Considerations	526
Runtime Considerations	527
Features	528
Restrictions	530
213 SCTAB - Scanner Characters	531
SCTAB Parameter Syntax	532
NTSCTAB Macro Syntax	533
Example of NTSCTAB Macro	533
Example of SCTAB Parameter	534
214 SENDER - Screen Output Destination for Asynchronous Processing	535
215 SERVER - Start Natural Session as an RPC Server Session	537
216 SF - Spacing Factor	539
217 SG - Sign Position	541
218 SI - Shift-In Code for Double-Byte Character Set	543
219 SKEY - Storage Protection Key	545
220 SL - Source Line Length	547
221 SLOCK - Source Locking	549
222 SM - Programming in Structured Mode	551
223 SO - Shift-Out Code for Double-Byte Character Set	553
224 SORT - Control of Sort Program	555
SORT Parameter Syntax	556
NTSORT Macro Syntax	556
Keyword Subparameters	557
225 SOSI - Shift-Out/Shift-In Codes for Double-Byte Character Set	561
SOSI Parameter Syntax	562
Positional Subparameters	563
Conversion of Logical Shift-Out/Shift-In Characters	564
Automatic Adaptation of Translation Tables	564
Compatibility of SOSI Profile Parameter and Obsolete SO and SI Profile Parameters	564
SOSI Parameter Examples	565
226 SRETAIN - Retain Source Format	567
227 SRVCMIT - Server Commit Time	569
228 SRVNAME - Name of RPC Server	571
229 SRVNODE - Name of Node	573
230 SRVTERM - Server Termination Event	575
231 SRVUSER - User ID for RPC Server Registry	577
232 SRVWAIT - Wait Time of RPC Server	579
233 SSIZE - Size of Source Area Allocated by the Editors	581
234 STACK - Place Data/Commands on the Stack	583
235 STACKD - Stack Delimiter Character	585
236 STEPLIB - Additional Steplib Library	587




237 SUBSID - Subsystem ID under z/OS and z/VSE	589
238 SYNERR - Control of Syntax Errors	591
239 SYS - Define and Activate a Set of Dynamic Profile Parameters	593
SYS Parameter Syntax	594
NTSYS Macro Syntax	595
Example of NTSYS Macro	595
240 SYSCIP - Adabas Cipher Key for Natural System Files	597
241 SYSPSW - Adabas Password for Natural System Files	599
242 TAB - Standard Output Character Translation	601
TAB Parameter Syntax	602
NTTAB Syntax	602
Example of NTTAB Macro	603
Example of TAB Parameter	603
243 TAB1 - Alternative Output Translation	605
TAB1 Parameter Syntax	606
NTTAB1 Macro Syntax	606
Example of NTTAB1 Macro	607
Example of TAB1 Parameter	607
244 TAB2 - Alternative Input Translation	609
TAB2 Parameter Syntax	610
NTTAB2 Macro Syntax	610
Example of NTTAB2 Macro	611
Example of TAB2 Parameter	611
245 TABA1 - EBCDIC-to-ASCII Translation	613
TABA1 Parameter Syntax	614
NTTABA1 Macro Syntax	615
Example of NTTABA1 Macro	615
Example of TABA1 Parameter	615
246 TABA2 - ASCII-to-EBCDIC Translation	617
TABA2 Parameter Syntax	618
NTTABA2 Macro Syntax	619
Example of NTTABA2 Macro	619
Example of TABA2 Parameter	619
247 TABL - SYS Library Output Translation	621
TABL Parameter Syntax	622
NTTABL Macro Syntax	622
Example of NTTABL Macro	623
Example of TABL Parameter	623
248 TC - Trailing Characters	625
249 TD - Time Differential	627
250 TF - Translation of Database ID/File Number	629
TF Parameter Syntax	631
NTTF Macro Syntax	631
Example of TF Parameter	631
Example of NTTF Macro	631

251 THSEPOCH - Thousands Separator Character	633
252 TIMEOUT - Wait Time for RPC Server Response	635
253 TMODEL - IBM 3270 Terminal Model	637
254 TPF (Internal Use)	639
255 TQ - Translate Quotation Marks	641
256 TRACE - Define Trace Level for Natural RPC Servers	643
257 TRACE - Define Components to be Traced	645
TRACE Parameter Syntax	646
NTTRACE Macro Syntax	647
Example of TRACE Parameter	647
Example of NTTRACE Macro	647
258 TRANSP - Server Transport Protocol	649
259 TRYALT - Try Alternative Server Address	651
260 TS - Translate Output from Programs in System Libraries	653
261 TSIZE - Size of Buffer for Adabas Text Retrieval	655
262 TTYPE - Terminal Type	657
263 UC - Underlining Character	659
264 UDB - User Database ID	661
265 ULANG - User Language	663
266 UPSI - z/VSE User Program Switches	665
267 USER - Restrict Use of Profile Parameter Strings and Modules	667
USER Parameter Syntax	669
NTUSER Macro Syntax	669
Example of NTUSER Macro	669
Example of USER Parameter	669
268 USERBUF (Internal Use)	671
269 UTAB1 - Lower-to-Upper-Case Translation	673
UTAB1 Parameter Syntax	674
NTUTAB1 Macro Syntax	674
Example of NTUTAB1 Macro	675
Example of UTAB1 Parameter	675
270 UTAB2 - Upper-to-Lower-Case Translation	677
UTAB2 Parameter Syntax	678
NTUTAB2 Macro Syntax	678
Example of NTUTAB2 Macro	679
Example of UTAB2 Parameter	679
271 VSIZE - Size of Buffer Area for Natural/VSAM	681
272 WEBIO - Web I/O Interface Screen Rendering	683
WEBIO Parameter Syntax	684
NTWEBIO Macro Syntax	684
Keyword Subparameters	685
273 WH - Wait for Record in Hold Status	687
274 WORK - Work-File Assignments	689
WORK Parameter Syntax	690
NTWORK Macro Syntax	691

Keyword Subparameters for All Environments	692
Keyword Subparameters for AM=STD in All Environments	696
Keyword Subparameters for AM=STD in z/OS Environments	697
Keyword Subparameters for AM=STD in z/VSE Environments	699
Keyword Subparameters for AM=STD in BS2000/OSD Environments	700
Keyword Subparameters for AM=CICS	701
Keyword Subparameters for AM=COMP	702
Keyword Subparameters for AM=SMARTS	702
275 WPSIZE - Sizes of Natural Work Pools	705
276 WSISIZE - Buffer for Natural Workstation Interface	707
277 XML - Activate PARSE XML and REQUEST DOCUMENT Statements	709
XML Parameter Syntax	710
NTXML Macro Syntax	711
Keyword Subparameters	711
Example of NTXML Macro	713
Example of XML Parameter	713
278 XREF - Creation of XRef Data for Natural	715
Extended XRef Data Generation (For Internal Use Only)	717
279 XSIZE - Size of Buffer for User Subsystem	719
280 YD - Year Differential	721
281 YSLW - Year Sliding or Fixed Window	723
282 ZD - Zero-Division Check	725
283 ZP - Zero Printing	727
284 ZSIZE - Size of Entire DB Buffer Area	729
Index	731

1 Parameter Reference

This documentation contains detailed information on the Natural profile and session parameters. It is organized under the following headings:

	Introduction to Profile Parameters	Contains general information on profile parameter usage, including a list of references to related documents.
	Introduction to Session Parameters	Contains general information on session parameter usage and evaluation.
	Parameters in Alphabetical Order	Descriptions of all profile parameters and session parameters in alphabetical order.

General information on using parameters can be found in the *Operations* documentation, see *Profile Parameter Usage*. See also *Using Macros in a Natural Parameter Module*.

2 Introduction to Profile Parameters

■ Profile Parameter Usage	4
■ Common Profile and Session Parameter Descriptions	4

Profile Parameter Usage

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

See the following documents in the *Natural Operations* documentation:

Profile Parameter Usage - Overview

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

Common Profile and Session Parameter Descriptions

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.

3

Introduction to Session Parameters

■ Session Parameter Usage	6
■ How to Set Session Parameters	6
■ Session Parameter Evaluation	8

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 parameter module `NATPARM`, which is set when Natural is installed;
- via dynamic parameters specified when invoking Natural (as described in your *Natural Operations* documentation);
- via the system command `GLOBALS`;
- via a `SET GLOBALS` statement (in reporting mode only);
- via a `FORMAT` statement;
- via parameter specification within statements where parameters also are evaluated, for example, `INPUT`, `DISPLAY`, `WRITE`;
- via terminal commands.

Instead of the parameter values `ON` and `OFF`, you can also specify `T` (true) or `F` (false) respectively.

Changing 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 ACI Version for Use with EntireX Broker

ACI

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

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

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

Possible settings	1 - 9	Single-digit number. The higher the version, the more features are available.
Default setting	2	API Version 2 is used.
Dynamic specification	yes	
Specification within session	no	

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

5

AD - Attribute Definition

■ AD Parameter Syntax	12
■ Field Representation	13
■ Field Alignment	13
■ Field Input/Output Characteristics	14
■ Interpretation of Alphanumeric Fields	15
■ Mandatory Input	16
■ Length of Input Value	16
■ Field Upper/Lower Case Characteristics	16
■ Filler Character	17

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

Related session parameter: [CD - Color Definition](#)

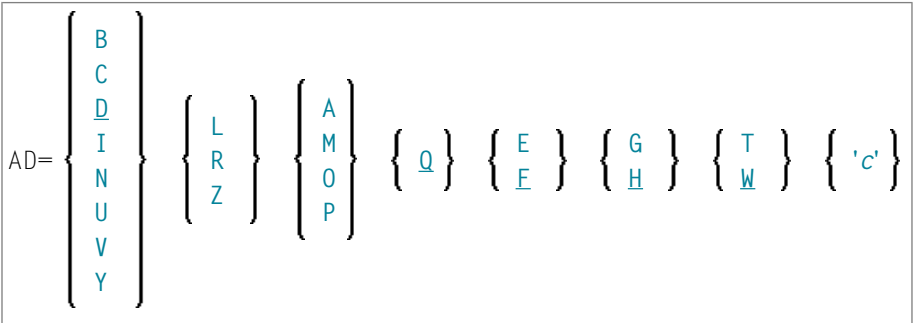
Possible settings	see below	You can specify multiple attributes in any sequence.
Default setting	see below	
Applicable statements:	FORMAT	
	DISPLAY INPUT NEWPAGE WITH TITLE PRINT REINPUT WRITE WRITE TITLE WRITE TRAILER	Parameter may be specified at statement level and/or at element level.
	ASSIGN CALLNAT CALLDBPROC COMPUTE MOVE PERFORM SEND METHOD	Parameter may be specified at element level, however, only the attributes specified in the relevant statement description can be used.
Applicable command:	none	

The following topics are covered below:

AD Parameter Syntax

```
AD=[field-representation] [field-alignment] [field-i/o-characteristics]  
[interpretation-of-alphanumeric-fields] [mandatory-input] [input-value-length]  
[field-upper/lower-case] [filler-character]
```

You can specify multiple attributes in any sequence. Possible values are:



The meaning of the attributes and the possible values are explained below.

Examples:

```
DISPLAY #FIELD A (AD=R)
INPUT #FIELD B (AD=M)
INPUT (AD=IM) #FIELD A #FIELD B
```

Field Representation

Value	Meaning	Statements	Explanation
B	blinking (*)	ASSIGN	The value of the field is displayed blinking.
C	cursive/italic (*)	COMPUTE	The value of the field is displayed cursive/italic.
D	default intensity	MOVE	The value of the field is displayed with normal intensity, that is, not highlighted in any way. This is the default value.
I	intensified	DISPLAY	
N	non-display	FORMAT	
U	underlined	INPUT	
V	reverse video (*)	PRINT	The value of the field is displayed intensified.
Y	dynamic attributes	REINPUT	A value entered in the field will not be displayed.
		WRITE	The value of the field is displayed underlined.
			The value of the field is displayed reverse video.
		INPUT	Attributes are to be controlled via an attribute control variable (Format C).
		DISPLAY	
		PRINT	
		WRITE	

* 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	The value of the field is displayed left-justified. This is the default value for alphanumeric fields.
R	right-justified	FORMAT	
Z	leading zeros	INPUT	The value of the field is displayed right-justified. This is the default value for numeric fields.
		PRINT	
		REINPUT	Numeric values are displayed with leading zeros, right-justified.
		WRITE	

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 PERFORM SEND METHOD	<p>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.</p> <p>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.</p> <p>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.</p> <p>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.</p>
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 PERFORM SEND METHOD	<p>By default, the passed value of a parameter can be changed in the called object (subprogram, stored procedure, subroutine, dialog, method) and the changed value passed back to the calling object, where it overwrites the original value.</p> <p>For a field defined with BY VALUE in the called object's parameter data area, no value is passed back.</p> <p>If, for SEND METHOD, a method is <i>not</i> 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.</p>
0	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.

Value	Meaning	Statements	Explanation
	non-modifiable	CALLNAT CALLDBPROC PERFORM SEND METHOD	<p>If you mark a parameter with AD=0, 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.</p> <p>Internally, AD=0 is processed in the same way as BY VALUE (see the section <i>Parameter Data Definition</i> in the description of the <code>DEFINE DATA</code> statement).</p> <p>If for <code>SEND METHOD</code>, 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 section <i>PARAMETER Clause</i> in the description of the <code>INTERFACE</code> statement).</p> <p>If for <code>SEND METHOD</code>, a method is <i>not</i> implemented in Natural, the behavior depends on the method implementation. The parameter is then passed BY VALUE. Whether the external component accepts a by reference or by value parameter is described in the documentation of the external component. It can also be viewed in the Natural Component Browser.</p>
P	temporarily protected	INPUT REINPUT	Used in conjunction with an attribute control variable (Format C), the DY parameter (dynamic attributes), and the <code>REINPUT</code> statement.

Interpretation of Alphanumeric Fields

Value	Meaning	Statements	Explanation
Q	display alphanumeric field as if it were a numeric field	ASSIGN COMPUTE MOVE DISPLAY FORMAT INPUT PRINT REINPUT WRITE	<p>This attribute is available on mainframe computers only. A corresponding hardware feature is required.</p> <p>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.</p>

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. To make AD=W effective, you have to specify the value ON for the Natural profile parameter LC. This is the default value.

Filler Character


Value	Meaning	Statements	Explanation
' c '	filler character	INPUT 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.

If the filler character is set to blank (X'40'), filling blanks are replaced by X'00' to allow for insertion of characters without having to clear the remainder of the input field before.

In BS2000/OSD environments, X'00' characters are displayed as dots on 97xx type terminals. Their appearance can be changed by means of the SIDA utility or with the configuration utility of the respective terminal emulation.

6 ADAMODE - Adabas Call Interface Mode

This Natural profile parameter controls the Adabas call interface mode and the number of Adabas user sessions used by Natural to issue Adabas calls.

Possible settings	See below.
Default setting	2
Dynamic specification	yes
Specification within session	no

The following values are possible for the ADAMODE parameter:

Value	Separate Adabas User Sessions for Nucleus and User Application Database Calls [1]	3GL Program Adabas Calls Use Natural's Adabas Session for User Application Calls [2]	Image Switching in a z/OS Parallel Sysplex Environment Supported [3]
0	No	Yes	No
1	No	No	Yes
2	Yes	No	Yes
3	Yes	Yes	No

Notes:

1. Separate Adabas User Sessions for Nucleus and User Application Database Calls

Two Separate Adabas User Sessions

If Natural uses two separate Adabas user sessions to issue Adabas calls, Natural uses one Adabas user session to handle Adabas calls issued by the Natural nucleus (for example, to load Natural objects from the system file), and the other Adabas user session to issue Adabas calls issued by the user application.

An Adabas timeout (leading to Natural error NAT3009) that occurs for the Adabas user session that is used to handle Adabas calls issued by the Natural nucleus does not affect the user application.

A separate Adabas user queue element (UQE) is generated for each Adabas user session, and it may be necessary to increase the Adabas ADARUN parameter NU.

Single Adabas User Session

If Natural uses only a single Adabas user session, `END TRANSACTION` and `BACKOUT TRANSACTION` statements issued by either the Natural nucleus or the user application affect transactions started by both the Natural nucleus and the user application.

An Adabas timeout (leading to Natural error NAT3009) that occurs for the Adabas user session is always reported to the user application, because it is not possible to check whether the timeout affects the application's transaction state.

If Natural uses a single Adabas user session to handle Adabas calls issued by the Natural nucleus as well as Adabas calls issued by the user application, only one UQE is necessary.

2. 3GL Program Adabas Calls Use Natural's Adabas Session for User Application Calls

Calls Using Natural's Adabas Session

If a 3GL program, which is called from the user application, issues Adabas calls, and if these Adabas calls use Natural's Adabas session for user application calls, these Adabas calls participate in the user application's transaction handling (`END TRANSACTION` and `BACKOUT TRANSACTION` statements), and they are affected by Natural transaction processing related profile parameter settings (see the parameters mentioned below).

Calls Not Using Natural's Adabas Session

If a 3GL program, which is called from the user application, issues Adabas calls, and if these Adabas calls do not use Natural's Adabas session for user application calls, these Adabas calls will not participate in Natural's transaction handling for the Adabas user session. As a consequence, such 3GL programs need to perform their own transaction handling.

3. Image Switching in a z/OS Parallel Sysplex Environment Supported

If image switching in a z/OS Parallel Sysplex environment is supported, the Natural session may, after a terminal I/O operation, seamlessly continue to execute in a z/OS image that is different to the z/OS image where the Natural session has executed before the terminal I/O operation. Installation of the Natural Roll Server is required to support execution in a z/OS Parallel Sysplex environment.

To ascertain support of image switching in a z/OS Parallel Sysplex environment, even if `ADAMODE=0` is set, Adabas System Coordinator (product code COR) must be installed.



Caution: Setting the value of `ADAMODE` so that image switching in a z/OS Parallel Sysplex environment is not supported may lead to unpredictable results if the Natural session continues execution in a another z/OS image after a terminal I/O operation. Depending on Natural transaction processing related profile parameter settings (see the parameters mentioned below), this may include:

- non-zero Adabas response codes (leading to, for example, Natural error NAT3021),
- database updates that have not yet been committed by an `END TRANSACTION` statement are unintentionally backed out or applied to the database.

Other transaction processing related parameters: `DBCLOSE` | `DBOPEN` | `ENDBT` | `ET` | `ETDB` | `ETEOP` | `ETIO` | `ETSYNC`

7

ADANAME - Name of Adabas Link Routine

This Natural profile parameter specifies the name of the Adabas link routine to be used.



Note: It does not apply to UTM and Com-plete.

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

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

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



Note: Under CICS, the Adabas link routine must not be linked to Natural.

8 ADAPRM - REVIEW/DB Support

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

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

9 ADASBV - Adabas Security by Setting

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

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

10

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*.

11 ASIZE - Entire System Server Auxiliary Buffer

This Natural profile parameter determines the size of the Entire System Server auxiliary buffer.



Caution: It only applies if Entire System Server is installed.

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

Possible settings	1 - 64	Buffer size in KB. If Entire System Server is to be used, this parameter <i>must</i> be set; see the Entire System Server documentation.
	0	If ASIZE=0 is specified or if the requested space is not available, the Entire System Server is not activated.
Default setting	0	
Dynamic specification	yes	
Specification within session	no	

12

ASPSIZE (Internal Use)

This parameter is reserved for internal use by Natural.



Caution: Do not change its setting.

13

ASYNNAM - Output System ID for Asynchronous Processing

This Natural profile parameter applies to Natural under UTM.

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

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

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

14

ATTN - Attention Key Interrupt Support

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

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

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

15

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 <i>SYSEXT - Natural Application Programming Interfaces</i> in the <i>Utilities</i> documentation.

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



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

16

AUTORPC - Automatic Natural RPC Execution

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

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

AUTORPC is specified on the client side only.

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

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

17

BPC64 - Buffer Pool Cache Storage Type

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

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

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

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

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

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

Example:

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

For general information on the Natural Buffer Pool, see *Natural Buffer Pool* in the *Operations* documentation.

18

BPCSIZE - Cache Size for Natural Buffer Pool

This Natural profile parameter is applicable under z/OS and z/VSE only (not for Complete and not for IMS TM).

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

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

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

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

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

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

Example:

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

For more information see *Buffer Pool Cache* in the *Operations* documentation.

19

BPI - Buffer Pool Initialization

■ BPI Parameter Syntax	48
■ NTBPI Macro Syntax	49
■ Keyword Subparameters	49
■ Examples of NTBPI Macros	54
■ Examples of BPI Parameter	55

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

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

Possible settings	See Keyword Subparameters , below.	Possible subparameter keywords: TYPE SEQ NAME SIZE CSIZE LIST TXTSIZE METHOD C64 Under BS2000/OSD, SIZE and CSIZE are ignored.
Default setting	TYPE=NAT,SEQ=0,NAME=' ',SIZE=256,CSIZE=0,TXTSIZE=4,METHOD=S,C64=OFF	
Dynamic specification	yes	The parameter BPI can only be specified dynamically. In NATPARM, use the macro NTBPI .
Specification within session	no	



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

The following topics are covered below:

BPI Parameter Syntax

The BPI parameter is specified as follows:

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

- To dynamically deactivate a buffer-pool definition, use the special value **OFF** as follows:

```
BPI=(TYPE=type,SEQ=n,OFF)
```

- If **OFF** is used, it must be specified after **TYPE** and **SEQ**. **OFF** is not allowed for the macro **NTBPI**.

- If you use the `BPI` parameter to overwrite an existing buffer pool definition in the parameter module, you must specify new settings in all those subparameters which are to be changed; if you do not, the old settings will still be used.

If, for example, you want to change from a global to a local buffer pool, you must specify: `NAME=' '`.

- If you use the `BPI` parameter to dynamically add a new backup buffer pool definition, you must specify a sequence number ([SEQ](#)) for it.

If you omit the `SEQ` specification, the definition of the primary buffer pool (`SEQ=0`) will be overwritten.

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

NTBPI Macro Syntax

The NTBPI macro is specified as follows:

```

.....1.....2.....3.....4.....5.....6.....7..
      NTBPI  TYPE=type,                      *
              SEQ=n,                        *
              NAME=name,                    *
              CSIZE=nnn,                    *
              SIZE=nnn,                     *
              LIST=name,                    *
              METHOD=x,                      *
              C64=xx,                       *
              TXTSIZE=n

```

Keyword Subparameters

[TYPE](#) | [SEQ](#) | [NAME](#) | [SIZE](#) | [CSIZE](#) | [LIST](#) | [TXTSIZE](#) | [METHOD](#) | [C64](#)

TYPE - Type of Buffer Pool

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

NAT	Natural buffer pool (this is the default). For general information on the Natural buffer pool, see <i>Natural Buffer Pool</i> in the <i>Operations</i> documentation. *
DLI	DL/I buffer pool. *
EDIT	Software AG Editor buffer pool. Alternatively, an editor auxiliary buffer pool can be defined per session, see also the profile parameter EDPSIZE .
ICU	Buffer pool for Unicode and code page support. For further information, see <i>ICU Buffer Pool</i> in the <i>Unicode and Code Page Support</i> documentation. See also profile parameter CFICU .
SORT	Sort buffer pool. *
MON	Buffer pool for monitoring function (SYSMON) of SYSTP utility.
SWAP	Buffer pool to hold the Natural CICS swap pool.



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

SEQ - Sequence Number of Buffer Pool

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

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

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

Possible values	0 to 9
Default value	0

NAME - Name of Global Buffer Pool

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



Note: Under BS2000/OSD, an **ADDON** macro with the same value for the keyword subparameter **NAME** is required in the **BS2STUB** used.

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

Possible values	1 to 8 characters
Default value	' '

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

Under Com-plete: Because an SD file is used under Com-plete as Editor work file, a global Editor buffer pool is not possible, but only a local Editor buffer pool.

Under IMS TM: Because a Natural session may be executed in different regions, a local Editor buffer pool is not possible, but only a global Editor buffer pool.

SIZE - Size of Buffer Pool

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



Note: Under BS2000/OSD, **SIZE** is ignored.

Possible values	256 to 2097151 (KB) for Natural buffer pools
	100 to 2097151 (KB) for other buffer pool types
Default value	256

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

CSIZE - Size of the Local Buffer Pool Cache

Applies to local buffer pools of **TYPE=NAT** only (not for Com-plete) and to pools of **TYPE=SWAP** under CICS.

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



Note: Under BS2000/OSD, **CSIZE** is ignored.

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

For more information see Buffer Pool Cache.

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

LIST - Name of Preload List to be Used

Applies only to local buffer pools of **TYPE=NAT**.

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

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

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

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

TXTSIZE - Size of Buffer Pool Text Segments

Applies to local buffer pools of the following **types**:

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

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

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

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

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

METHOD - Search Algorithm for Allocating Space in Buffer Pool

Applies to local buffer pools of **TYPE=NAT** only.

Determines the algorithm for allocating storage in the buffer pool.

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

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

C64 - Type of Buffer Pool Cache Storage

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

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

Possible values	ON	This indicates that a memory object “above the bar” (that is, in 64-bit memory) is to be used for the buffer pool cache. Note that C64=ON is only honored if the prerequisites are met, namely: <ul style="list-style-type: none">■ z/ architecture hardware,■ operating system z/OS Version 1.2 or higher. If the prerequisites are not met, the default value is taken.
	OFF	This indicates that a data space is to be used for the buffer pool cache.
Default value	OFF	

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

Examples of NTBPI Macros

```
.....1.....+.....2.....+.....3.....+.....4.....+.....5.....+.....6.....+.....7..
NTBPI TYPE=NAT,                                     *
      SEQ=0,                                         *
      NAME=NATBP1
NTBPI TYPE=NAT,                                     *
      SEQ=1,                                         *
      NAME=NATBP2
NTBPI TYPE=NAT,                                     *
      SEQ=2,                                         *
      SIZE=1000,                                     *
      METHOD=N
```

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

Examples of BPI Parameter

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

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

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

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

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

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

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

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

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

20

BPLIST - Name of Preload List for Natural Buffer Pool

This Natural profile parameter specifies the name of a preload list to be used for the Natural buffer pool.

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

For general information, see *Natural Buffer Pool* in the *Operations* documentation.

The parameter corresponds to the LIST specification of the BPI profile parameter or the [NTBPI](#) macro.

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

Example:

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

21

BPMETH - Buffer Pool Space Search Algorithm

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

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

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

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

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

Example:

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

For general information on the Natural buffer pool, see *Natural Buffer Pool* in the *Operations* documentation.

22

BPNAME - Name of Natural Global Buffer Pool

This Natural profile parameter specifies the name of the Natural global buffer pool.

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

For general information, see *Natural Global Buffer Pool* in the *Operations* documentation.

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

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

Example:

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

23

BPPROP - Global Buffer Pool Propagation

This Natural profile parameter only applies under z/OS and BS2000/OSD.

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

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

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

For further information on the propagation, see *Natural Global Buffer Pool* in the *Operations* documentation.

24

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	<p>Search Sequence 1 is used (search buffer pool first for all libraries, then the system file(s)).</p> <p>Natural looks for the object in the following sequence until it is found:</p> <ol style="list-style-type: none"> 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. <p>For performance reasons, it is recommended that you set BPSFI=ON in production environments.</p> <p>Caution: If you set BPSFI=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.</p>
	OFF	<p>Search Sequence 2 is used (alternating search in buffer pool and system file(s) for each library).</p> <p>Natural looks for the object in the following sequence until it is found:</p> <ol style="list-style-type: none"> 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 *Steplib Libraries* and *Search Sequence for Object Execution* in the *Using Natural* documentation.

25

BPSIZE - Size of Natural Local Buffer Pool

This Natural profile parameter specifies the size of the Natural local buffer pool.

It corresponds to the [SIZE](#) specification of the [BPI](#) profile parameter or the [NTBPI](#) macro.



Notes:

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

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

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

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

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

Example:

```
BPSIZE=1500
```

is converted into:

```
BPI=(TYPE=NAT,SEQ=0,SIZE=1500)
```

For general information, see *Natural Buffer Pool* in the *Operations* documentation.

26

BPTEXT - Size of Text Segments in Natural Buffer Pool

This Natural profile parameter specifies the size of the segments into which the text pool area of the Natural buffer pool is divided.

It corresponds to the **TXTSIZE** specification of the **BPI** profile parameter or the **NTBPI** macro.

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

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

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

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

Example:

```
BPTEXT=4
```

is converted into:

```
BPI=(TYPE=NAT,SEQ=0,TXTSIZE=4)
```

For general information on the Natural Buffer pool, see *Natural Buffer Pool* in the *Operations* documentation.

27

BSIZE - Size of EntireX Broker Buffer

This Natural profile parameter only applies if EntireX Broker is installed.

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

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

Possible settings	1 - 64	Buffer size in KB.
Default setting	0	
Dynamic specification	yes	
Specification within session	no	

28

BX - Box Definition

With this session parameter, you specify which parts of a box are to be displayed for field outlining.

Outlining (boxing) is the capability to generate a line around certain fields when they are displayed on the terminal screen. Drawing such “boxes” around fields is another method of showing the user the lengths of fields and their positions on the screen.

The outlining feature is only available on certain types of terminals, usually those which also support the display of double-byte character sets. If the terminal used does not support outlining, this parameter will be ignored at execution time.

Possible settings	T	Top horizontal line. See Note 1.
	B	Bottom horizontal line. See Note 1.
	L	Lefthand vertical line. See Notes 1 and 2.
	R	Righthand vertical line. See Notes 1 and 2.
	ON	Corresponds to BX=TBLR.
	OFF	Causes no boxes to be drawn around the fields concerned.
Default setting	none	
Applicable statements:	FORMAT	
	DISPLAY INPUT WRITE	Parameter may be specified at statement level and/or at element level.
Applicable command:	none	



Notes:

1. You can specify the values T, B, L, R in any order.

2. If you use the session parameter settings `BX=L` or `BX=R`, you should switch off Natural's screen optimization using the profile parameter setting `DSC=OFF` or the Natural terminal command `%R0`.

Example:

```
DISPLAY #FIELD1 (BX=RLT) /  
        #FIELD2 (BX=TLRB)
```

See also the terminal command `%D=`.

29

CANCEL - Session Cancellation with Dump

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

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

30

CC - Error Processing in Batch Mode

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

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

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

Possible settings	ON	Natural flushes the input data stream for the batch input files CMSYNIN and CMOBJIN until a line containing %% in the first two positions is encountered or until an end-of-file condition is detected. If more data are available in the input stream, Natural resumes reading after the line containing %%.	
	OFF	Natural attempts to process the next program (or command) in the input stream.	
Default setting	OFF		
Dynamic specification			
Specification within session	yes	Applicable Statements:	SET GLOBALS
		Applicable Command:	GLOBALS
Application Programming Interface	USR1005N	See SYSEXT - <i>Natural Application Programming Interfaces in the Utilities</i> documentation.	

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

31

CCTAB - Printer Escape Sequence Definition

■ CCTAB Parameter Syntax	80
■ NTCCTAB Macro Syntax	82
■ String Syntax for OPN, CLS, CODE, CS, CSS or CSE	82
■ Proportional Fonts	83
■ Examples of NTCCTAB Macros	83
■ Examples of CCTAB Parameter	83

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

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

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

The following topics are covered below:

CCTAB Parameter Syntax

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

1st Variant

CCTAB=([name](#) , [OPN](#)='xxxxx' , [CLS](#)='yyyyy')

Where

name is the name of the profile form, that is, the `DEFINE PRINTER (n) OUTPUT 'nnnnn' PROFILE 'name'`, which is required and which has a maximum length of 8 bytes.

`OPN='xxxxx'` is optional and defines a data string (up to 250 bytes) which is sent to the printer with each open operation.

`CLS='yyyyy'` is optional and defines a data string (up to 250 bytes) which is sent to the printer before each close operation.

`OPN` and `CLS` can be specified in any sequence.

2nd Variant

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

Where

`CODE='n'` is a character which is recognized by Natural once it appears in the output string.

`CS='xxxx'` is the string to replace the `CODE` character.

The `CS` subparameter must follow the `CODE` subparameter.

3rd Variant

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

Where

`ATR='nnnn'` is the Natural internal field attribute. The name is defined by the macro `NAMATR`.

`CSS='xxxx'` is the string (up to 20 bytes) which is inserted before the field. `CSS` is mandatory.

`CSE='yyyy'` is the string (up to 20 bytes) which is inserted after the field. `CSE` is mandatory.

The `CSS` and `CSE` subparameters must follow the `ATR` subparameter.

NTCCTAB Macro Syntax

The NTCCTAB macro can be specified in three variants:

1st Variant

```
....+....1....+....2....+....3....+....4....+....5....+....6....+....7..  
    NTCCTAB name, *  
        OPN='xxxxx', *  
        CLS='yyyyy'
```

For details, refer to the CCTAB parameter syntax, [1st Variant](#).

2nd Variant

```
....+....1....+....2....+....3....+....4....+....5....+....6....+....7..  
    NTCCTAB name, *  
        CODE='n', *  
        CS='xxxx'
```

For details, refer to the CCTAB parameter syntax, [2nd Variant](#).

3rd Variant

```
....+....1....+....2....+....3....+....4....+....5....+....6....+....7..  
    NTCCTAB name, *  
        ATR=nnnn, *  
        CSS='xxxx', *  
        CSE='yyyy'
```

For details, refer to the CCTAB parameter syntax, [3rd Variant](#).

String Syntax for OPN, CLS, CODE, CS, CSS or CSE

You specify character strings either as characters (enclosed in apostrophes) or as the corresponding hexadecimal representation of the characters (without apostrophes).

Proportional Fonts

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

Examples of NTCCTAB Macros

```
.....1.....2.....3.....4.....5.....6.....7..
NTCCTAB DBCST
NTCCTAB CODE=0E,CS=400E
NTCCTAB CODE=0F,CS=0F40<
NTCCTAB ATR=P5DBCS,CSS=0E,CSE=0F
```

```
.....1.....2.....3.....4.....5.....6.....7..
NTCCTAB TEST,OPN=27C5274DA2F1F188275093F0D6,CLS='LAST LINE'
NTCCTAB CODE='<',CS=' B(SOB'
NTCCTAB CODE='>',CS='B(S3B '
NTCCTAB CODE='(',CS=' B(S1S'
NTCCTAB CODE=')',CS='B(SOS '
NTCCTAB ATR=P2UL,CSS=' B&&DD',CSE='B&&D$'
NTCCTAB ATR=P2UL,CSS=405FF1C25084C4,CSE=5FF1C250847C
NTCCTAB ATR=P2ITAL,CSS=' B(S1S',CSE='B(SOS'
NTCCTAB ATR=P1HIGH,CSS=' B(S3B',CSE='B(SOB'
NTCCTAB ATR=P2RVID,CSS=' B(S-3B',CSE='B(SOB'
```

Examples of CCTAB Parameter

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

```
CCTAB=(OPN=27C5274DA2F1F188275093F0D6,CLS='LAST LINE')
```


32

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
```

33

CDYNAM - Dynamic Loading of Non-Natural Programs

This Natural profile parameter defines whether or not non-Natural programs can be loaded dynamically by Natural during the execution of a single Natural program.

Possible settings	1 - 1024	<p>If CDYNAM is set to 0, no dynamic loading of non-Natural programs will be performed.</p> <p>If CDYNAM is set to a value greater than 0, dynamic loading of non-Natural programs will be performed by Natural.</p>
	0	
Default setting	> 0	
Dynamic specification	yes	
Specification within session	no	

34

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.

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

Possible settings	any special character	<p>A terminal command must begin with the character specified. The character specified with the CF parameter</p> <ul style="list-style-type: none"> ■ 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	USR0350N, USR1005N *	<p>See SYSEXT - <i>Natural Application Programming Interfaces in the Utilities</i> documentation.</p> <p>* Recommended.</p>	

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

35

CFICU - Unicode and Code Page Support

■ CFICU Parameter Syntax	92
■ NTCFICU Macro Syntax	93
■ Keyword Subparameters	93
■ Example of NTCFICU Macro	96
■ Example of Dynamic Parameter CFICU	96

This Natural profile parameter is required to enable Unicode and code page support for various Unicode settings, for example, if U format variables or the statement `MOVE ENCODED` are to be used. It corresponds to the `NTCFICU` macro in the parameter module `NATPARM`.

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

Possible settings	See syntax, below.	Possible subparameter keywords: See Keyword Subparameters , below.
Default setting	ON or OFF	Enables or disables the Unicode and code page support. The default value is OFF if profile parameter CP is set to OFF. Otherwise, the default is ON.
Dynamic specification	yes	The parameter CFICU can only be specified dynamically. In NATPARM, use the macro <code>NTCFICU</code> .
Specification within session	no	

CFICU=ON is enforced if profile parameter CP is set to any value other than OFF.

The following topics are covered below:

CFICU Parameter Syntax

The CFICU profile parameter is specified as follows:

CFICU=(ON,[COLLATE=value](#),[COLNORM=value](#),[CNVNORM=value](#),[LOCALE=value](#),[CPOPT=value](#),[DATFILE=value](#),[I](#)

Or:

CFICU=(OFF,[COLLATE=value](#),[COLNORM=value](#),[CNVNORM=value](#),[LOCALE=value](#),[CPOPT=value](#),[DATFILE=value](#),[I](#)

Or:

CFICU=[{ [\(ON\)](#)
 [\(OFF\)](#) }]

NTCFICU Macro Syntax

The NTCFICU macro is specified as follows:

```

.....1.....2.....3.....4.....5.....6.....7..
      NTCFICU ON/OFF,                                *
          COLLATE=value,                              *
          COLNORM=value,                              *
          CNVNORM=value,                              *
          LOCALE=value,                               *
          CPOPT=value,                                *
          DATFILE=value,                              *
          BPOONLY=value                               *

```

Or:

```
NTCFICU ON
```

Or:

```
NTCFICU OFF
```

Keyword Subparameters

COLLATE | COLNORM | CNVNORM | LOCALE | CPOPT | DATFILE | BPOONLY

COLLATE - Collation Services

Collation is the process of ordering units of textual information (alphabetic sorting). Collation is usually specific to a particular language.

Examples: The character “Ä” is sorted in german locale between “A” and “B”, but in Swedish locale it is sorted after “Z”. In Lithuanian, “y” is sorted between “i” and “k”.

This subparameter determines the collation service used.

ON	Use Locale ID and ICU's collation services to compare Unicode strings. This is the default value.
OFF	Use ICU's simple Unicode compare.

COLNORM - Normalization Check of Collation Services

Normalization is the process of removing alternate representations of equivalent sequences from textual data, to convert data into a form that can be binary-compared for equivalence.

This subparameter is honored only if `COLLATE=ON` is set.

The ICU Collation Service handles un-normalized text properly, producing the same results as if the text were normalized. This maximizes performance for the majority of text that does require normalization. If Unicode data is known with certainty not to contain un-normalized text, then even the overhead checking for normalization can be eliminated.

ON	Check for un-normalized text.
OFF	Disable check for un-normalized text. This is the default value.

CNVNORM

The German character “ä”, for example, can be represented in Unicode as U+00E4 or with the use of a combining character as U+0061, U+0308. Conversion to a codepage considers the combined “ä” (U+0061 U+0308) as two code points and produces an “a” and a substitution character, if U+0308 is no valid character of the target code page. Normalization before conversion creates one code point U+00E4 from the combined code points U+0061 U+0308 and the subsequent conversion will deliver the result “ä”. The parameter is honored whenever a conversion from U to A format is performed, for example `MOVE U TO A` or `DISPLAY U`, when the output device is a terminal emulation. The additional operation consumes of course additional storage as well as additional CPU time.

If `CNVNORM=OFF`, the `MOVE NORMALIZED` statement can be used to normalize selected strings.

This subparameter activates/deactivates normalization before conversion..

ON	Enable normalization before conversion.
OFF	Disable normalization before conversion. This is the default value.

LOCALE - Locale ID

The Locale ID is used by ICU's Collation Service to consider language and even region-dependent features of collation. The language code of the Locale ID follows ISO639, and the region code follows ISO 3166.

Possible values	<i>LLL_RRR</i>	<i>LLL</i> is a 2- or 3-byte language code of lower-case characters. If specified in upper case, it will be translated into lower case automatically. <i>RRR</i> is a 2- or 3-byte region code of upper-case characters to classify the language.
Default value	en_US	

Examples:

en_US	English (United States)
en_UK	English (United Kingdom)
de_DE	German (Germany)
de_AT	German (Austria)
de_CH	German (Switzerland)
sv_SE	Scandinavian (Sweden)

CPOPT - Fast Code Page Conversion

By default, a conversion from alpha to Unicode format and vice versa is performed by calling ICU functions. Certain code pages are mapping characters to Unicode with 1:1 relationship. In this case, the conversion performance can be increased by using internal translation tables rather than ICU functions.

ON	Use internal translation tables instead of ICU functions, if possible.
OFF	Use ICU functions in any case. This is the default value.

DATFILE - Additional Data Files

Optional data file name. It must be loadable by using RCA technique. The data file contains the converter mapping tables, collation rules, break iterator rules and other locale data. The ICU development kit provides tools to build data files that comply with particular requirements. Refer to the chapter *Data Management* in the *ICU User Guide* for more information.

Possible values	<i>name</i>	The data file name can be up to 8 characters long.
	OFF	Removes any data files defined, that is, the default data file is used.
Default value	None	No additional data files are defined. The default data file is used.

BPONLY – Use of ICU Buffer Pool

In thread environments (for example, under CICS) an ICU buffer pool can be used to share ICU data and to reduce the overhead caused by thread compression and roll-out. The keyword sub-parameter `BPONLY` determines whether the ICU must use the ICU buffer pool or if it can use thread storage if the ICU buffer pool is not available.

For further information, see *ICU Buffer Pool* in the *Unicode and Code Page Support* documentation.

ON	An ICU buffer pool must be used. Initialization messages NAT3419 and NAT3410 are displayed if the ICU buffer pool cannot be used, and the ICU initialization will be terminated. An error is displayed if ICU is to be used during the session, but ICU is disabled.
OFF	If an ICU buffer pool is not available, ICU will use thread storage. Initialization message NAT3419 will be displayed if ICU uses thread storage instead of an ICU buffer pool. This is the default value.

Example of NTCFICU Macro

```
NTCFICU COLNORM=ON,LOCALE=de_DE,DATFILE=TEST15
```

Example of Dynamic Parameter CFICU

```
CFICU=(COLNORM=ON,LOCALE='de_DE',DATFILE=TEST15)
```

36 CFWSIZE (Internal Use)

This parameter is reserved for internal use by Natural.



Caution: Do not change its setting.

37

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	<p>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.</p> <p>Example:</p> <pre>CF=% CLEAR=R</pre> <p>Natural executes the terminal command %R when the CLEAR key is pressed in NEXT mode.</p>
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	

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

38 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	yes	
Specification within session	no	



Note: In asynchronous Natural sessions under the TP monitors Complete, CICS and UTM, the setting of this parameter is forced to OFF to avoid problems when changing over to NEXT mode.

39

CMPO - Compilation Options

■ CMPO Parameter Syntax	104
■ NTCMPO Macro Syntax	104
■ Keyword Subparameters	104
■ Example of CMPO Parameter	105
■ Example of NTCMPO Macro	105

This Natural profile parameter can be used at session start to specify dynamically or to override the same options which you can specify statically with the [NTCMPO](#) macro in the parameter module or, during an active session, with the `COMPOPT` system command.

Possible settings	See system command	
Default setting	<code>COMPOPT</code> .	
Dynamic specification	yes	This parameter can only be specified dynamically. In the Natural parameter module NATPARM, the macro NTCMPO must be used instead.
Specification within session	yes	See system command <code>COMPOPT</code> .

The following topics are covered below:

CMPO Parameter Syntax

The parameter syntax of `CMPO` is as follows:

```
CMPO=(keyword-subparameter1=value,keyword-subparameter2=value,...)
```

NTCMPO Macro Syntax

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

```
NTCMPO keyword-subparameter1=value,keyword-subparameter2=value,...
```

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

Keyword Subparameters

The following keyword subparameters are available:

`CPAGE` | `DB2ARRY` | `DBSHORT` | `GFID` | `FINDMUN` | `KCHECK` | `LOWSRCE` | `MASKCME` | `NMOVE22` | `PCHECK` | `PSIGNF` | `THSEP` | `TQMARK` | `TSENL` | `V41COMP`

In the macro, the keyword subparameters can be specified in any sequence. For details, follow the links shown above. For a complete description, refer to the system command `COMPOPT` in the *System Commands* documentation.

Example of CMPO Parameter

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

Example of NTCMPO Macro

```
.....1.....2.....3.....4.....5.....6.....7..  
      NTCMPO KCHECK=ON,  
            PCHECK=ON
```


40

CMPR - General Default Compression Optimization

Algorithm

This Natural profile parameter enables the Natural administrator to define the general default compression optimization in order to preserve main storage for the sessions which are currently processing and to improve the performance of Natural.

In addition, the type of storage compression optimization can be defined specifically for individual buffer types, using the parameter `CMPR` of the `NTBUFID` macro in the `NATCONFIG` module. The setting of this macro parameter overrides the general default setting of profile parameter `CMPR`. For further information, see *Customization of Buffer Characteristics* in the *Operations* documentation.

Possible settings	OPT0	Compression without optimization
	OPT1	Compression with optimization of identical characters from the buffer used low end and high end.
	OPT2	Compression with optimization by tiles with identical characters. The tile size is 128 bytes.
	(OPT2, <i>nnn</i>)	Compression with optimization by tiles with identical characters. The tile size is a multiple of 128 bytes. <i>nnn</i> determines the tile size by multiplying with 128. Possible values: 1 - 255. Example: (OPT2, 5) yields a tile size of 640 bytes.
Default setting	OPT2	This is a synonym for (OPT2, 1).
Dynamic specification	yes	
Specification within session	no	

41

COMPR - Set RPC Buffer Compression

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

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

COMPR is specified on the client side only.

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

For further information, see the *Natural Remote Procedure Call (RPC)* documentation.

42

CP - Default Code Page Name

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

If no code page is specified for a code page sensitive operation such as data conversions to and from Unicode (for example, by means of a statement specific `ENCODED` option or by another profile parameter), the default code page applies.

For the current Natural session, it is assumed that all code page data, for example, Natural sources, contents of A-format fields, etc., are stored in this code page format.

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

Possible settings	code-page	The name of the desired code page. Length: 1 - 64 characters.	
		Any character string is possible, but must be predefined by one of the code page parameters CCSID, CCSN, IANA or ALIAS of the macro NTCPAGE in the source module NATCONFIG. UTF-32 is not allowed.	
		For information on multi-byte code page support, see <i>Multi-Byte Code Pages</i> in the <i>Unicode and Code Page Support</i> documentation.	
	ON	Set the default code page for IBM and Siemens mainframes as follows:	
		In case of Siemens mainframes, the code page is EDF03IRV.	
		In case of IBM mainframes, it depends on the setting of Natural profile parameter ULANG :	
		ULANG Setting:	Code Page Used:
		ULANG=1 (English)	IBM01140
		ULANG=2 (German)	IBM01141
		ULANG=3 (French)	IBM01147
ULANG=4 (Spanish)		IBM01145	
For other languages, IBM01140 is used as default code page.			

		Note: The language code related adaptation of the profile parameter CP applies only to the ULANG profile parameter active at session time. Any subsequent language code modification(s) in Natural Security or by terminal command %L= do not influence the initial definition of the default code page.
	OFF	Disable code page support.
	' ' (blank)	Same as ON.
	AUTO	The code page name from the user terminal is taken, if available. This applies to the following online environments only: TSO, CICS, Com-plete and VM/CMS. For information on multi-byte code page support, see <i>Multi-Byte Code Pages</i> in the <i>Unicode and Code Page Support</i> documentation. Note: CP=AUTO is not supported in a Natural Single Point of Development environment.
Default setting	OFF	Disable code page support.
Dynamic specification	yes	
Specification within session	no	

If the CP profile parameter is set to a value other than OFF, the value of the [CFICU](#) profile parameter will change to ON.

If the profile parameter CP is set to a multi-byte code page (MBCS), the logical shift-in and shift-out characters will be supplied with the code page and therefore [SOSI](#) will be ignored.

Tips:

- You can find out the default code page that is the result of the evaluation of the CP parameter by viewing the content of the system variable *CODEPAGE or by using the *Unicode Properties* function of the SYSCP utility.
- You can use the LIST DIRECTORY system command or the SYSCP utility to find out the default code page used for encoding a Natural source object. The SYSCP utility can also be used to change the code page for a source object.

43

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. This parameter is not regarded for the conversion of Natural sources when loading them into the source area or during catalog.

On mainframe platforms, 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 place holder character which is defined in NATCONFIG.

See also:

- *Profile Parameters in the Unicode and Code Page Support documentation.*
- *Code Page Support for Editors, System Commands and Utilities on the Mainframe in the Unicode and Code Page Support documentation.*

Possible settings	ON	A Natural error NAT3413 is issued, if at least one code point could not be translated correctly during ICU conversion. For output statements, no error message is issued.
	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

44

CPOBJIN - Code Page of Batch Input File

This Natural profile parameter specifies the code page of the batch input file CPOBJIN (see *Natural in Batch Mode*).

If Natural code page support is disabled (for example, by parameter CP=OFF), any value specified for this parameter is ignored.

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

Possible settings	1 -64 characters	The name of the desired code page. Any character string is possible, but must be predefined by one of the code page parameters CCSID, CCSN, IANA or ALIAS of the macro NTCPAGE in the source module NATCONFIG. UTF-32 is not allowed.
	' ' (blank)	The code page resulting from the evaluation of the profile parameter CP is used.
Default setting	' ' (blank)	The code page resulting from the evaluation of the profile parameter CP is used.
Dynamic specification	yes	
Specification within session	no	

45

CPPRINT - Code Page of Batch Output File

This Natural profile parameter specifies the code page of the batch output file `CMPRINT` (see *Natural in Batch Mode*).

If Natural code page support is disabled (for example, by parameter `CP=OFF`), any value specified for this parameter is ignored.

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

Possible settings	1 - 64 characters	The name of the desired code page. Any character string is possible, but must be predefined by one of the code page parameters <code>CCSID</code> , <code>CCSN</code> , <code>IANA</code> or <code>ALIAS</code> of the macro <code>NTCPAGE</code> in the source module <code>NATCONFIG</code> .
	' ' (blank)	The code page resulting from the evaluation of the profile parameter <code>CP</code> is used.
Default setting	' ' (blank)	The code page resulting from the evaluation of the profile parameter <code>CP</code> is used.
Dynamic specification	yes	
Specification within session	no	

46 CPRPC - Define Code Page Name

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

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

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

See also *Unicode and Code Page Support, Configuration and Administration of the Unicode/Code Page Environment, Profile Parameters*.

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

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

47

CPSYNIN - Code Page of Batch Input File for Commands

This Natural profile parameter specifies the code page of the batch input file for commands `CMSYNIN` (see *Natural in Batch Mode*).

If Natural code page support is disabled (for example, by parameter `CP=OFF`), any value specified for this parameter is ignored.

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

Possible settings	1 - 64 characters	The name of the desired code page. Any character string is possible, but must be predefined by one of the code page parameters <code>CCSID</code> , <code>CCSN</code> , <code>IANA</code> or <code>ALIAS</code> of the macro <code>NTCPAGE</code> in the source module <code>NATCONFIG</code> . UTF-32 is not allowed.
	' ' (blank)	The code page resulting from the evaluation of the profile parameter <code>CP</code> is used.
Default setting	' ' (blank)	The code page resulting from the evaluation of the profile parameter <code>CP</code> is used.
Dynamic specification	yes	
Specification within session	no	

48

CSIZE - Size of Con-nect/Con-form Buffer Area

This Natural profile parameter determines the size of the Con-nect/Con-form buffer area. It only applies if Con-nect/Con-form is installed.

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

Possible settings	1 - 512	Buffer size in KB.
	0	If <code>CSIZE=0</code> is specified or if the requested space is not available, Con-nect/Con-form cannot be used.
Default setting	0	
Dynamic specification	yes	
Specification within session	no	

See the Con-nect/Con-form *Installation* documentation for further information.

49

CSTATIC - Programs Statically Linked to Natural

■ Example of CSTATIC Parameter	126
■ Example of NTCSTAT Macro	127

This Natural profile parameter can be used to define a list of names of non-Natural programs which are to be linked together with the Natural parameter module (NATPARM).

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

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

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

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

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

For further information, see *Statically Linked Non-Natural Programs* in the *Operations* documentation.

The following topics are covered below:

Example of CSTATIC Parameter

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

Example of NTCSTAT Macro

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


50

CV - Attribute Control Variable

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

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

Example:

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

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 ...
```

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.

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

51

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. If an attribute control variable has been assigned the status `MODIFIED`, the `MODIFIED` option evaluates this as `TRUE`. This applies regardless of whether the input was entered manually, read from the stack or supplied in batch mode.

Possible settings	ON	If a field setting is overwritten by the same setting, the corresponding control variable will be assigned the status <code>MODIFIED</code> .
	OFF	If a field setting is overwritten by the same setting, the corresponding control variable will not be assigned the status <code>MODIFIED</code> .
Default setting	OFF	
Dynamic specification	yes	
Specification within session	no	

52

DATSIZE - Minimum Size of Buffer for Local Data

This Natural profile parameter can be used to set the minimum size of the local data buffer (DATSIZE).

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

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

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

Function of the DATSIZE Buffer

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

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

Calculating the DATSIZE Requirement

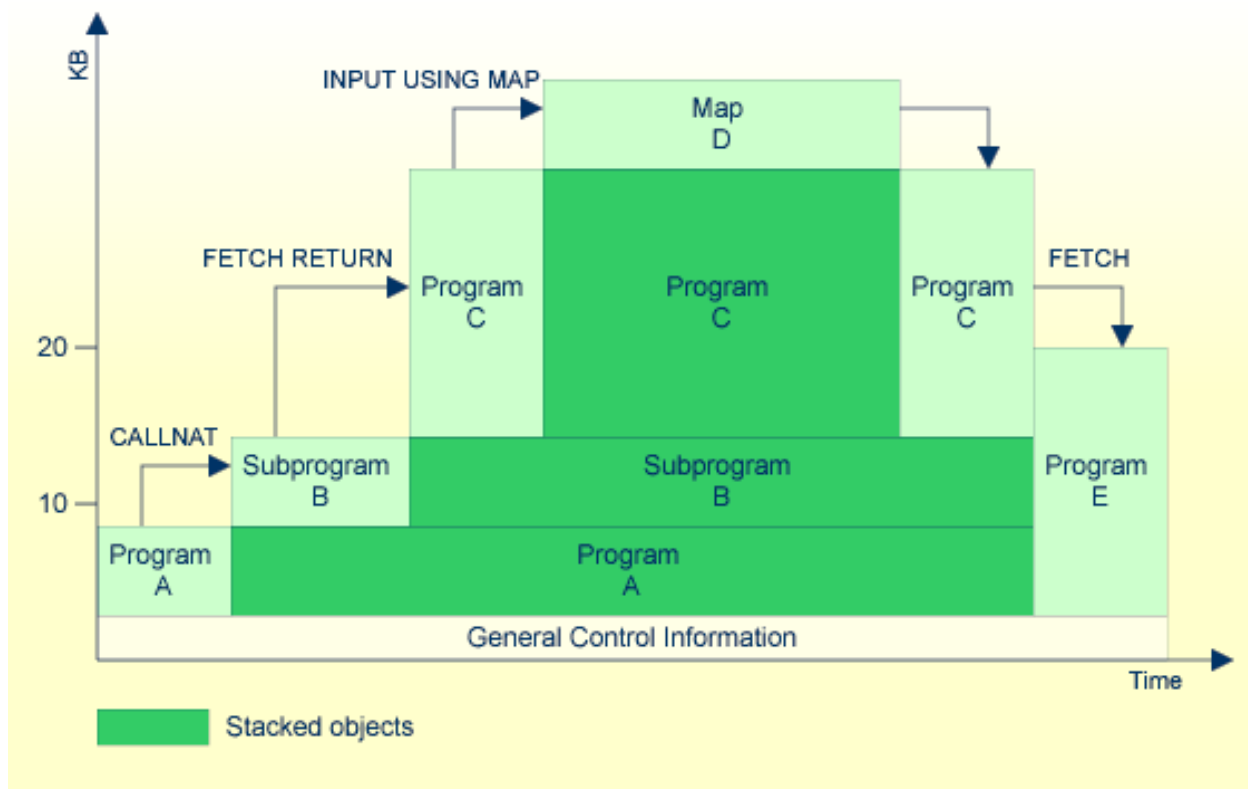
The actual DATSIZE requirement can be calculated as follows (refer to the illustration below):

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

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

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

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



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

53

DB - Database Types and Options

■ DB Parameter Syntax	136
■ NTDB Macro Syntax	137
■ Possible Database Options	138
■ Examples of NTDB Macro	139
■ Examples of DB Parameter	139

This Natural profile parameter can be used to define database types and options for all and for specific database IDs.

It corresponds to the [NTDB](#) macro in the parameter module NATPARM.

Possible settings	<i>database-type</i>	<i>database-type</i> is besides ADABAS and its synonym ADAV7, for example, ADAV8, DLI, VSAM, DB2, etc. For further information on the corresponding Natural database management interface, see the <i>Database Management System Interfaces</i> documentation. This subparameter is mandatory for the NTDB macro.
	<i>database-ID</i>	<i>database-ID</i> must be in the range from 0 to 65535. Database ID 255 must not be specified, because it is reserved for internal use. You can specify a single database ID, a list of database IDs enclosed in parentheses, or an asterisk (*) to indicate the default for all databases not specified explicitly.
	<i>options</i>	<i>options</i> , see Possible Database Options below.
Default setting	ADABAS, *	Adabas Version 7.
Dynamic specification	yes	This parameter can only be specified dynamically. In the Natural parameter module NATPARM, the macro NTDB must be used instead.
Specification within session	no	

At compile time, Natural Data Manipulation Language (DML) statement functionality will be limited to the functionality that is available for the specified database type.

At runtime, the specified database type defines which Natural database management interface is called for a database ID.

The following topics are covered below:

DB Parameter Syntax

The DB parameter is specified as follows:

1. Default Database Definition

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

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

2. Single Database Definition

A single database ID is specified as follows:

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

3. Multiple Database Definition

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

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

NTDB Macro Syntax

The `NTDB` macro is specified as follows:

1. Default Database Definition

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

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

2. Single Database Definition

A single database ID is specified as follows:

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

3. Multiple Database Definition

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

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

Possible Database Options

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

ACODE	The Natural application must communicate to Adabas whether code page or Unicode support is desired if the Adabas DBID used is enabled for character encoding and data conversion. Therefore the ACODE setting specifying the application-specific code page for all A fields and/or the WCODE=4095 (UTF-16) setting for all W fields must be sent with the OP call. See also <i>Unicode and Code Page Support</i> .
WCODE	
ENTIRE	The database is to be handled by Entire DB.
ETP	The database is to be handled by Entire Transaction Propagator.
OPEN	This option applies to Adabas databases only, for which Adabas requires an open request to be issued. If OPEN is specified for such a database, an open request is always issued (even if the ETID is blank).
READ	The database is to be read-only.

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

NOENTIRE	Resets the ENTIRE option.
NOETP	Resets the ETP option.
NOOPEN	Resets the OPEN option.
NOREAD	Resets the READ option.
OFF	Removes any DB or NTDB definition for the specified databases, see Examples of DB Parameter below.

Examples of NTDB Macro

```
NTDB DLI,7
```

This defines Database 7 as DL/I database.

```
NTDB  
ADAV8,(10,15,57),ETP
```

In the Natural parameter module NATPARM, this defines Databases 10, 15 and 57 as Adabas Version 8 databases which are to be handled by Entire Transaction Propagator.

Examples of DB Parameter

```
DB=(VSAM,(22,26,33))
```

This defines Databases 22, 26 and 33 as VSAM databases.

```
DB=(*,READ)
```

This sets all databases for which the default database definition applies to read-only.

```
DB=(8,9),NOREAD)
```

This removes the read-only option for Databases 8 and 9.

```
DB=(17,OFF)
```

This resets the database definition of Database 17 to defaults.

54

DB2SIZE - Natural Buffer Area for DB2 or SQL/DS

This Natural profile parameter applies to Natural for DB2 and Natural for SQL/DS.

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

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

55

DBCLOSE - Database Close at Session End

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

Possible settings	ON	Natural closes all databases.
	OFF	<p>Natural closes only those databases which had been opened by an explicit open command.</p> <p>An explicit open command will be issued in the following cases:</p> <ul style="list-style-type: none"> ■ profile parameter ETID is not set to ' ' (blank), ■ profile parameter DBOPEN=ON, ■ the open is forced by the OPEN option of macro NTDB or profile parameter DB.
	ETDB	Natural closes only the database specified with the profile parameter ETDB .
Default setting	OFF	
Dynamic specification	yes	
Specification within session	no	

Other transaction processing related parameters: [ADAMODE](#) | [DBOPEN](#) | [ENDBT](#) | [ET](#) | [ETDB](#) | [ETEOP](#) | [ETIO](#) | [ETSYNC](#)

56

DBGERR - Automatic Start of Debugger at Runtime Error

This Natural profile parameter enables the Natural Debugger to be started automatically if a Natural error occurs at runtime.

The runtime environment will cede control to the Debugger in the event of a Natural error, independent of whether the Debugger is already on or not. This measure avoids the manual control effort of using the Natural system command `TEST ON` in such a case.

Possible settings	ON	The Debugger is started automatically and produces a screen which enables you to get further information on the existing error.
	OFF	The Debugger is not started automatically.
Default setting	OFF	
Dynamic specification	yes	
Specification within session	no	

For further information, see *Start the Debugger* in the *Natural Debugger* documentation.

57

DBID - Default Database ID of Natural System Files

This Natural profile parameter identifies the default database in which the Natural system files (FNAT, FUSER, FDIC, FSEC, FSP00L) are located.

Possible settings	0 - 254, 256 - 65535	Database ID. Note: Database ID 255 is reserved for internal use.
Default setting	0	
Dynamic specification	yes	If you specify the DBID parameter dynamically, the database ID for all system files is set to this setting. Therefore, you must specify the DBID parameter <i>before</i> any individual system file parameter (FNAT, FUSER, FDIC, FSEC, FSP00L) if you want to specify any of these parameters, too.
Specification within session	no	

Database IDs for individual system files can be specified with the parameters FNAT, FUSER, FDIC, FSEC and FSP00L. The database ID specified with the DBID parameter applies to all Natural system files for which no individual database IDs are specified.

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

58

DBOPEN - Database Open without ETID

This Natural profile parameter controls the database open handling of Natural. DBOpen overrides the setting `ETID=' '` (blanks).

Possible settings	ON	A database open will be issued even if the ETID parameter is set to blanks.
	OFF	No database open will be issued if the ETID parameter is set to blanks. Exception: One open command will always be sent to the database specified as <code>ETDB</code> , even if ETID is set to blanks and DBOpen is set to OFF.
Default setting	OFF	
Dynamic specification	yes	
Specification within session	no	

Other transaction processing related parameters: `ADAMODE` | `DBCLOSE` | `ENDBT` | `ET` | `ETDB` | `ETEOP` | `ETIO` | `ETSYNC`

59

DBROLL - Database Calls before Session Suspension

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

It determines the number of database calls after which a Natural session is suspended, that is, a potential roll-out of the Natural thread is to be performed.

Possible settings	0 - 32767	Number of database calls.
Default setting	0	No session suspension for database calls.
Dynamic specification	yes	
Specification within session	no	

When the non-zero DBROLL count is reached, Natural issues a conditional CMROLL request. (see [Note Concerning CMROLL](#) in the description of profile parameter [MAXROLL](#)); that is, when other sessions are waiting for a thread, the session is suspended, which may result in a roll-out of the Natural thread. In CICS if no other session is waiting, just an EXEC CICS SUSPEND is executed to relinquish control to other tasks of higher or equal dispatching priority.

60

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. A Natural statement which would cause a database update (STORE, UPDATE, DELETE) is not executed and an error message is generated.
Default setting	ON	
Dynamic specification	yes	
Specification within session	no	
Application Programming Interface	USR1005N	See <i>SYSEXT - Natural Application Programming Interfaces</i> in the <i>Utilities</i> documentation.
	USR1042N *	
		* Recommended.

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

61

DC - Character for Decimal Point Notation

This Natural profile and session parameter determines the character to be used for decimal point notation.

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

Possible settings	any character (except numeric characters)	You specify the <code>DC</code> parameter as <code>DC=' c '</code> where <i>c</i> represents the character to be used as decimal point. The character specified with the <code>DC</code> parameter <ul style="list-style-type: none">■ must not be the same as the one specified with the <code>IA</code> profile/session parameter (input assign character) or <code>ID</code> profile/session parameter (input delimiter character),■ should not be the same as the one specified with the <code>CF</code> profile/session parameter (control character for terminal commands) or <code>HI</code> profile parameter (help character).		
Default setting	. (period)			
Dynamic specification	yes			
Specification within session	yes	Applicable Statements:	SET GLOBALS	Parameter is evaluated at runtime.
		Applicable Command:	GLOBALS	Parameter may be specified dynamically with the GLOBALS system command.
Application Programming Interface	USR0350N, USR1005N *	See <i>SYSEXT - Natural Application Programming Interfaces in the Utilities</i> documentation. * Recommended.		

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

62 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	-32767 to +32767	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 - <i>Natural Application Programming Interfaces</i> in the <i>Utilities</i> documentation.

See also the profile parameters [TD](#) and [YD](#).

63

DELETE - Deletion of Dynamically Loaded Programs

This Natural profile parameter determines whether dynamically loaded non-Natural programs are to be deleted on the completion of the Natural program in which they are used.

Possible settings	ON	Dynamically loaded non-Natural programs are deleted at the end of the Natural program in which they were loaded.
	OFF	Dynamically loaded non-Natural programs are not deleted at the end of the Natural program in which they were loaded; they are kept until command mode is entered again, and the deletion is performed immediately before Natural enters command mode processing.
Default setting	ON	
Dynamic specification	yes	
Specification within session	no	
Application Programming Interface	USR1005N	See <i>SYSEXT - Natural Application Programming Interfaces</i> in the <i>Utilities</i> documentation.

The following platform-specific requirements apply:

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

64

DF - Date Format

With the `DF` session parameter, you determine the length of a date when converted to alphanumeric representation without an edit mask being specified. The `DF` parameter is evaluated at compilation time.

The sequence of the day, month and year components and the delimiter characters used are determined by the profile parameter `DTFORM`.

Possible settings	S	8-byte representation with 2-digit year component and delimiters (<i>yy-mm-dd</i>). With <code>DF=S</code> , 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 (<i>yyyymmdd</i>). * See Note.
	L	10-byte representation with 4-digit year component and delimiters (<i>yyyy-mm-dd</i>). * 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	

When the value of a date field is converted to alphanumeric format (for example, in a `MOVE`, `DISPLAY`, `WRITE` or `INPUT` statement) and no edit mask is specified for the conversion, the default date format as determined by the profile parameter `DTFORM` is used as edit mask. The same is true for the input validation of a date variable used in an `INPUT` statement: If no edit mask is specified, the input is validated according to the date format determined by the `DTFORM` parameter.



Note: By using `DF=I` or `DF=L`, you can gradually change your applications to use 4-digit year representations and at the same time continue to make use of the flexibility provided by the profile parameter `DTFORM`.

See also Processing of Date Information, *Date Format for Alphanumeric Representation - DF Parameter* in the *Programming Guide*.

65

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.

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

Possible settings	S	Date variables are displayed with a 2-digit year component, and delimiters as determined by the profile parameter DTFORM . For example: <i>yy-mm-dd</i> .	
	I	Date variables are displayed with a full 4-digit year component and no delimiters. For example: <i>yyyymmdd</i> .	
Default setting	S		
Dynamic specification	yes		
Specification within session	yes	Applicable Statements:	SET GLOBALS
		Applicable Command:	GLOBALS
Application Programming Interface	USR1005N	See <i>SYSEXT - Natural Application Programming Interfaces</i> in the <i>Utilities</i> documentation.	



Notes:

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

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

66

DFS - Specify RPC Client's Default Server Address

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

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

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

DFS is specified on the client side only.

Possible settings	<i>server-name</i>	(1 - 192 characters). There is no default, the value must be specified.	
	<i>server-node</i>	(1 - 192 characters). There is no default, the value must be specified.	
	<i>logon-indicator</i>	L	The client initiates a Natural logon to the server with the library name of the current library on the client.
		(blank)	No server logon will be executed.
		If nothing is specified, blank is the default. Note for Windows platforms: Instead of specifying L, check the selection box.	
	<i>transport-protocol-name</i>	The transport protocol to be used. ACI is the only possible value and the default.	

	<i>service-directory-indicator</i>	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.	
Default setting	none	Subparameter defaults, see above.	
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 <i>Application Programming Interfaces for Use with Natural RPC</i> in the <i>Natural Remote Procedure Call (RPC)</i> documentation and <i>SYSEXT - Natural Application Programming Interfaces</i> in the <i>Utilities</i> documentation.	

For dynamic specification, the syntax is as follows:

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

For the possible values of *server-name* and *server-node*, refer to [SRVNAME](#) and [SRVNODE](#).

For further information, see the *Natural Remote Procedure Call (RPC)* documentation.

67

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.

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

Possible settings	S	Date variables are placed on the stack with a 2-digit year component, and delimiters as determined by the profile parameter DTFORM . For example: <i>yy-mm-dd</i> .	
	C	Same as <code>DFSTACK=S</code> . In addition, if the century used when the setting is read from the stack is not the same as that of the original date setting, Natural will issue a runtime error.	
	I	Date variables are placed on the stack with a full 4-digit year component and no delimiters. For example: <i>yyyymmdd</i> .	
Default setting	S		
Dynamic specification	yes		
Specification within session	yes	Applicable Statements:	SET GLOBALS
		Applicable Command:	GLOBALS
Application Programming Interface	USR1005N	See <i>SYSEXT - Natural Application Programming Interfaces</i> in the <i>Utilities</i> documentation.	

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

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

68

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).

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

Possible settings	S	The date is output with a 2-digit year component and delimiters. For example: <i>yy-mm-dd</i> .	
	L	The date is output with a 4-digit year component and delimiters. For example: <i>yyyy-mm-dd</i> .	
	I	The date is output with a 4-digit year component and no delimiters. For example: <i>yyyymmdd</i> .	
Default setting	S		
Dynamic specification	yes		
Specification within session	yes	Applicable Statements:	SET GLOBALS
		Applicable Command:	GLOBALS
Application Programming Interface	USR1005N	See <i>SYSEXT - Natural Application Programming Interfaces</i> in the <i>Utilities</i> documentation.	



Notes:

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

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

69

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 topics in the *Programming Guide*:

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

70

DLISIZE - Size of Natural Buffer Area for DL/I

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

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

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

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

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

71

DO - Display Order of Output Data

This Natural profile and session parameter specifies how fields are to be interpreted for display on terminals that support bidirectional data.

The I/O device must be able to create the correct display order depending on the character properties. This is for instance the case if an application runs in a browser under Natural Web I/O Interface. For other terminal types, this parameter does not have any effect.

Possible settings	L	Specifies that the data from the application is in logical display order. The field characters are displayed according to their character property (left-to-right or right-to-left).	
	V	Specifies that the data from the application is in visual order. All fields are inverted by Natural before they are sent to the terminal. This option is required for old applications written for terminals which support inverse (right-to-left) print mode using profile parameter <code>PM=I</code> or terminal command <code>%VON</code> .	
Default setting	L		
Dynamic specification	yes		
Specification within session	yes	Applicable Statement:	SET GLOBALS
		Applicable Command:	GLOBALS

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

72

DS - Define Size of Storage Buffer

■ DS Parameter Syntax	179
■ NTDS Macro Syntax	179
■ Table of Buffer Sizes	179
■ Examples	181

This Natural profile parameter defines the default initial size of various Natural storage buffers.

In previous versions of Natural, individual profile parameters (for example, [SSIZE](#)) were used to define the sizes of the buffers. The `DS` profile parameter is a universal parameter to specify all buffer sizes. It corresponds to the `NTDS` macro in the Natural parameter module `NATPARM`.

See also *Natural Storage Management* and *General Rules for Parameter Usage* in the *Operations* documentation.



Note: There are some buffer sizes (for example, [ESIZE](#), [VSIZE](#), etc.) which cannot be specified by the profile parameter `DS`, due to certain reasons, for example, the size is part of a larger buffer or the size defines the total maximum of a number of buffers.

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

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

The following topics are covered below:

DS Parameter Syntax

The DS parameter is specified as follows:

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

NTDS Macro Syntax

The NTDS macro is specified as follows:

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

Table of Buffer Sizes

Buffer Name	Description	Buffer Size (KB)	Default	Available as subparameter of DS and alternatively as individual profile parameter
ASIZE	Entire System Server auxiliary buffer	0, 1-64	0	yes
BSIZE	Size of EntireX Broker buffer	0, 1-64	0	yes
CSIZE	Size of Con-nect/Con-form buffer area	0-512	0	yes
DATSIZE	Size of buffer for local data	10-2097151	32	yes
DSIZE	Initial size of DBLOG buffer	0, 2 - 2097151	2	yes Note: The individual profile parameter DSIZE allows you to set a maximum size in addition.
EDPSIZE	Size of the Software AG Editor auxiliary buffer pool	0, 48-2097151	0	yes
ETPSIZE	Size of Entire Transaction Propagator buffer	0, 10-128	0	yes
EXCSIZE	Size of buffer for Natural Expert C interface	0, 1-256	0	yes

Buffer Name	Description	Buffer Size (KB)	Default	Available as subparameter of DS and alternatively as individual profile parameter
EXRSIZE	Size of buffer for Natural Expert rule tables	0, 1-256	0	yes
MONSIZE	Size of SYSTP monitor buffer	0, 5-256	0	yes
MULFETCH	Size of Multi-fetch buffer	0-1024	64	no (only available as subparameter of DS) Note: A value specified for this buffer does not represent the default initial size but the maximum size which can be allocated for multi-fetch purposes.
NAFSIZE	Size of buffer for Natural Advanced Facilities	0, 1-64	0	yes
NSFSIZE	Size of SAF interface buffer.	0, 8-64	0	no (only available as subparameter of DS)
RDCSIZE	Size of buffer for the Natural Data Collector	0, 2-128	0	yes
RJESIZE	Initial size of NATRJE buffer	0, 1-2097151	8	yes
RUNSIZE	Size of runtime buffer	10-64	16	yes
SSIZE	Size of Software AG Editor buffer	0, 40-512	64	yes
TSIZE	Size of the buffer for Adabas Text Retrieval	0, 1-2097151	0	yes
XSIZE	Size of buffer for user subsystem	0, 1-64	0	yes
ZSIZE	Size of Entire DB buffer area	0, 1-64	0	yes

For more information, refer to the descriptions of the individual buffer size parameters.

Examples

Example of DS parameter:

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

Equivalent in Natural parameter module NATPARM:

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


73

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

This Natural profile parameter only applies to 3270-type terminals.

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



Note: Screen optimization means that only those fields of the screen are sent to the terminal whose content has changed. Screen compression constitutes a (further) reduction of the amount of data sent by using counters for repeating characters.

Possible settings	ON	Data-stream optimization and compression are used.
	OFF	Data-stream optimization and compression are not used.
Default setting	ON	
Dynamic specification	yes	
Specification within session	no	

Natural's screen optimization causes screen data to be sent as compressed as possible. If this should conflict with any TP monitor's screen optimization or hardware limitation, you can use this parameter to switch off Natural's screen optimization; screen data will then be sent in non-compressed form; for example, see *Profile Parameter DSC=OFF Recommended* in the *Natural under CICS* documentation.

This parameter has the same function as the terminal command %R0.

If you use the BX session parameter settings BX=L or BX=R, you should switch off Natural's screen optimization using DSC=OFF or %R0OFF.

74

DSIZE - Size of DBLOG Buffer

This Natural profile parameter specifies the size of the Natural DBLOG buffer.

Alternatively, you can use the Natural profile parameter [DS](#) or macro NTDS (see *Using Optional Macros in a Natural Parameter Module* in the *Operations* documentation) to specify DSIZE.

Possible settings	<i>initial-size</i>	2 - 2097151	DBLOG buffer initial size in KB. If the initial size is not sufficient, Natural automatically increases the buffer size (repeatedly, if necessary) up to the specified maximum (see below).
	0		Disables the DBLOG utility.
	<i>maximum-size</i>	2 - 2097151	DBLOG buffer maximum size in KB. If the value is not greater than the initial size (see above), the DSIZE buffer is not increased.
Default setting	2,256		
Dynamic specification	yes		
Specification within session	no		

The Natural DBLOG buffer area is used by the DBLOG utility, which is described in the *Utilities* documentation.

Examples:

```
DSIZE=100
```

```
DSIZE=( ,2500)
```

```
DSIZE=(50,800)
```


75

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 <i>SYSEXT - Natural Application Programming Interfaces</i> in the <i>Utilities</i> documentation.	

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

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

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

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

76

DU - Dump Generation

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

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

Possible settings	ON	A memory dump is produced in the case of an abnormal termination (TP-monitor dump dataset or <code>SYSDUMP</code> in z/OS batch mode or under TSO). Then the Natural session terminates with the error message NAT9967 or NAT9974.
	OFF	No memory dump is produced. In batch mode, subsequent action taken by Natural is determined by the setting of the <code>CC</code> profile parameter. In online mode, Natural responds with the error message NAT0950, NAT0953, NAT0954, NAT0955 or NAT0956. For further information on the abnormal termination, you can use the system command <code>DUMP</code> .
	SNAP	This setting forces an immediate dump in the case of an abnormal termination during a Natural session. The Natural session continues as with <code>DU=OFF</code> after the dump has been taken.
	FORCE	This setting forces an immediate dump in the case of an abnormal termination during a Natural session and terminates the Natural session immediately. This is useful for testing purposes in some environments. Note: If Natural is LE enabled, Natural terminates the Natural session immediately without a dump and passes control to the LE error handling. It is therefore strongly recommended to specify the LE run-time option <code>TERMTHDACT(UAImm)</code> to get all the required diagnostic informations.
	ABEND	This works as with <code>DU=ON</code> , except that the session is terminated with the abend occurred - instead of the error message NAT9974. <code>DU=ABEND</code> is not available with the Natural session parameter <code>DU</code> .
Default setting	OFF	

Dynamic specification	yes		
Specification within session	yes	Applicable Statements:	SET GLOBALS
		Applicable Command:	GLOBALS

**Notes:**

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

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

77

DUE - Dump for Specific Errors

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

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

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

Examples:

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


78

DY - Dynamic Attributes

■ DY Parameter Syntax	194
■ Examples	196

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

Possible settings		See below.
Default setting	none	
Applicable statements:	DISPLAY INPUT PRINT WRITE	Parameter may be specified at statement level and/or at element level.
Applicable command:	none	

Special identification characters (escape characters) are used to indicate the beginning and end of attribute definitions.

An alphanumeric field which is processed with an INPUT, DISPLAY, WRITE or PRINT statement, and which contains escape characters, is split into subfields at the escape character position. The corresponding attribute is then assigned to the subfield. A blank is substituted for the escape character.

For a part of a field for which a DY specification applies, the current field presentation and color is changed to what is newly defined in the DY entry. If the DY segment *does not contain a new setting* for the

■ field presentation

(means *no* B, C, D, I, N, U, V), the attribute active for the complete field remains in effect, regardless of whether originally derived from a static setting (for example, AD=I) or from a control variable (for example, CV=#C).

■ field color

(means *no* BL, GR, NE, PI, RE, TU, YE), the color is set to what is statically assigned to the field (with CD=. .), without considering a color which was possibly set via a control variable (CV=. .). If the field has no static (CD=. .) assignment, the color information is completely removed from the field segment affected by the DY manipulation.

DY Parameter Syntax

DY={{ <i>escape-character1</i> } [<i>color-attribute</i>] [<i>i/o-characteristics</i>] [<i>field-representation-attribute</i>]} ... { <i>escape-character2</i> }

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 <i s> UNDERLINED
```

is printed as:

```
THIS i s 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)

79

DYNPARM - Control Use of Dynamic Parameters

■ DYNPARM Parameter Syntax	201
■ NTDYNP Macro Syntax	201
■ Examples	201

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

Possible settings	ON	All profile parameters can be specified dynamically.
	OFF	No profile parameters can be specified dynamically.
	DYNPARM=(ON, <i>parameter-name</i> ,...)	Only those parameters whose <i>parameter-name</i> is specified, can be specified dynamically. Other parameters cause Error Message NAT7008 to be issued.
	DYNPARM=(OFF, <i>parameter-name</i> ,...)	All profile parameters can be specified dynamically - except those whose <i>parameter-name</i> is specified. These parameters cause Error Message NAT7008 to be issued.
Default setting	ON	All profile parameters can be specified dynamically.
Dynamic specification	yes	Outside of PROFILE or " SYS " parameter strings, the DYNPARM parameter can be used only once and only if the NTDYNP macro is not specified in the Natural parameter module.
Specification within session	no	
Application Programming Interface	USR1005N	See SYSEXT - Natural Application Programming Interfaces in the Utilities documentation.

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

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

The following topics are covered below:

DYNPARM Parameter Syntax

The DYNPARM parameter is specified as follows:

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

or

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

NTDYNP Macro Syntax

The NTDYNP macro is specified as follows:

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

or

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

Examples

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

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

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

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

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

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

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

80

ECHO - Control Printing of Batch Input Data

This Natural profile parameter only applies in batch mode.

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

Possible settings	ON	Natural prints the input data provided during batch mode processing to the batch output file CMPRINT.
	OFF	Natural does <i>not</i> print input data provided during batch processing.
Default setting	ON	
Dynamic specification	yes	
Specification within session	no	

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

81

EDBP - Software AG Editor Buffer Pool Definitions

- EDBP Parameter Syntax 207
- NTEDBP Macro Syntax 207
- Keyword Subparameters 207

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

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

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

Shared Editor Buffer Pool

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

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

Editor Auxiliary Buffer Pool

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

FTOUT, **LRECL**, **MAXLF**

More Information

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

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

The following topics are covered below:

EDBP Parameter Syntax

The EDBP parameter is specified as follows:

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

NTEDBP Macro Syntax

The NTEDBP macro is specified as follows:

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

Keyword Subparameters

The following keyword subparameters are available:

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

COLD - Buffer Pool Cold Start

Determines whether a buffer pool cold start is performed.

Possible values	ON or OFF
Default value	OFF

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

CTOUT - Timeout for Changed Buffer Pool Blocks

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

Possible values	1-32767
Default value	120

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

DDNAME - Logical Work File Name of the JCL Definition

Determines the logical work file name of the JCL definition.

Possible values	1 to 8 bytes
Default value	CMEDIT

**Notes:**

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

DSNAME - Work File Dataset Name

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

Possible values	1-44 bytes
Default value	None

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

DTOUT - Logical File Timeout Check Value

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

Possible values	1-32767
Default value	300

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

FMODE - Work File Mode (VM/CMS and Com-plete/SMARTS only)

Determines the file mode for the work file.

Possible values	1-2 characters
Default value	A1

Under VM/CMS, the specified value determines the file mode for the work file. The file type is always DATA.

Under Com-plete/SMARTS, the value SM determines that a SMARTS work file is used. In this case, the SMARTS environment variable \$NAT_WORK_ROOT determines the path.

Under Com-plete/SMARTS, if a value other than SM is specified, a Com-plete SD file is used.

In a SMARTS environment without Com-plete, SM must be specified.

FTOUT - Timeout Value for Logical Files

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

Possible values	60-16777215
Default value	86400

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

IMSG - Buffer Pool Initialization and Termination Message

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

Possible values	ON or OFF
Default value	OFF

ITOUT - Buffer Pool Initialization Timeout Value

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

Possible values	1-32767
Default value	300

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

LRECL - Work File Record Length

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

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

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

Possible values	800-16384
Default value	4096

Under BS2000/OSD, the record length must be a multiple of 2048 bytes.

LTOUT - Timeout Value for Locked Buffer Pool Blocks

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

Possible values	1-32767
Default value	20

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

MAXLF - Maximum Number of Logical Files

Determines the maximum number of logical files.

Possible values	100-999999
Default value	1000

PWORK - Percentage of Work File Records Used as Work Records

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

Possible values	0-100
Default value	50

The remaining records are used as recovery records.

RECNUM - Number of Work File Records

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

Possible values	100-65535
Default value	200

This number determines the size of the work file.



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

RWORK - Percentage of Work Records Used for Regular Logical Files

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

Possible values	51-100
Default value	90

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

UTOUT - Timeout Value for Unchanged Buffer Pool Blocks

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

Possible values	1-32767
Default value	20

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

82

EDPSIZE - Size of Software AG Editor Auxiliary Buffer

Pool

This Natural profile parameter determines the size of the Software AG Editor auxiliary buffer pool.

It must be used when the Software AG Editor runs in a z/OS Parallel Sysplex environment. It allows the Software AG Editor to be run without a Software AG Editor (local or global) buffer pool.

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

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

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

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

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

For further information on the Software AG Editor, see *Operating the Software AG Editor* in the *Operations* documentation.

83

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.

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

Possible settings	ON	A page eject is performed.		
	OFF	No page eject is performed. This setting may be used to save paper during test runs where page ejects are not needed.		
Default setting	ON			
Dynamic specification	yes			
Specification within session	yes	Applicable Statements:	SET GLOBALS	Parameter is evaluated at runtime.
		Applicable Command:	GLOBALS	Parameter may be specified dynamically with the GLOBALS system command.
Application Programming Interface	USR1005N	See SYSEXT - <i>Natural Application Programming Interfaces</i> in the <i>Utilities</i> documentation.		

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



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

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

84

EM - Edit Mask

▪ Syntax	218
▪ Blanks in Edit Masks	219
▪ Default Edit Masks	219
▪ Edit Masks for Numeric Fields	220
▪ Edit Masks for Alphanumeric Fields	223
▪ Edit Masks for Binary Fields - Format B	225
▪ Hexadecimal Edit Masks	225
▪ Edit Masks for Date and Time Fields - Formats D and T	227
▪ Edit Masks for Logical Fields - Format L	232

With this session parameter, you can specify an edit mask for an input and/or output field that is used in a `DEFINE DATA`, `DISPLAY`, `INPUT`, `MOVE EDITED`, `PRINT`, `PROCESS PAGE` or `WRITE` statement.

The parameter `EM` 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`.

Possible settings		See below.
Default setting	none	
Applicable statements:	<code>FORMAT</code>	
	<code>DEFINE DATA</code> <code>DISPLAY</code> <code>INPUT</code> <code>PRINT</code> <code>WRITE</code>	Parameter may be specified at statement level and/or at element level.
	<code>MOVE EDITED</code>	Parameter may be specified at element level.
Applicable command:	none	

The following topics are covered below:

See also *Edit Masks - EM Parameter* in the *Programming Guide*.

Syntax

For input fields, values must be entered exactly matching the edit mask. If you would like to display the edit mask for an input field, the field should be defined as modifiable (`AD=M`).

For a database field, a default edit mask may have been defined in the DDM. If you specify with the `EM` parameter an edit mask for a database field, this edit mask specified will be used instead of any default edit mask which may be defined for the field in the DDM.

If you specify `EM=OFF` for a field, no edit mask will be used for the field, not even one that may be defined in the DDM.

At statement level of a `DISPLAY`, `FORMAT`, `INPUT` or `WRITE` statement, no detail field edit mask may be specified, except `EM=OFF`.

An edit mask overrides any settings for the session parameters `AL`, `NL` and `SG`.

The characters `9`, `H`, `X` and `Z` represent significant print positions in numeric (`9`, `Z`), hexadecimal (`H`), and alphanumeric (`X`) edit masks. For the difference between `9` and `Z`, see *Edit Masks for Numeric Fields*, below.

Examples:

```
DISPLAY AA(EM=OFF) AB(EM=XX.XX)
WRITE SALARY (EM=ZZZ,ZZ9)
```

An abbreviated notation can be used for long edit masks. The following examples demonstrate the abbreviated notation which may be used for numeric, hexadecimal, and alphanumeric edit masks:

```
EM=9(4)-9(5) is equivalent to: EM=9999-99999
EM=H(10)      is equivalent to: EM=HHHHHHHHHHH
EM=X(6)..X(3) is equivalent to: EM=XXXXXX..XXX
```

Blanks in Edit Masks

Blanks within an edit mask are represented by the character on your keyboard that in hexadecimal code corresponds to H'20' (ASCII) or H'5F' (EBCDIC), that is, the character ^ (or ¬).

Default Edit Masks

If no edit mask is specified for a field, a default edit mask is assigned to the field depending on the field format:

Field Format	Default Edit Mask
A	X
B	H
N, P, I	Z9
F	scientific representation
D	depends on default date format (as set with the profile parameter DTFORM)
T	HH:II:SS
L	blank / X

Edit Masks for Numeric Fields

An edit mask specified for a field of format N, P, I, or F must contain at least one 9 or Z.

If more 9s or Zs exist than the number of positions contained in the field value, the number of print positions in the edit mask will be adjusted to the number of digits defined for the field value.

If fewer 9s or Zs exist, the high-order digits before the decimal point and/or low-order digits after the decimal point will be truncated.

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)	<p>The first period inserted is used as a decimal point. Subsequent periods are treated as literal characters.</p> <p>Note: At this point, the period represents the sign currently defined as decimal point character. If another character is chosen (for example, a comma) with the session or profile parameter DC, this character is to be used instead.</p>
Z	<p>Zero suppression for leading zeros. This is the default for numeric fields. The letter Z may be repeatedly specified to represent floating zero suppression. Z must not be specified to the right of the decimal point. A zero value may be displayed as blanks using all Zs in the edit mask (see also session parameter ZP).</p>

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=*DMZZ9.9	DM**0.0	DM*54.	DM*87.	DM962.	DM830.
EM=999+	000+	054-	087+	962+	830+
EM=999-	000	054-	087	962	830
IC=\$ EM=ZZZ.99	\$.03	\$54.	\$87.	\$962.	\$830.
EM=H(6)					
- ASCII:	303030303033	30303574	30303837	30393632	31383330
- EBCDIC:	F0F0F0F0F0F3	F0F0F5D4	F0F0F8F7	F0F9F6F2	F1F8F3F0

By combining edit masks with the parameters IC and TC, negative numbers can be displayed in varying formats using a DISPLAY statement.

Edit Masks for Alphanumeric Fields

An alphanumeric edit mask which is only to be used with A format fields must contain at least one X which represents a character to be displayed. An H as the first character designates a **hexadecimal edit mask**. A blank is represented by a (^) symbol. All other characters except parentheses are permissible including leading, trailing, and insertion characters. It is also possible to specify leading, insertion, or trailing characters enclosed within apostrophes. If the character X, a closing parenthesis, or a quotation mark is specified as an insertion character, it must be enclosed within apostrophes.

If leading characters are used before the first displayable position X 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)
WRITE #TEXT (EM=X-X-X-X-X)
```

displays as 'B-L-U' .. only three bytes of field displayed
displays as 'B-L-U-E-' .. mask definition was truncated to (EM=X-X-X-X-)

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 point positions.

If a hexadecimal edit mask is used as an input edit mask, every 0-9, a-f, A-F, blank and hex zero are accepted as a hex digit.



Note: Blank and hex zero are regarded as 0 and a lower-case letter (a-f) is regarded as an upper-case letter.

Edit Mask Examples for Hexadecimal Fields:

The tables below list the hexadecimal edit masks with results obtained from the original fields and values shown above each column. All numeric values (-10, +10, 01) to which edit masks have been applied originated in fields defined with N2 format. The alphanumeric value AB originated from a field defined with format/length A2.

ASCII:

<i>Value =></i>	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:

<i>Value =></i>	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.
NN . . . or N(<i>n</i>)	Name of day (language-dependent). The maximum length is determined by the number of Ns or by <i>n</i> . 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.
0	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(<i>n</i>)	Name of month (language-dependent). The maximum length is determined by the number of L characters or by <i>n</i> . 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. The upper limit for displayable year values is 2887.

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(<i>n</i>)
day name	with	week day number	NN or N(<i>n</i>)	with	0

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(<i>n</i>)
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(<i>n</i>)	without	WW or ZW
Week day number	without	week	0	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 the profile parameter **MAXYEAR** is set to 9999, the range of valid year values is 1582 - 9999.

If only year (YY or YYYY) but no month or day is specified within an input edit mask, the values for month and day will both be set to 01. If only year (YY or YYYY) and month (MM) but no day is specified within an input edit mask, the value for day will be set to 01.

If a 2-digits year (YY) is used, the century used to fill up the year representation is the current century by default. However, this does not apply when a Sliding or Fixed Window is set. For more details, please refer to profile parameter **YSLW** in the Natural Parameter Reference documentation.

If a week number (WW or ZW) but no number of week day (0) or name of day (NN. . .) is specified, the first day of the week is assumed.

Hints for Week Display (WW or ZW) in Output Edit Mask

When DTFORM=U (USA format) is set, the week starts on Sunday; whereas for all other DTFORM settings the first weekday is Monday. Whether a week is week 52/53 of the old year or week 01 of the new year depends on which year contains more days of the week. In other words, if Thursday (Wednesday for DTFORM=U) of that week is in the previous year, the week belongs to the previous year; if it is in the next year, the week belongs to the next year.

If the number of week (WW or ZW) and a year representation (YYYY or YY or Y) is in the same edit mask, the display for year always corresponds to the week number, regardless of the year in the underlying date field.

Example:

```
DEFINE DATA LOCAL
1 D (D)
END-DEFINE
MOVE EDITED '31-12-2003' TO D(EM=DD-MM-YYYY)
DISPLAY D(EM=DD-MM-YYYY_N(10)) D(EM=DD-MM-YYYY/WW)
```

Although the underlying date is the 31 Dec. 2003, when the week number WW is contained in the edit mask, it displays as:

```
          D          D
-----
31-12-2003_Wednesday 31-12-2004/01
```

Time - Format T - only

Character	Usage
T	Tenths of a second.
SS	Seconds.
ZS	Seconds, with zero suppression.
II	Minutes.
ZI	Minutes, with zero suppression.
HH	Hours.
ZH	Hours, with zero suppression.
AP	AM/PM element.

Examples of Date and Time Edit Masks

```

** Example 'EMDATI': Edit 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

85

EMFM - Edit Mask Free Mode

This Natural profile parameter is used to activate/deactivate the Edit Mask Free mode at session startup. This mode allows you to omit literals during input into a field with a numeric edit mask.

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

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

86

ENDBT - BACKOUT TRANSACTION at Session End

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

Possible settings	ON	Natural will issue an implicit BACKOUT TRANSACTION statement at session end.
	OFF	Natural will not issue an implicit BACKOUT TRANSACTION statement at session end.
	ETDB	Natural will issue an implicit BACKOUT TRANSACTION statement at session end only for the database specified with the profile parameter ETDB .
Default setting	ON	
Dynamic specification	yes	
Specification within session	no	

Other transaction processing related parameters: [ADAMODE](#) | [DBCLOSE](#) | [DBOPEN](#) | [ET](#) | [ETDB](#) | [ETEOP](#) | [ETIO](#) | [ETSYNC](#)

87

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 will be displayed at the end of the session.
	OFF	Message NAT9995 will not be displayed at the end of the session.
Default setting	ON	
Dynamic specification	yes	
Specification within session	no	

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

88

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 <code>DISPLAY</code> or <code>WRITE</code> statement which contains all blank values will not be printed. This setting is particularly useful when displaying arrays (for example, multiple-value fields or fields contained within a periodic group) to avoid printing a large number of empty lines.
	OFF	Empty line suppression is disabled.
Default setting	OFF	
Specification within session	yes	
Applicable statements:	DISPLAY FORMAT WRITE	
Applicable command:	none	

To achieve empty suppression for numeric values, the field must be specified with `ZP=OFF` and `ES=ON` in order to have null values printed as blanks. See also the session parameters [IS](#) and [ZP](#).

Example:

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

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

89

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	

90

ESIZE - Size of User-Buffer Extension Area

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

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

The user-buffer extension area contains:

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

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

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

91 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.

Other transaction processing related parameters: [ADAMODE](#) | [DBCLOSE](#) | [DBOPEN](#) | [ENDBT](#) | [ETDB](#) | [ETEOP](#) | [ETIO](#) | [ETSYNC](#)

92

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=' ' (blank), no error transaction program is called.
Default setting	blank	
Dynamic specification	yes	
Specification within session	yes	
Application Programming Interface	USR1041N	With USR1041N you can install your own error transaction, where USR1041P serves as example how to build such a routine. See <i>SYSEXT - Natural Application Programming Interfaces</i> in the <i>Utilities</i> documentation.

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



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

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

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

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

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

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

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

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

93

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

Other transaction processing related parameters: [ADAMODE](#) | [DBCLOSE](#) | [DBOPEN](#) | [ENDBT](#) | [ET](#) | [ETEOP](#) | [ETIO](#) | [ETSYNC](#)

94 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 <code>END TRANSACTION</code> statement at the end of a Natural program.
	OFF	Natural will not issue any implicit <code>END TRANSACTION</code> statement at the end of a Natural program.
Default setting	OFF	
Dynamic specification	yes	
Specification within session	no	

Other transaction processing related parameters: [ADAMODE](#) | [DBCLOSE](#) | [DBOPEN](#) | [ENDBT](#) | [ET](#) | [ETDB](#) | [ETIO](#) | [ETSYNC](#)

95

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

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

If the `ETID` setting is the same as the setting of `*INIT-USER`, or if the `ETID` parameter is not specified, Natural issues an Adabas open with the `*INIT-USER` setting as `ETID` (and the `OPRB` setting, if spe-

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

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

96

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 <code>END TRANSACTION</code> statement whenever a terminal I/O occurs. Whenever a transaction monitor commits the associated databases because of a terminal I/O, all related databases are also committed. This is useful for the synchronization of database transactions. Note: Natural add-on products (except for Natural Security) may not function correctly with <code>ETIO=ON</code> .
	OFF	Natural will issue no implicit <code>END TRANSACTION</code> statements upon terminal I/Os.
Default setting	OFF	
Dynamic specification	yes	
Specification within session	no	

Other transaction processing related parameters: [ADAMODE](#) | [DBCLOSE](#) | [DBOPEN](#) | [ENDBT](#) | [ET](#) | [ETDB](#) | [ETEOP](#) | [ETSYNC](#)

97

ETPSIZE - Size of Entire Transaction Propagator Buffer

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

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

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

98

ETRACE - External Trace Function

This Natural profile parameter is used to activate/deactivate the (normal) external trace function or the Generalized Trace Facility (GTF) offered under z/OS and TSO.



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

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

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

99

ETSYNC - Issue Syncpoint upon End of

Transaction/Backout Transaction

This Natural profile parameter determines whether or not an implicit syncpoint is issued whenever an `END TRANSACTION` or `BACKOUT TRANSACTION` statement is to be issued.

Possible settings	ON	Natural issues an implicit syncpoint <code>COMMIT</code> whenever an <code>END TRANSACTION</code> statement is to be issued. Natural issues an implicit syncpoint <code>ROLLBACK</code> whenever a <code>BACKOUT TRANSACTION</code> statement is to be issued. This is useful for the synchronization of database transactions that are performed from within 3GL programs.
	OFF	Natural does not issue an implicit syncpoint when an <code>END TRANSACTION</code> or <code>BACKOUT TRANSACTION</code> statement is to be issued.
Default setting	OFF	
Dynamic specification	yes	
Specification within session	no	

Notes:

To issue syncpoints, Natural uses

- Resource Recovery Services (RRS) under TSO and in batch mode to commit or rollback the unit of recovery,
- CICS commands `SYNCPOINT` and `SYNCPOINT ROLLBACK` under CICS,
- system service calls `CHECKPOINT (CHKP)` and `ROLLBACK (ROLB)` under IMS TM.

The processing sequence is as follows:

- an `END TRANSACTION / BACKOUT TRANSACTION` statement is issued to the database specified with the profile parameter `ETDB`,
- the syncpoint `COMMIT / ROLLBACK` is issued,
- `END TRANSACTION` or `BACKOUT TRANSACTION` statements are issued to the remaining databases.

Restrictions and Limitations:

- This functionality is available under the z/OS operating system
 - in batch mode,
 - under the TP monitor CICS,
 - under the TP monitor TSO,
 - under the TP monitor IMS TM in a non-message driven BMP (in all other environments under IMS TM, only a `ROLLBACK` is executed, but no `CHECKPOINT`).
- To synchronize Adabas transactions, the Adabas Transaction Manager (ATM) must be installed.
- If you use this feature to commit transactions that update data stored in a DB2 database, you must configure Natural for DB2 or your 3GL application to use the RRSF interface.
- For transactions in batch mode that update data stored in a DL/I database, Resource Recovery Services are not supported due to a DL/I restriction. If, additionally, data stored in a DB2 database is updated in the same transaction, synchronization is performed by means of the DL/I synchronization mechanism.

As a consequence, if data stored in an Adabas database is updated in addition to data stored in DB2 and DL/I databases, no synchronization is possible, not even if the Adabas Transaction Manager is installed.

Other transaction processing related parameters: `ADAMODE` | `DBCLOSE` | `DBOPEN` | `ENDBT` | `ET` | `ETDB` | `ETEOP` | `ETIO`

100

EXCSIZE - Size of Buffer for Natural Expert C Interface

This Natural profile parameter determines the size of the buffer required by the C interface of Natural Expert. See the Natural Expert documentation.

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

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

101

EXRSIZE - Size of Buffer for Natural Expert Rule Tables

This Natural profile parameter determines the size of the buffer required by the rule tables of Natural Expert. See the Natural Expert documentation.

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

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

102

FAMSTD - Overwriting of Print and Work File Access

Method Assignments

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

See also the AM subparameter of the macros [NTPRINT](#) and [NETWORK](#).

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

103

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. It is used to pre-fill fields non-protected input fields (field attribute specification <code>AD=A</code>) when fields are written to a terminal by an <code>INPUT</code> statement. For modifiable input fields (field attribute specification <code>AD=M</code>), it is used to fill the rest of the field.
Default setting	X'00'	For TTY or batch mode, the default setting is X'40', i.e. blank in hexadecimal format.
Dynamic specification	yes	
Specification within session	no	

104

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.

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

Possible settings	any character	Filler character for individual headings. FC only applies if the column width is determined by the field length and not by the header (see also session parameter <code>HW</code>); otherwise the <code>FC</code> setting will be ignored.	
Default setting	blank		
Specification within session	yes	Applicable Statements:	DISPLAY FORMAT
		Applicable Command:	none

Example:

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


105

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).

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

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

Possible settings	ON	Dynamically protected input fields are displayed filled with filler characters. This may suggest to the users that they could enter something in the fields.	
	OFF	Dynamically protected input fields are displayed filled with blanks.	
Default setting	ON		
Dynamic specification	yes		
Specification within session	yes	Applicable Statements:	SET GLOBALS
		Applicable Command:	GLOBALS
Application Programming Interface	USR1005N	See <i>SYSEXT - Natural Application Programming Interfaces</i> in the <i>Utilities</i> documentation.	

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 _____

106

FDIC - Predict System File

This Natural profile parameter defines five subparameters for the Predict system file which Predict uses to retrieve and/or store data.



Note: 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.

Possible settings	<i>database-ID</i>	1 - 65535, except 255	Database ID 255 is reserved for logical system files for Software AG products, see profile parameter LFILE .
	<i>file-number</i>	1 - 65535:	File number.
	<i>password</i>	1 to 8 characters	The password is only required if the Predict system file has been password-protected using the Adabas security feature. Note for Natural with VSAM system files: The password is used to specify the logical name (DD or DLBL) of the system file as defined to VSAM. Example: FDIC=(10,5,SYSVSAM) For further information, see <i>Using Natural with VSAM System Files</i> in the <i>Natural for VSAM</i> documentation.
	<i>cipher-key</i>	1 to 8 numeric digits	Cipher key for the Predict system file. It is only required if the Predict system file has been ciphered using the Adabas security feature.
	R0		Read-only option. R0 indicates that the Predict system file is “read-only” and is only specified if modifications on the file are to be disabled.
Default setting	none		
Dynamic specification	yes		If you specify the FDIC parameter dynamically in conjunction with any of the parameters DBID , FNR , SYSPSW and SYSCIP ,

		you must specify the FDIC parameter <i>after</i> any of these other parameters.
Specification within session	no	

The syntax for this parameter is:

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

Examples:

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

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

107

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. 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
```


108

FNAT - Natural System File for System Programs

This Natural profile parameter is used to specify the database ID, file number, password and cipher key and read-only flag for the Natural system file for Natural system programs.

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

Possible settings	<i>database-ID</i>	1-65535, except 255	Database ID 255 is reserved for logical system files for Software AG products, see profile parameter LFILE .
	<i>file-number</i>	1-65535	File number.
	<i>password</i>	1 to 8 characters	The password is only required if the Natural system file has been password-protected using the Adabas security feature. For Natural with VSAM system files: The password is used to specify the logical name (DD or DLBL) of the system file as defined to VSAM. Example: FNAT=(22,5,SYSVSAM) For further information, see <i>Using Natural with VSAM System Files</i> in the <i>Natural for VSAM</i> documentation.
	<i>cipher key</i>	8 numeric digits	The cipher key is only required if the Natural system file has been ciphered using the Adabas security feature. The cipher key is reserved for future use; currently, it is ignored.
	R0		Read-only option. R0 indicates that the Natural system file is “read-only” and is only specified if modifications on the file are to be disabled.
Default setting	none		
Dynamic specification	yes		If you specify the FNAT parameter dynamically in conjunction with any of the parameters DBID , FNR , SYSPSW , SYSCIP or

		ROSY , you must specify the FNAT parameter <i>after</i> any of these parameters.
Specification within session	no	

The syntax of this parameter is:

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

Examples:

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



Notes:

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

109

FNR - Default File Number of Natural System Files

This Natural profile parameter identifies the default number of the file in which the Natural system files (FNAT, FUSER, FDIC, FSEC, FSP00L) are located.

Possible settings	1 - 65535	File number. It applies to all Natural system files for which no individual file numbers are specified.
Default setting	none	
Dynamic specification	yes	If you specify the FNR parameter dynamically in conjunction with any of the individual profile parameters which define the system files FNAT, FUSER, FDIC, FSEC and FSP00L, you must specify the FNR parameter <i>before</i> any individual system file parameter.
Specification within session	yes	

File numbers for individual system files can be specified with the profile parameters FNAT, FUSER, FDIC, FSEC and FSP00L.

Example 1:

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

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

Example 2:

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

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

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

111

FS - Default Format/Length Setting for User-Defined

Variables

This Natural profile and session parameter only applies to reporting mode; it has no effect in structured mode.

This parameter determines whether a default format/length setting is to be in effect for the definition of user-defined variables in reporting mode; see also *Format and Length of User-Defined Variables* in the *Programming Guide*.

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

Possible settings	ON	No default format/length is assigned by Natural for a newly introduced variable in reporting mode. The format/length of all user-defined variables must be explicitly specified.	
	OFF	A user-defined variable in a Natural program for which no format/ length is specified is assigned the default format/length N7.	
Default setting	OFF		
Dynamic specification	yes		
Specification within session	yes	Applicable Statements:	SET GLOBALS
		Applicable Command:	GLOBALS
Application Programming Interface	USR1005N	See <i>SYSEXT - Natural Application Programming Interfaces</i> in the <i>Utilities</i> documentation.	

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

112

FSEC - Natural Security System File

This Natural profile parameter only applies if Natural Security is used.

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

Possible settings	<i>database-ID</i>	1-65535, except 255	Database ID 255 is reserved for logical system files for Software AG products, see profile parameter LFILE .
		0	DBID=0 sets FSEC inactive. This is mandatory for a non-security environment.
	<i>file-number</i>	1-65535	File number for the Natural Security system file.
		0	FNR=0 sets FSEC inactive. This is mandatory for a non-security environment.
	<i>password</i>	1 to 8 characters	The password is only required if the Natural Security system file has been password-protected using the Adabas security feature.
	<i>cipher-key</i>	8 numeric digits	The cipher key is only required if the Natural Security system file has been ciphered using the Adabas security feature.
	R0		Read-only option. R0 indicates that the system file is “read-only” and is only specified if modifications on the file are to be disabled.
Default setting	none		
Dynamic specification	yes		If you specify the FSEC parameter dynamically in conjunction with any of the parameters DBID , FNR , SYSPSW , SYSCIP or ROSY , you must specify the FSEC parameter <i>after</i> any of these other parameters.
Specification within session	no		

The syntax of this parameter is:

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

Example: FSEC=(10,8)



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

113

FSP00L - Natural Advanced Facilities Spool File

This Natural profile parameter only applies to Natural Advanced Facilities.

It defines five subparameters for the Natural Advanced Facilities spool file. The spool file is the database file that is used by Natural Advanced Facilities. This file must be different from the [FNAT](#), [FUSER](#), [FDIC](#) and [FSEC](#) system files.

Possible settings	<i>database-ID</i>	1 - 65535, except 255	Database ID 255 is reserved for logical system files for Software AG products, see profile parameter LFILE . If any component of the FSP00L setting is not specified, the corresponding setting of the parameter DBID , FNR , SYSPSW or SYSCIP applies for the spool file.
	<i>file-number</i>	1 - 65535	Database file number.
	<i>password</i>	1 - 8 characters	The password is only required if the spool file has been password-protected using the Adabas security feature. Note for Natural with VSAM System Files: The password is used to specify the logical name (DD or DLBL) of the system file as defined to VSAM. Example: FSP00L=(10,8,SYSVSAM) For further information, see <i>Using Natural with VSAM System Files</i> in the <i>Natural for VSAM</i> documentation.
	<i>cipher-key</i>	8 characters	The cipher key is only required if spool file has been ciphered using the Adabas security feature.
	RO		Read-only option. RO indicates that the Natural Advanced Facilities spool file is “read-only” and is only specified if modifications on the file are to be disabled. This would mean, for example, that no reports could be stored on the spool file.
Default setting	none		

Dynamic specification	yes	If you specify the FSP00L parameter dynamically in conjunction with any of the parameters DBID , FNR , SYSPSW or SYSCIP , you must specify the FSP00L parameter <i>after</i> any of these parameters.
Specification within session	no	

The syntax for this parameter is:

FSP00L=(*database-ID,file-number,password,cipher-key,RO*)

Example:

FSP00L=(10,8)

114

FUSER - Natural System File for User Programs

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

Possible settings	<i>database-ID</i>	1- 65535, except 255	Database ID 255 is reserved for logical system files for Software AG products, see profile parameter LFILE .
	<i>file-number</i>	1-65535	File number.
	<i>password</i>	1 to 8 characters	The password is only required if the Natural user-program system file has been password-protected using the Adabas security feature. Note for Natural with VSAM System Files: The password is used to specify the logical name (DD or DLBL) of the system file as defined to VSAM. Example: FUSER=(22 , 5 , SYSVSAM) For further information, see <i>Using Natural with VSAM System Files</i> in the <i>Natural for VSAM</i> documentation.
	<i>cipher-key</i>	8 numeric digits	The cipher key is only required if the Natural user-program system file has been ciphered using the Adabas security feature.
	R0		Read-only option. R0 indicates that the Natural user-program system file is “read-only” and is only specified if modifications on the file are to be disabled.
Default setting	none		
Dynamic specification	yes		If you specify the FUSER parameter dynamically in conjunction with any of the parameters DBID , FNR , SYSPSW , SYSCIP or ROSY , you must specify the FUSER parameter <i>after</i> any of these parameters.
Specification within session	no		

The syntax of this parameter is:

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

Examples:

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



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

115

GC - Filler Character for Group Headers

With this session parameter, you specify the filler character which will appear on either side of a group heading produced by a `DISPLAY` statement across all field columns that belong to that group.

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

Possible settings	any character	Filler character for group headers.	
Default setting	blank		
Specification within session	yes	Applicable Statements:	<code>DISPLAY</code> <code>FORMAT</code>
		Applicable Command:	None

Example:

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


116

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)
```


117

HCAM - Hardcopy Access Method

This Natural profile parameter determines which access method is to be used for hardcopy output processing.

HCAM=xxx is equivalent to the AM subparameter of the profile parameter [PRINT](#) for Print File 0, that is, PRINT=((0),AM=xxx).

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

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

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

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

118

HCDEST - Hardcopy Output Destination

This Natural profile parameter presets the hardcopy output destination for the terminal command %H (without the *destination* operand).

HCDEST=xxx is equivalent to the DEST subparameter of the profile parameter PRINT for Printer 0, that is, PRINT=((0), DEST=xxx).

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

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

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

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

119

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.

This parameter can be specified

- at field/element level in a `DEFINE DATA` statement; see the sections *View Definition* and *EM, HD, PM Parameters for Field/Variable*;
- in the `Miscellaneous` field of the Data Area Editor (see *Columns in the Editing Area*);
- in the `SYSDDM` utility (see *Specifying Extended Field Attributes*).

Possible settings	'text'	120 alphanumeric or Unicode characters at maximum.
Default setting	none	
Applicable Statement:	DEFINE DATA	Parameter may be specified at field/element level.
Applicable Command:	none	

120

HE - Helproutine

■ HE Parameter Syntax	306
■ Execution of Helproutines	308
■ Examples	308

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 (question mark (?) by default) and press ENTER.
- 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

[

,

{

=

nX

}

]

...20

Operand Definition Table:

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

Syntax Description:

<i>operand1</i>	<p><i>operand1</i> 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 case of the specified name is not translated. 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.</p> <p>For additional information on using <i>operand1</i> within a map, see the HE helproutine option described in <i>Extended Field Editing in Map Editor</i> in the <i>Editors</i> documentation.</p>
<i>operand2</i>	<p>You may specify 1 to 20 parameters (<i>operand2</i>) 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.</p> <p>For additional information on using <i>operand2</i> within a map, see the HE helproutine option described in <i>Extended Field Editing in Map Editor</i> in the <i>Editors</i> documentation.</p>
=	<p>The equals sign (=) is used to pass an object or a field name to a helproutine or help map:</p> <ul style="list-style-type: none"> ■ 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. <p>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.</p>
<i>nX</i>	<p>The notation <i>nX</i> 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.</p>

**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
...

```

121

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	<p>The character which is to be used to invoke a field-specific helproutine or a map helproutine.</p> <p>The character specified with the profile parameter HI must not be the same as the one specified with the profile/session parameter CF (control character for mainframe terminal commands); it should not be the same as the one specified with the profile/session parameter DC (decimal character), profile/session parameter IA (input assign character) or profile/session parameter ID (input delimiter character).</p>
	blank	<p>Numeric fields which have a helproutine assigned are internally translated to alphanumeric format so as to make it possible for the user to enter a question mark into the field to invoke the helproutine.</p> <p>To prevent this internal translation (that is, if you wish to make sure that alphabetical characters cannot be entered into a numeric field) you can set the profile parameter HI to blank.</p> <p>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.</p>
Default setting	?	Question mark.
Dynamic specification	yes	
Specification within session	no	
Application Programming Interface	USR0350N	See <i>SYSEXT - Natural Application Programming Interfaces in the Utilities</i> documentation.

122

HW - Heading Width

With this session parameter you determine the width of a column output with a `DISPLAY` statement.

Possible settings	ON	The width of a DISPLAY column is determined by either the length of the heading text or the length of the field, whichever is longer. This is true even if no heading text is output, either because the DISPLAY statement contains the keyword NOHDR or the DISPLAY statement is a subsequent DISPLAY (see also the DISPLAY statement).	
	OFF	The width of a DISPLAY column is determined by the length of the field. HW=OFF only applies to DISPLAY statements which do not create headers (that is, either a first DISPLAY statement with NOHDR option or a subsequent DISPLAY statement).	
Default setting	ON		
Specification within session	yes	Applicable Statements:	DISPLAY FORMAT
		Applicable Command:	none

Example:

```
DISPLAY (HW=OFF)
```


123

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.

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

Possible settings	any special character	Assignment character for the input parameter processing in <code>INPUT</code> statements. The character specified with the <code>IA</code> parameter <ul style="list-style-type: none"> ■ must not be the same as the character specified with the profile/session parameters <code>CF</code> (control character for mainframe terminal commands), <code>DC</code> (decimal character) or <code>ID</code> (input delimiter character); ■ should not be the same as the one specified with the profile parameter <code>HI</code> (help character). 	
Default setting	=	Equals sign.	
Dynamic specification	yes		
Specification within session	yes	Applicable Statements:	SET GLOBALS
		Applicable Command:	GLOBALS
Application Programming Interface	USR0350N, USR1005N *	See <i>SYSEXT - Natural Application Programming Interfaces in the Utilities</i> documentation. * Recommended.	

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

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

124

IC - Insertion Character

With this session parameter, you specify the character string to be inserted in the column immediately preceding the value of a field output with a `DISPLAY` statement. The width of the output column is enlarged accordingly.

For numeric values, the insertion characters will be placed before the first significant digit printed.

The `IC` and `LC` parameters are mutually exclusive.

The parameter `IC` can also be used with `U` format fields.

Possible settings	any character	Character string to be inserted. You can specify a string of one to ten characters.	
		Insertion characters may optionally be specified within apostrophes, in which case any characters can be specified. Any character string specified which contains a closing parenthesis or a quotation mark must be enclosed within apostrophes. A blank in a character string not enclosed within apostrophes is represented by a circumflex (^).	
Default setting	none		
Specification within session	yes	Applicable Statements:	DISPLAY FORMAT
		Applicable Command:	none

Examples:

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

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

125

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.

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

Possible settings	any special character	Input delimiter character. The character specified with this parameter <ul style="list-style-type: none"> ■ must not be the same as the one specified with the profile/session parameter DC (decimal character) or IA (input assign character); ■ should not be the same as the one specified with the CF parameter (control character for mainframe terminal commands) or HI parameter (help character). The period (.) should not be used as input delimiter, because this might lead to situations in which a program termination period would be misinterpreted as input delimiter. An asterisk (*) should not be used either.	
	blank	No input delimiter character is defined.	
Default setting	, (comma)	If the input delimiter character is to be a comma (,), it must be specified as ID= ' , ' when using the dynamic parameter facility, because the 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 - <i>Natural Application Programming Interfaces</i> in the Utilities documentation.	

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

126

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	The setting <code>ENTR</code> is placed in the Natural system variable <code>*PF-KEY</code> ; that is, Natural reacts as if <code>ENTER</code> had been pressed.
	OFF	A <code>REINPUT</code> message is generated, prompting the user to press a valid key.
Default setting	OFF	
Dynamic specification	yes	
Specification within session	no	

127

IM - Input Mode

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

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

Possible settings	F	Forms mode.	
	D	Delimiter mode.	
Default setting	F		
Dynamic specification	yes		
Specification within session	yes	Applicable Statements:	SET GLOBALS
		Applicable Command:	GLOBALS
Application Programming Interface	USR1005N	See <i>SYSEXT - Natural Application Programming Interfaces</i> in the <i>Utilities</i> documentation.	



Notes:

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

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

128

IMSG - Session Initialization Error Messages

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



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

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

129

INTENS - Printing of Intensified Fields

This Natural profile parameter specifies how many times an intensified field or the underline character is to be overprinted when it is printed on a print device.

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

130

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 <code>INPUT</code> 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 <code>INPUT</code> 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
```


131

IS - Identical Suppress

With this session parameter, you can suppress the printing of identical information in successive lines created by a `WRITE` or `DISPLAY` statement.

The `IS` parameter setting can be suspended for one record by issuing the `SUSPEND IDENTICAL SUPPRESS` statement.

The `IS` parameter may be used in combination with the parameters `ES` and `ZP` to cause empty line suppression.

Possible settings	ON	A value which is identical to the previous value for the field will not be displayed. If a <code>DISPLAY</code> or <code>WRITE</code> statement is used to create multiple output lines using the <code>VERT</code> or slash (/) notation, <code>IS=ON</code> applies only to the first line.	
	OFF	No automatic suppression will be used.	
Default setting	OFF		
Specification within session	yes	Applicable Statements:	<code>DISPLAY</code> <code>FORMAT</code> <code>WRITE</code>
		Applicable Command:	<code>none</code>

Example:

```
FORMAT IS=ON
```

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

132

ISIZE - Size of Initialization Buffer

This Natural profile parameter specifies the size of the Natural initialization buffer.

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

Possible settings	8 - 32	Buffer size in KB.
Default setting	12	
Dynamic specification	yes	
Specification within session	no	



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

133

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.

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

134

ITRACE - Internal Trace Function

This Natural profile parameter is used to activate/deactivate the internal trace function.

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

The internal trace function is intended primarily for Software AG internal use for debugging purposes.

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

135

KD - Key Definition

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

This information will always be displayed automatically in the two bottom lines of the physical screen with any output created by the `INPUT`, `WRITE`, `DISPLAY`, and `PRINT` statement.

As this display requires two lines, the logical page size (see the session parameter [PS](#)) must be reduced by two.

Possible settings	ON	The names assigned to the PF keys are displayed.	
	OFF	The names assigned to the PF keys are not displayed.	
Default setting	OFF		
Specification within session	yes	Applicable Statements:	FORMAT
		Applicable Command:	none

Example:

```
FORMAT KD=ON
```


136

KEY - Setting Assignments to PA, PF and CLEAR Keys

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

Possible settings	any character string	Settings can be assigned to the keys PA1 to PA3, PF1 to PF24 and to the CLEAR key. The setting assigned to each key can be any character string. The character string must represent a Natural system command or a user command (user program). If the setting contains embedded blanks, it must be enclosed in apostrophes.
Default setting	none	
Dynamic specification	no	
Specification within session	yes	
Application Programming Interface	USR4005N	See <i>SYSEXT - Natural Application Programming Interfaces</i> in the <i>Utilities</i> documentation.

Assignments made with the profile parameter **KEY** are only valid from the Natural **NEXT** prompt.

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

Examples:

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

137

LC - Lower to Upper Case Translation

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



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

Possible settings	ON	No translation of lower-case characters to upper case is performed.
	OFF	All lower-case characters, except input from the Natural stack which was place there by the <code>STACK</code> statement, is translated to upper case by Natural.
Default setting	OFF	
Dynamic specification	yes	
Specification within session	yes	To disable or enable lower-case to upper-case translation dynamically within the active Natural session, you should use the terminal commands <code>%L</code> or <code>%U</code>
Application Programming Interface	USR1005N	See <i>SYSEXT - Natural Application Programming Interfaces</i> in the <i>Utilities</i> documentation.



Notes:

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

138

LC - Leading Characters

With this session parameter, you can specify leading characters that are displayed immediately before a field output by a `DISPLAY` statement. The width of the output column is enlarged accordingly.

The session parameters `LC` and `IC` are mutually exclusive.

The parameter `LC` 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*.

Possible settings	any character	Up to 10 characters may be specified. Leading characters may optionally be specified enclosed within apostrophes, in which case, any characters can be specified. Any character string specified which contains a closing parenthesis or a quotation mark must be enclosed within apostrophes. 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:	DISPLAY FORMAT
		Applicable Command:	none

Example:

```
DISPLAY {LC=*
```

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

139

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. 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.

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

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. 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 - <i>Natural Application Programming Interfaces</i> in the <i>Utilities</i> documentation.	

The `LE` parameter applies to `READ`, `FIND` and `HISTOGRAM` statements with a limit specified.

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').

■ LFILE Parameter Syntax	353
■ NTLFILE Macro Syntax	353
■ Old NTFIL Macro Syntax	353
■ Example of LFILE Parameter	353
■ Example of NTLFILE Macro	354

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

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

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

Possible settings	<i>logical-FNR</i>	1 - 251	Logical file number (LFL). This parameter is mandatory.
	<i>physical-DBID</i>	0 - 65535, except 255	Physical database ID (DBID). Database ID 255 is reserved for logical system files for Software AG products.
	<i>physical-FNR</i>	1 - 65535	Physical file number (FNR).
	<i>password</i>		Must be a setting of 1 to 8 characters. *
	<i>cipher-key</i>		Must be a setting of 8 numeric digits. *
	<i>options</i>	R0	For read-only access.
Default setting	none		
Dynamic specification	yes	This parameter can only be specified dynamically. In the Natural parameter module NATPARM, the macro NTLFILE must be used instead. It replaces the old macro NTFIL E for logical system file definition which is still available, but should not be used any longer.	
Specification within session	no		
Application Programming Interface	USR0011N	See SYSEXT - Natural Application Programming Interfaces in the Utilities documentation.	
	USR2004N *		

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



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

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

The following topics are covered below:

LFILE Parameter Syntax

The LFILE parameter is specified as follows:

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

NTLFILE Macro Syntax

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

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

Old NTFIL Macro Syntax

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

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

Example of LFILE Parameter

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

Example of NTLFILE Macro

Equivalent specification in the Natural parameter module:

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


141

LIBNAM - Name of External Program Load Library

This Natural profile parameter only applies under BS2000/OSD, z/OS batch mode, and TSO.

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

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

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

142

LOG (Internal Use)

This parameter is reserved for internal use by Natural.



Caution: Do not change its setting.

143

LOGONRQ - Logon for RPC Server Request Required

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

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

LOGONRQ is specified on the server side only.

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

For Natural clients the logon data can either be requested:

- by setting the LOGON option of the SYSRPC Service Directory Maintenance;
- using the logon indicator of the profile parameter [DFS](#).

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

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

144

LS - Line Size

■ Profile Parameter LS	362
■ Session Parameter LS	362
■ Specification with Statements	363

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.
Default setting	0	Physical line size.
Applicable command:	GLOBALS	
Applicable statements:	FORMAT SET GLOBALS	
Application Programming Interface	USR1005N	See <i>SYSEXT - Natural Application Programming Interfaces</i> in the <i>Utilities</i> documentation.



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

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.
-------------------------------	---------------------------	--

145

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. This limit 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`.

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`.

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

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.

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.

Possible settings	0 - 2147483647 Within a session: 0 to <i>n</i> (<i>n</i> = value of profile parameter <code>LT</code> at session start)	Maximum number of records that can be retrieved. All retrieved records (including records rejected by means of a <code>WHERE</code> clause) are counted and compared with this limit. <code>LT=0</code> defines that no limit is in effect for the number of retrieved records.	
Default setting	99999999		
Dynamic specification	yes		
Specification within session	yes	Applicable Statements:	<code>SET GLOBALS</code> , see Note.
		Applicable Command:	<code>GLOBALS</code> , see Note.
Application Programming Interface	USR1005N	See <i>SYSEXT - Natural Application Programming Interfaces</i> in the <i>Utilities</i> documentation.	



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 module `NATPARM`.

146

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 <i>SYSEXT - Natural Application Programming Interfaces</i> in the <i>Utilities</i> documentation.
	USR1068N *	
		* Recommended.

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

147

MAINPR - Override Default Output Report Number

This Natural profile parameter can be used to separate program output from Natural system output, which may be useful particularly in batch mode.

Possible settings	0 - 31	Valid printer number.
Default setting	0	
Dynamic specification	yes	
Specification within session	no	
Application Programming Interface	USR6002N	See <i>SYSEXT - Natural Application Programming Interfaces</i> in the <i>Utilities</i> documentation.

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

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

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

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

148

MAXBUFF - Maximum Buffer Size

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

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

On the server side, it determines the size of the buffer provided by the server to receive the client request including data and to send back the result. The buffer must be large enough to hold the largest of the following two data areas for all client requests:

- the request received by the client,
- the result send 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.

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

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

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 *Stubs and Automatic RPC Execution in the Natural Remote Procedure Call (RPC) documentation*.

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

Possible settings	1 - 2097147, but smaller than or equal to <code>RPCSIZE - 4</code>	Maximum buffer size in KB. The maximum buffer size must be equal to or less than the value (minus 4) specified with the profile parameter <code>RPCSIZE</code> (for the server side, see below).
	0	No buffer is allocated.
Default setting	0	
Dynamic specification	yes	
Specification within session	no	

Dependency on Number of Parameters on Server Side

On the server side, the difference between `RPCSIZE` and `MAXBUFF` depends on the maximum number of parameters n in the PDA and can be calculated as follows:

- If group structures are present:

$$\text{MAXBUFF} = \text{RPCSIZE} - (3 + n/10)$$

- If no group structures are present:

$$\text{MAXBUFF} = \text{RPCSIZE} - (3 + n/20)$$

Example:

If $n=100$ and `RPCSIZE=128`, then `MAXBUFF=120`.

Dependency on `ACIVERS` Settings

In case of an EntireX Broker node, special considerations apply if you are using Entire Net-Work as a transport layer. With Entire Net-Work, the receive buffer length passed to the EntireX Broker stub is restricted by the startup parameter `IUBL` and must not exceed 32 KB. Depending on the setting of the Natural profile parameter `ACIVERS`, the receive buffer length is set as follows:

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

For further information, see the *Natural Remote Procedure Call (RPC)* documentation.

149

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.

If the specified limit is exceeded, the Natural program is interrupted and the user is notified with an appropriate Natural error message (NAT1029).

Possible settings	10 - 32767	Maximum number of program calls.
	0	MAXCL=0 indicates that no limit is to be in effect.
Default setting	50	
Dynamic specification	yes	
Specification within session	no	
Application Programming Interface	USR1005N	See <i>SYSEXT - Natural Application Programming Interfaces</i> in the <i>Utilities</i> documentation.
	USR1068N *	
		* Recommended.

150

MAXROLL - Number of CMROLL Calls before Session

Suspension

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

It specifies the number of CMROLL calls after which a Natural session is suspended, that is, a potential roll-out of the Natural thread is to be performed.

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

The MAXROLL parameter can be used to control the frequency of conditional CMROLL requests. For example, MAXROLL=128 means that a conditional CMROLL request is issued after every 128th statement at compilation.

In certain cases, the Natural nucleus issues a conditional CMROLL request (wait time = 0), particularly at compilation after each statement. This is done to reset the CPU time window (under Com-plete) in order to avoid an automatic cancel due to the CPU time limit being exceeded; however, this has a negative impact on performance.

Note Concerning CMROLL

Calling CMROLL is the Natural interface for WAIT or DELAY functionality (see also sample Natural program SUSPEND in library SYSEXTP); when calling CMROLL, you may pass a delay interval/wait time as parameter. When a session has to wait in CMROLL, shared resources as a thread in Com-plete or a shared thread in CICS (THREADS=nonzero) are released, and as a consequence a potential roll-out of the Natural thread is performed. Calling CMROLL with a delay interval of 0 is called conditional, as the session actually needs not wait for a certain time; however, when other sessions

are waiting for a thread, the session is suspended, which may result in a roll-out of the Natural thread. In CICS if no other session is waiting, just an EXEC CICS SUSPEND is executed to prevent AICA abends.

151

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	

MAXYEAR=9999 changes the maximum date value that can be entered from 2699-12-31 to 9999-12-31.



Note: 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



Caution: Natural applications that were cataloged with the Natural Optimizer Compiler using the `OVFLW=ON` option set (See *Optimizer Options* for more information) have to be recataloged with Natural Optimizer Compiler Version 4.2.2 or above and must be executed with Natural Version 4.2.2 or above to ensure that the MAXYEAR profile parameter setting is correctly applied. If an application that uses date/time fields has been recataloged with Natural Optimizer Version 4.2.2 or above, and is afterwards executed with Natural Version 4.2.1, an ABEND will occur at runtime.

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
- Profile parameter [YD](#) in the *Parameter Reference* documentation

152

MC - Multiple-Value Field Count



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

With this session parameter, you determine the number of values of a multiple-value field to be output by default when the field is specified without an index in a `DISPLAY` or `WRITE` statement.

Possible settings	0 - 191	Number of values. If MC=0 is specified, then there is no default index range for the output of an MU field. Therefore, when an MU field is output, it is necessary to specify an explicit index or index range. Otherwise, a syntax error (NAT0281) will be raised.	
Default setting	1		
Specification within session	yes	Applicable Statements:	DISPLAY FORMAT INPUT PRINT WRITE
		Applicable Command:	None

Example:

```
FORMAT MC=5
```

153

MENU - Menu Mode

This Natural profile parameter is used to enable or disable Natural menu mode.

Possible settings	ON	Menu mode is enabled.
	OFF	Menu mode is disabled.
Default setting	ON	
Dynamic specification	yes	
Specification within session	yes	Within a Natural session, the MENU parameter can be overridden by the Natural system command MAINMENU (described in the System Command documentation).



Note: In asynchronous Natural sessions under the TP monitors Com-plete, CICS and UTM, the setting of this parameter is forced to OFF to avoid problems when changing over to NEXT mode.

154

ML - Position of Message Line

This profile 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. For information on the operand 'M', see also Natural terminal command %M (Control of Message Line).

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

Possible settings	B	Natural messages are displayed at the bottom of the screen.	
	T	Natural messages are displayed at the top of the screen.	
Default setting	T		
Dynamic specification	yes		
Specification within session	yes	Applicable Statements:	SET CONTROL 'M'
		Applicable Command:	
Application Programming Interface	USR1005N	See <i>SYSEXT - Natural Application Programming Interfaces</i> in the <i>Utilities</i> documentation.	

155

MONSIZE - Size of SYSTP Monitor Buffer

This Natural profile parameter specifies the size of the buffer used by the Monitor function of the SYSTP utility (described in the *Utilities* documentation).

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

Possible settings	5 - 256	Buffer size in KB.
	0	If MONSIZE=0 or if the requested space is not available, the Monitor function of the SYSTP utility cannot be used, except there is a monitor buffer pool defined by means of profile parameter BPI or parameter macro NTBPI.
Default setting	0	
Dynamic specification	yes	
Specification within session	no	

156

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.

Within a Natural session, the setting of profile parameter `MP` can be reduced, but not increased by the `FORMAT` statement. The value specified with the session parameter `MP` applies only to the specified report.

Possible settings	1 - 99999	The value specified is the number of physical pages and has no effect on the starting page number used. The program will be terminated with an error message if the MP value is exceeded.	
	0	No page limit is defined.	
Default setting	0		
Dynamic specification	yes		
Specification within session	no	Applicable Statements:	DISPLAY FORMAT PRINT WRITE
		Applicable Command:	

157

MS - Manual Skip

With this session parameter, you control the cursor positioning during the processing of an INPUT statement.

Possible settings	ON	See example below. Note: The setting MS=ON is not supported under BS2000/OSD.	
	OFF	The cursor will be positioned to the next input field as soon as the value for the current field is entered with all positions.	
Default setting	OFF		
Specification within session	yes	Applicable Statements:	FORMAT INPUT
		Applicable Command:	none

Example:

```
INPUT (MS=ON) #A #B
```

158

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	Within a Natural session, the profile parameter MSGSF can be overridden by the Natural terminal command %MSGSF.

By default, a Natural system error message consists of the following:

- the name of the program,
- the number of the line that caused the error,
- the actual text of the message.

Depending on the size of the window in which the message is displayed, the text may be truncated. With this parameter, you can avoid such truncation.

159

MT - Maximum CPU Time

This Natural profile and session parameter only applies to programs executed in batch mode, under Natural Development Server (SPoD) or under Natural for TSO.

It determines the maximum amount of CPU time which can be used by a Natural program.

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.

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

Possible settings	1 - 9999999	Maximum amount of CPU time in seconds. If Natural Security is installed, the profile parameter MT can be overridden within Natural Security. With Natural Security, the maximum value for the profile parameter MT is 32767. To use a higher value as specified with the MT profile or session parameter, specify MT=0 within Natural Security.	
	0	MT=0 defines that no Natural CPU time limit is to be in effect.	
Default setting	60		
Dynamic specification	yes		
Specification within session	yes	Applicable Statements:	SET GLOBALS
		Applicable Command:	GLOBALS
Application Programming Interface	USR1005N	See SYSEXT - <i>Natural Application Programming Interfaces</i> in the Utilities documentation.	



Notes:

1. The limit for programs operating in interactive mode is controlled by the TP monitor in use.

2. 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.
3. 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.



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 (NDV) servers.

160

NAFSIZE - Size of Buffer for Natural Advanced Facilities

This Natural profile parameter only applies if Natural Advanced Facilities is installed.

It specifies the size of the work buffer used by Natural Advanced Facilities.

Alternatively, you can use the equivalent Natural profile parameter [DS](#) or macro [NTDS](#) (see *Using Optional Macros in a Natural Parameter Module in the Operations* documentation) to specify `NAFSIZE`.

Possible settings	1 - 64	Buffer size in KB.
	0	<code>NAFSIZE=0</code> disables Natural Advanced Facilities.
Default setting	0	
Dynamic specification	yes	
Specification within session	no	

If Natural Advanced Facilities is to be used, a setting has to be specified for this parameter; see *NATSPool Initialization* in the *Natural Advanced Facilities* documentation.

If the requested space is not available, Natural Advanced Facilities cannot be used.

161

NAFUPF - Natural Advanced Facilities User Profile

This Natural profile parameter only applies if Natural Advanced Facilities is installed.

It is used to specify the user-profile name for Natural Advanced Facilities.

Possible settings	1 to 8 characters	Name of the user profile.
Default setting	none	
Dynamic specification	yes	
Specification within session	no	

See *NATSPool Initialization* in the *Natural Advanced Facilities* documentation.

162

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.

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

Possible settings	ON	System commands cannot be used - except FIN, LAST, LOGOFF, LOGON, MAINMENU, RENUMBER, RETURN, SETUP and TECH. If you have Natural Security installed, any system command restrictions you set with Natural Security are valid, regardless of the setting of the NC profile parameter. In a Natural Development Server environment on mainframe computers, the value OFF will be assumed for the Natural Development Server, even if NC=ON has been specified. If NC=ON has been specified on the client side, subsequent system commands issued on the client side will be rejected as described above.	
	OFF	All system commands can be used.	
Default setting	OFF		
Dynamic specification	yes		
Specification within session	yes	Applicable Statements:	SET GLOBALS
		Applicable Command:	
Application Programming Interface	USR1005N	See SYSEXT - <i>Natural Application Programming Interfaces in the Utilities</i> documentation.	



Note: Natural terminal commands and user-created commands (object module names) are not affected by the NC parameter.

163

NISN (Internal Use)

This parameter is reserved for internal use by Natural.



Caution: Do not change its setting.

164

NL - Numeric Length for Output

This session parameter determines the default input/output length for a numeric field used in a `DISPLAY`, `INPUT`, `PRINT` or `WRITE` statement.

The `NL` parameter must not be specified for groups.

Any edit mask specified for a field will override the `NL` parameter for this field.

Possible settings	<i>nn.m</i>	<p>The length is specified as <i>nn.m</i>, where <i>nn</i> represents the number of positions before the decimal point and <i>m</i> represents the number of positions after the decimal point.</p> <p>The <i>m</i> notation is optional. The value of <i>m</i> must not exceed 7. The total of <i>n+m</i> must not exceed 29.</p> <p>Note:</p> <ol style="list-style-type: none"> 1. If <code>NL</code> is set less than the field length, values are truncated. No error is produced when relevant digits are truncated. 2. If <code>NL</code> is set greater than the field length, values are expanded with blanks. No error is produced when an input field is truncated. 	
Default setting	none		
Specification within session	yes	Applicable Statements:	DISPLAY FORMAT INPUT PRINT WRITE
		Applicable Command:	none

Example:

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

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

165

NTASKS - Number of Server Tasks to be Started

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

`NTASKS=(min[,max])` specifies the minimum number *min* of server tasks to be started during server initialization and the maximum number *max* of server tasks that may be active at any time. If only *min* is specified, the parentheses may be omitted.

If the server has to handle a large number of client requests, you can use this subparameter to improve the throughput by starting multiple (identically named) replicas of the same server task.

`NTASKS` applies only to servers started in batch mode under z/OS or VSE and to servers started by an RPC server front-end.

The maximum number *max* of server tasks applies only to servers started by an RPC server front-end.

Possible values:	<i>min</i>	Minimum number of server tasks to be started during server initialization: 1 - 99.
	<i>max</i>	Maximum number of server tasks that may be active at any time: 0 - <i>n</i> 0 = unlimited.
Default value:	1 , 0	
Dynamic specification	yes	
Specification within session	no	

For further information, see the *Natural Remote Procedure Call (RPC)* documentation and especially *Considerations for Mainframe Natural RPC Servers with Replicas*.

166

NUCNAME - Name of Shared Nucleus

This Natural profile parameter specifies the name of the (environment-independent) Natural shared nucleus if it is to be loaded dynamically and not linked to the environment-dependent Natural nucleus.



Note: The profile parameter `NUCNAME` does not apply under BS2000/OSD.

See also *Natural Shared Nucleus under z/OS and z/VSE* in the *Operations* documentation.

Possible settings	1 to 8 characters	Valid load module name.
Default setting	none	
Dynamic specification	yes	By specifying this parameter dynamically, you are able to use different Natural shared nuclei (for example, for production and for testing) together with the same environment-dependent Natural nucleus without having to relink the nucleus.
Specification within session	no	

The profile parameter `NUCNAME` is ignored if it is specified in a parameter string activated by a [SYS](#) or [PROFILE](#) profile parameter or in an alternative parameter module (as specified with the [PARM](#) profile parameter).

167

OBJIN - Use of CMOBJIN as Natural Input File

This Natural profile parameter only applies to batch mode.

It indicates whether the CMOBJIN file, see *Natural in Batch Mode* in the *Operations* documentation is to be used for input data provided with the INPUT statement in batch mode.

Possible settings	Y	Data for a Natural INPUT statement are read from the CMOBJIN file.
	N	The CMOBJIN file is not used and any data for an INPUT statement are read from the CMSYNIN file.
	R	Natural determines which option has been selected for a particular session by the presence or absence of the CMOBJIN DD/FILE statement in the Natural execution JCL/JCS.
Default setting	R	
Dynamic specification	yes	
Specification within session	yes	

168

OPF - Overwriting of Protected Fields by Help routines

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

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

Possible settings	ON	A helproutine assigned to a field can overwrite the field's content, even if the field is write-protected.	
	OFF	Helproutines cannot overwrite the contents of write-protected fields.	
Default setting	ON		
Dynamic specification	yes		
Specification within session	yes	Applicable Statements:	SET GLOBALS
		Applicable Command:	GLOBALS
Application Programming Interface	USR1005N	See <i>SYSEXT - Natural Application Programming Interfaces</i> in the Utilities documentation.	



Notes:

1. The **OPF** profile parameter only applies to the field for which a help routine is invoked; it does not affect parameters explicitly passed to the help routine. 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 help routine.
2. In addition, in reporting mode you can change the **OPF** setting using the statement **SET GLOBALS**.

■ Dynamic OPRB with Natural Security	415
■ OPRB for VSAM	415
■ OPRB for Adabas	415
■ NTOPRB Macro Syntax	417
■ Examples of NTOPRB Macros	417

This Natural profile parameter only applies to Adabas and VSAM databases.

It controls the use of database `Open/Close` commands during a Natural session.

The `NTOPRB` macro can be used as an alternative to the profile parameter `OPRB` in the `NTPRM` macro. The maximum length of an `OPRB` parameter specification is 256 bytes. If you require a longer specification, use the `NTOPRB` macro instead of the `OPRB` parameter.

If you wish to make `OPRB` specifications that are to apply to all databases, it is strongly recommended that you use the `OPRB` parameter in the `NTPRM` macro (and not an `NTOPRB` macro).

Possible settings	<code>OPRB=(string)</code>	With this syntax, you specify an Open request for <i>all</i> databases.
	<code>OPRB=(DBID=nn1, string, DBID=nn2, string, ...)</code>	With this syntax, you specify an Open request for specific individual databases. As defined in the macro <code>NTDB</code> , the specified DBID identifies the type of database.
	<code>OPRB=(string, DBID=nn1, string, DBID=nn2, string, ...)</code>	With this syntax, you specify an Open request for specific individual databases and also a default Open request - the initial <i>string</i> - which applies to all databases for which you do not specify an individual <i>string</i> .
	<code>OPRB=(DBID=nn1, NR=ON/OFF, string, ...)</code>	With this syntax, you specify whether Natural is to issue a restricted or a non-restricted Open request when an Adabas database is accessed for the first time. This is to enhance control for Adabas files for which exclusive update (EXU) usage is desired.
Default setting	none	
Dynamic specification	no	
Specification within session	no	

Generally, the `OPRB` parameter uses one of the above syntaxes (the possible contents of the *strings* depend on the database system).

Instead of using the `OPRB` parameter, you can also use the macro `NTOPRB` in the Natural parameter module `NATPARM`.

The following topics are covered below:

- [Dynamic OPRB with Natural Security](#)
- [OPRB for VSAM](#)
- [OPRB for Adabas](#)
- [NTOPRB Macro Syntax](#)
- [Examples of NTOPRB Macros](#)

Dynamic OPRB with Natural Security

A dynamically specified `OPRB` parameter applies for all logons to libraries in whose security profiles no `OPRB` parameter is specified. For a logon to a library in whose security profile the `OPRB` parameter is specified, any dynamically specified `OPRB` parameter is ignored and the one from the security profile applies.

OPRB for VSAM

The *strings* which can be specified for VSAM databases are described under *The OPRB Parameter for VSAM Databases* in the *Natural for VSAM* documentation.

OPRB for Adabas

For Adabas databases, the `OPRB` parameter is required if either of the following conditions are true for the Natural session:

- An explicit list of Adabas files to be accessed/updated is to be provided. This is necessary, for example, if Adabas cluster updating or exclusive file control is to be requested.
- A single logical transaction is to span two or more Natural programs and, therefore, it is not desired to have Natural issue an `END TRANSACTION` and `CLOSE` command at the termination of any given Natural program.

Possible content of the parameter string:

ACC=(<i>file-list</i>)	Specifies access permission (read) for the files in the file list.
UPD=(<i>file-list</i>)	Specifies access/update permission (read/write) for the files in the file list.
EXF=(<i>file-list</i>)	Specifies exclusive file control: no other users may access/update the file.
EXU=(<i>file-list</i>)	Specifies exclusive update permission (exclusive read/write) for the files in the file list.
ACODE	Specifies the option to enforce a user encoding for A fields. Note: The required encoding code for ACODE is derived from the current CP parameter setting of the Natural session.
WCODE	Specifies the option to enforce a user encoding for W fields. Note: The required encoding code for WCODE is always 4095.
ARC	Defines a special data architecture for fields in the record and value buffers. This definition overrides the architecture key defined for remote calls in Entire Net-work.

For further information on these settings, refer to the description of the Adabas OP command in the *Adabas Command Reference* documentation.

If the OPRB parameter is omitted in the NATPARM module or OPRB=OFF is specified as a dynamic parameter, a Natural session commences with an Adabas Open command requesting UPD (access/update) to the Natural system file. Natural also issues RELEASE CID (Adabas RC) commands to release all ISN lists (ISN lists specified in a RETAIN clause of a Natural FIND statement are not released).

The Adabas record buffer to be used with the initial Adabas OP command can be explicitly provided. The format is similar to that used in an Adabas record buffer for the OP command with the exception that no blanks can be embedded, and the complete setting must be enclosed in parentheses (not apostrophes).

Example 1:

```
OPRB=(ACC=2,4,6,UPD=8.)
```

This specifies that Adabas Files 2,4 and 6 are to be made available for access only and that Adabas File 8 is to be made available for update (which also implies access).

Example 2:

```
OPRB=(EXU=1,2,3.)
```

This specifies that Adabas Files 1,2 and 3 are to be placed under exclusive control for this Natural session.

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

In all of the above situations, the `OP` command, which is always issued at the start of a Natural session, contains in the Additions 1 field of the Adabas control block the user ID for the Natural session. In batch mode, this is the job name. In TP mode, this is the setting supplied at system initialization by the Natural interface module. In both cases, the setting used is available in the Natural system variable `*INIT-USER`.

NTOPRB Macro Syntax

The syntax of the `NTOPRB` macro is as follows:

```
NTOPRB dbid,'string'
```

For possible values, see the [OPRB](#) parameter; if you use Natural with VSAM, see also the *Natural for VSAM* documentation.

If *string* is very long, it can be divided in up to five strings separated by commas (see below), as the Assembler allows single strings up to 256 bytes only.

Examples of NTOPRB Macros

```
NTOPRB 12,'ACC=40,UPD=20'
```

```
NTOPRB 15,'EXU=1','2,3'
```

170

OPT - Control of Natural Optimizer Compiler

■ OPT Parameter Syntax	420
■ NTOPT Macro Syntax	420

This Natural profile parameter only applies if the Natural Optimizer Compiler is to be used.

This parameter is used to activate/deactivate the Natural Optimizer Compiler and controls the various options related to it. It corresponds to the macro `NTOPT` in the Natural parameter module NATPARM.

Possible settings		See the <i>Dynamic Profile Parameter OPT</i> in the <i>Natural Optimizer Compiler</i> documentation.
Default setting	OFF	
Dynamic specification	yes	This parameter can only be specified dynamically. In the Natural parameter module NATPARM, the macro <code>NTOPT</code> must be used instead.
Specification within session	yes	

The following topics are covered below:

OPT Parameter Syntax

The parameter syntax of `OPT` is, for example, as follows:

```
OPT=(INDX,OVFLW,ZD=OFF)
```

For more syntax examples, refer to *Dynamic Profile Parameter OPT* in the *Natural Optimizer Compiler* documentation.

NTOPT Macro Syntax

The syntax of the `NTOPT` macro is, for example, as follows:

```
NTOPT 'INDX,OVFLW,ZD=OFF'
```

For more syntax examples, refer to *Macro NTOPT* in the *Natural Optimizer Compiler* documentation.

171

OUTDEST - Output Destination for Asynchronous Processing

This Natural profile parameter only applies to Natural under CICS, Com-plete and UTM.

It specifies the destination to which any Natural error message produced by an asynchronous application is to be sent.

Possible settings	1 to 8 characters.	Destination to which a Natural error message is sent. 1 to 8 characters.
Default setting	Setting of profile parameter SENDER	
Dynamic specification	yes	
Specification within session	no	

After an error message has been sent, Natural terminates the asynchronous session.

Under UTM, this parameter is used to specify the ID of the terminal where output from an asynchronous application is to be displayed.

When and how error messages/output from an asynchronous application are output depends on the respective TP monitor. For further information, see:

- *Asynchronous Natural Processing under CICS*
- *Asynchronous Natural Processing under Com-plete/SMARTS*
- *Asynchronous Transaction Processing under UTM*

172

OVSIZE - Storage Thread Overflow Size

This Natural profile parameter specifies the maximum total amount of variable storage that may be allocated by one Natural session outside its storage thread.

Possible settings	0-2097151	Maximum total storage outside the thread in KB.
Default setting	2097151	That is, the storage outside the thread is limited by the region size only.
Dynamic specification	yes	
Specification within session	no	

If the storage within the thread is exhausted during a Natural session, additional storage can be allocated outside of the thread. `OVSIZE` can be used to limit the total amount of variable storage. This does not affect physical storage (see profile parameter [WPSIZE](#)), which is allocated outside the thread always.

For non-thread environments (e.g. in batch mode or under TSO), this parameter is not honored.

173

PARM - Alternative Parameter Module

This Natural profile parameter specifies an object module containing profile parameter definitions.

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

These definitions are coded using the various macros as described under *Creating a New Natural Parameter Module* in the *Operations* documentation. The macros are then assembled, resulting in an object module whose name is specified by the user.

When the `PARM` parameter is specified (either in the linked parameter module or as a dynamic parameter at Natural session start), the appropriate object module is loaded and the profile parameter definitions contained therein take effect. The parameter module is loaded dynamically from the steplib.

Under CICS, a PPT entry is required for this parameter module.

Under BS2000/OSD, z/OS batch mode and TSO, the current steplib can be defined by profile parameter `LIBNAM`.

Any profile parameter definitions in effect before the `PARM` parameter is processed (for example, definitions contained in the linked parameter module or prior dynamic parameters), except the profile parameters `ISIZE` and `NUCNAME`, are overridden when the specified parameter module is loaded. Therefore, any dynamic parameters should be specified after the `PARM` specification.

The profile parameters `ISIZE` and `NUCNAME` are ignored if specified in an alternative parameter module.

To restrict the use of an alternative parameter module, you can use the macro `NTUSER` (described in the `USER` profile parameter description).

174

PC - Control of Personal-Computer Access Method

This Natural profile parameter only applies if Natural Connection is installed.

It determines whether support of the personal-computer access method is to be provided using Natural Connection.

Possible settings	ON	Personal-computer support is enabled. The Natural statements READ PC FILE or WRITE PC FILE can be used (for uploading or downloading); see UPLOAD PC FILE and DOWNLOAD PC FILE. With PC=ON, the system variable *DEVICE will always contain the value PC.
	OFF	No personal-computer support is to be provided.
	NAM	Field names are sent when data are uploaded/downloaded. This value is for mainframe environments only.
	NONAM	No field names are sent when data are uploaded/downloaded. This value is for mainframe environments only.
Default setting	(OFF , NAM)	
Dynamic specification	yes	
Specification within session	yes	The terminal commands %+ and %- can be used to control the PC support.

Multiple values are specified in a value list:

Example:

PC=(ON , NONAM)

See the Natural Connection documentation for further information.

The files used for the PC access method have to be defined with the macros `NTPRINT` and `NTWORK` or the profile parameters `PRINT`, `WORK` and `HCAM`.



Note: In asynchronous Natural sessions under the TP monitors Complete, CICS and UTM, the setting of this parameter is forced to OFF to avoid problems when changing over to NEXT mode.

175

PC - Periodic Group Count

This session parameter may be used in reporting mode only. It 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	0 - 191	Number of values. If <code>PC=0</code> is specified, then there is no default index range for the output of a PE field. Therefore, when a PE field is output, it is necessary to specify an explicit index or index range. Otherwise, a syntax error (NAT0281) will be raised
Default setting	1	
Applicable statements:	FORMAT	
	INPUT DISPLAY WRITE PRINT	Parameter may be specified at statement level and/or at element level.
Applicable command:	none	

Example:

```
FORMAT PC=5
```


176

PCNTRL - Print-Control Characters

This Natural profile parameter specifies the line-advance characters for printing which are inserted in Column 0 of each print line.

Possible settings	Any character string	This parameter can be specified in character or hexadecimal format.
Default setting	Siemens Environments	X '404142434445464748494A4B4C4D4E4F '
	IBM Environments	<p>The default setting (according to ASA standard settings) is:</p> <p>' 0 - '</p> <p>which means that a blank causes a line advance of 1 line, 0 of 2 lines and - of 3 lines.</p> <p>Caution: In any IBM environment, do not change the default setting of this parameter.</p>
Dynamic specification	yes	
Specification within session	no	

177

PD - Limit of Pages for NATPAGE

This Natural profile and session parameter specifies the maximum number of pages (screens) which can be stored at the same time in the Natural system file (FUSER) with the NATPAGE screen-paging utility.

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

Possible settings	0 - 255	Maximum number of pages (screens).	
Default setting	50		
Dynamic specification	yes		
Specification within session	yes	Applicable Statements:	SET GLOBALS
		Applicable Command:	GLOBALS



Notes:

1. If the number of stored screens exceeds the setting of PD, wrap-around technique is used for the system file, which means that the oldest page is overwritten.
2. For further information on the NATPAGE screen page utility, see the terminal commands %E, %I, %O, %P and %S.

178

PLOG - Logging of Dynamic Parameters

This Natural profile parameter only applies in batch mode, under TSO and under CICS.

It enables you to print a list of all profile parameters that were specified dynamically at the start of the session. This may be useful to ascertain which dynamic profile parameters were actually used, particularly if profile parameters like [PROFILE](#) or [SYS](#) are specified, which in turn “contain” other profile parameters (for a [PROFILE](#) or [SYS](#) parameter, the entire string of profile parameters activated by it is listed).

Possible settings	ON	In batch mode:	At session start, a list of the dynamically specified profile parameters and their settings is written to the output dataset CMPLOG. (If CMPLOG is not available, the list is written to the standard output dataset CMPRINT.)
		In online mode under TSO:	At session start, a list of the dynamically specified profile parameters and their settings is written to the output dataset CMPLOG. (If CMPLOG is not available, the list is sent to the terminal.)
		In online mode under CICS:	At session start, a list of the dynamically specified profile parameters and their settings is sent to the terminal.
	OFF	No list of dynamic profile parameters is written.	
Default setting	OFF		
Dynamic specification	yes	When specified dynamically, the PLOG parameter applies to all subsequent dynamic profile parameters until the next PLOG specification. This allows you to exclude individual parameters from being printed, for example, if their settings contain passwords or other sensitive information that should not be printed. All dynamic parameters which are specified between a PLOG=OFF specification and a subsequent PLOG=ON specification are not printed.	
Specification within session	no		

179

PLUGIN - Enable the Natural Plug-In Components

This Natural profile parameter is used to enable new or additional Natural components without having to link new components to your Natural nucleus or to apply specific fixes.

Possible settings	OFF	Deactivates all plug-in components, see note below.
Default setting	OFF	
Dynamic specification	yes	
Specification within session	no	



Note: Because all components of Natural Version 3.1 that could be enabled using the profile parameter `PLUGIN` are an integral part as of Natural Version 4.1, the only possible setting of this parameter is `OFF`.

180

PM - Print Mode

■ Profile Parameter PM	440
■ Session Parameter PM	441

The following topics are covered below:

Profile Parameter PM

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

Possible settings	C, P, I, R, or combinations	PM=C	An alternative character set is to be used. It can be defined by the profile parameters TAB1 and TAB2 .
	CI, CR, PI, PR	PM=P	The primary (standard) character set is to be used.
		PM=I	Specifies inverse, that is, right-to-left direction (for example, for use in Middle Eastern countries).
		PM=R	This resets the PM=I setting to normal (left to right) display direction.
Default setting	PR		
Dynamic specification	yes		
Specification within session	yes		System command GLOBALS or terminal command %V.
Application Programming Interface	USR1005N		See <i>SYSEXT - Natural Application Programming Interfaces</i> in the <i>Utilities</i> documentation.

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.

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

Session Parameter PM

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

Possible settings	PM=C	An alternative character set is used (see the module NATPM in the Natural source library).
	PM=D	Defines DBCS-only fields that do not contain shift-out/shift-in characters (see <i>Double-Byte Character Sets</i> in the <i>Operations</i> documentation).
	PM=I	Field values are displayed in inverse direction; that is, from right to left (for example, for use in Middle East countries).
	PM=N	No hardcopy of the display can be made.
Default setting	none	The standard character set is used.
Applicable Statements	DEFINE DATA DISPLAY FORMAT INPUT MOVE LEFT/RIGHT JUSTIFIED PRINT WRITE	



Note: More than one value may be specified.

Example:

```

LIMIT 1
  READ EMPLOYEES
  DISPLAY NOTITLE NAME
  DISPLAY NOTITLE NAME (PM=I)
  DISPLAY NOTITLE NAME
END

```

Result:

```

          NAME
-----
MORENO
          ONEROM
MORENO

```


This Natural profile parameter can be used to assure that fields are marked / referenced correctly.

See also *POS - Field Identification Function* in the *System Functions* documentation.

Possible settings	ON	The old Version 2.2 algorithm will be used and the correct field will be marked/referenced Note: Set POS22=ON only if you execute objects using the system function POS which were compiled with Natural Optimizer Compiler Version 2.2. For any other objects, the change in the internal POS algorithm does <i>not</i> lead to different results!
	OFF	The algorithm introduced with Natural Version 2.3 is used.
Default setting	OFF	
Dynamic specification	yes	
Specification within session	no	

As of Natural Version 2.3 for Mainframes, the internal algorithm for the computation of the system function POS (internal field identification) is different. As a result, if you execute with Version 2.3 or higher programming objects which use the POS function in conjunction with the MARK option of an INPUT or REINPUT statement or with the system variable *CURS-FIELD and which were compiled with Version 2.2 of the Natural Optimizer Compiler, the wrong field may be marked/referenced.



Caution: This parameter will be available only for a limited period of time to allow you a smooth transition from Version 2.2. It will be removed again with a subsequent release of Natural.

In a z/OS Parallel Sysplex environment, or if the Natural thread size makes buffer reallocations necessary, POS22=ON cannot always be guaranteed to yield the desired results. In these cases, the objects concerned should be recataloged in the new version of the Natural Optimizer Compiler to ensure correct POS result.

182

PRINT - Print File Assignments

■ PRINT Parameter Syntax	447
■ NTPRINT Macro Syntax	447
■ Keyword Subparameters for All Environments	448
■ Keyword Subparameters for AM=STD in All Environments	452
■ Keyword Subparameters for AM=STD in z/OS Environments	455
■ Keyword Subparameters for AM=STD in z/VSE Environments	457
■ Keyword Subparameters for AM=STD in BS2000/OSD Environments	458
■ Keyword Subparameters for AM=CICS	458
■ Keyword Subparameters for AM=COMP (Com-plete)	459
■ Keyword Subparameters for AM=SMARTS (Com-plete)	460
■ Keyword Subparameters for AM=IMS	460
■ Keyword Subparameters for DEFINE PRINTER Statement	461

This Natural profile parameter specifies the print files to be used during the session. Within a session, up to 31 logical print files (numbered 1 to 31) and the hardcopy print file (Number 0) can be used.

The old dynamic parameter `PRINTER` can be used as a synonym for `PRINT`.

`PRINT` corresponds to the `NTPRINT` macro in the parameter module `NATPARM`. To provide different print file definitions, `PRINT` or `NTPRINT` can be specified multiple times.

Possible settings	See Keyword Subparameters below.	
Default setting	See below.	
Dynamic specification	yes	The parameter <code>PRINT</code> can only be specified dynamically. In <code>NATPARM</code> , the macro <code>NTPRINT</code> must be used.
Specification within session	no	

The software components for accessing print files in different environments are called access methods. For the duration of a Natural session, each logical print file can be assigned to one access method only. The access method for a print file is determined by the keyword subparameter `AM` (see below).

In z/OS under TSO and in batch mode, print files need not be predefined in the JCL. Provided they are defined by subparameter `AM=STD`, they can be allocated dynamically during the session by a Natural program using the `DEFINE PRINTER` statement or the application programming interface `USR2021` (in library `SYSEXT`).

This document covers the following topics:

See also *Print and Work File Handling with External Datasets in a Server Environment* in the *Operations* documentation.

PRINT Parameter Syntax

With the `PRINT` parameter, you first specify one or more logical print file numbers, and then several keyword subparameters, which define the characteristics for these print files:

```
PRINT=((print-file-numbers),keyword-subparameters,...)
```

print-file-numbers

The file numbers must be specified first and enclosed in parentheses. The numbers can be from 0 to 31. They can be specified in any sequence. Multiple numbers must be separated from one another by commas or blanks. To specify a range of numbers, you can use a hyphen (-).

keyword-subparameters

The various types of keyword subparameters are described below.

For print files with different characteristics, you specify different `PRINT` parameters. If any previous definition (or default) for the same print file exists, only the values for the specified keyword subparameters are overwritten, all other values remain unchanged.

Examples:

```
PRINT=((2,12,18),AM=STD,DEST='PRINT**',OPEN=INITOBJ,CLOSE=CMD)
PRINT=((1,3,6-11,15),AM=NAF)
PRINT=((0),AM=STD,DEST=HARDCOPX)
```

NTPRINT Macro Syntax

With an `NTPRINT` macro, you first specify one or more logical print file numbers, and then several keyword subparameters which define the characteristics that are to apply to these print files:

```
NTPRINT (print-file-numbers),keyword-subparameters,...
```

print-file-numbers

The file numbers must be specified first and enclosed in parentheses. The numbers can be from 0 to 31. They can be specified in any sequence. Multiple numbers must be separated from one another by commas. To specify a range of numbers, you can use a hyphen (-).

keyword-subparameters

The various types of keyword subparameters are described below.

For print files with different characteristics, you specify different `NTPRINT` macros. If any previous definition (or default) for the same print file exists, only the values for the specified keyword subparameters are overwritten, all other values remain unchanged.

Examples:

```
NTPRINT (2,12,18),AM=STD,DEST='PRINT**',OPEN=INITOBJ,CLOSE=CMD
NTPRINT (1,3,6-11,15),AM=NAF

NTPRINT (0),AM=STD,DEST=HARDCOPX
```

Keyword Subparameters for All Environments

The following keyword subparameters are available: [AM](#) | [DEST](#) | [OPEN](#) | [CLOSE](#) | [ROUTE](#) | [CP](#)

AM - Type of Access Method

`AM=xxx` specifies the type of access method to be used.

For an online session, all print files to be used have to be assigned to a specific access method.

For a batch session, any print files not assigned to a specific access method will be automatically detected and assigned by the standard batch access method (`AM=STD`), provided that they have been predefined in the JCL. See also profile parameter [FAMSTD](#) (overwriting of print and work file access method assignments).

Value	Meaning
STD	Standard sequential batch files (batch, TSO, TIAM, VM/CMS OS simulation).
CMS	CMS disk and SFS files.
COMP	Com-plete print files.
CICS	CICS transient data or temporary storage.
NAF	Natural Advanced Facilities.
IMS	IMS TM destinations.
PC	Entire Connection.
USER	Third-party vendor print interface.
SMARTS	SMARTS print file.
ESS	Entire System Server.
NOM	Entire Output Management. Prints to an Entire Output Management container file without using the spool of the operating system. Refer to the Entire Output Management documentation for details.
OFF	Unassigned. No automatic assignments if FAMSTD=OFF is set.
0	Unassigned. Automatic assignments if FAMSTD=OFF is set. This is the default value.



Note: PRINT=OFF is equivalent to: PRINT=((1-31)), AM=OFF). It does not affect any of the other keyword subparameter specifications. PRINT=((0),AM=xxx) or NTPRINT (0),AM=xxx determines the hardcopy print access method and is equivalent to the profile parameter `HCAM=xxx`.

DEST - External Dataset Name

DEST=*name* specifies the print destination (1 - 8 characters).

This corresponds to the OUTPUT value of the DEFINE PRINTER statement (and can be overwritten by a DEFINE PRINTER OUTPUT specification).

The meaning of this keyword subparameter depends on the access method.

Access Method	Meaning of DEST
AM=STD	<p>DEST is the logical dataset name (DDNAME, LINK name, DTF name).</p> <p>If the destination is to be for multiple files, two asterisks (**) have to be specified for the file number. These will be replaced by the corresponding logical file number for each print file. A DEST value including two asterisks must be enclosed in apostrophes when it is used as a dynamic parameter.</p> <p>The default value is DEST='CMPRT**' for IBM and DEST='P**' for SIEMENS environments.</p> <p>Under z/VSE, only 7-character names are supported.</p>

Access Method	Meaning of DEST
AM=CICS	There is no default value for print files under CICS. Here, the DEST subparameter is mandatory, that is, CICS print files defined without a valid DEST specification are ignored. The Natural CICS interface also supports a variable (see TERMVAR parameter in the NCIPARM generation macro; &TID is the default) as part of the DEST value which, when being specified, is replaced by the actual CICS terminal ID. See also <i>Natural Print and Work Files under CICS</i> in the <i>TP Monitor Interfaces</i> documentation.
AM=CMS	For usage of DEST under CMS, refer to <i>Natural under VM/CMS</i> in the <i>Operations</i> documentation.
AM=IMS	Specifies the IMS TM destination.



Note: PRINT=((0),DEST=xxx) or NTPRINT (0),DEST=xxx determines the hardcopy print destination and is equivalent to the Natural profile parameter HCDEST=xxx.

OPEN - Time of File Opening

OPEN=xxx determines when the file is to be opened:

Value	The file is opened
INIT	for output at session initialization.
OBF	according to the default OPEN value for the different environments (batch, CICS, Com-plete, TSO).
OBJ	when the execution of the first object which accesses the file starts. This is the general default, except for AM=COMP and AM=IMS.
OBJ1	when the execution of the first object on Level 1 that accesses the file starts. Otherwise, it is opened when it is first accessed.
ACC	when it is first accessed by a statement. This is the default for AM=COMP and AM=IMS.
INITOBF	for output at session initialization. Any subsequent re-opening of the file sets the default OPEN value for the different environments (batch, CICS, Com-plete, TSO).
INITOBJ	for output at session initialization. Any subsequent re-opening of the file will be performed when the execution of the first object which accesses the file starts.
INITOBJ1	when the execution of the first object on Level 1 that accesses the file starts. Otherwise, it is opened when it is first accessed.
INITACC	for output at session initialization. Any subsequent re-opening of the file will be performed when it is first accessed by a statement.

CLOSE - Time of File Closure

`CLOSE=xxx` determines when the file is to be closed:

Value	The file is closed
OBJ	either when processing of the object in which it was first accessed is finished or when command mode, NEXT mode or MAINMENU is reached.
CMD	when command mode, NEXT mode or MAINMENU is reached. This is the default for <code>AM=NAF</code> , <code>AM=COMP</code> and <code>AM=IMS</code> .
FIN	at session end (this is the default for <code>AM=STD</code>). With <code>CLOSE=FIN</code> , a <code>DEFINE PRINTER</code> statement causes an error if the printer was opened already. A <code>CLOSE PRINTER</code> statement for the printer is ignored.
USER	only if the file is open and one of the following conditions is true: <ul style="list-style-type: none"> ■ a <code>CLOSE PRINTER</code> statement is issued, ■ a <code>DEFINE PRINTER</code> statement is issued, ■ the session terminates.

ROUTE - Logical Print File Routing

`ROUTE=xxx` determines whether logical print file routing is done according to the `OUTPUT` clause of the `DEFINE PRINTER` statement.

ON	Print file routing is done. The target print file can be any available print file except PC. This is the default value.
OFF	No print file routing is done.
<i>am</i>	Print file routing is done to printers of the specified access method <i>am</i> only. Possible value is any valid print file access method (see description of subparameter AM above). PC is not allowed for <i>am</i> .

Print file routing means that, if the name defined in the `OUTPUT` clause of a `DEFINE PRINTER` statement denotes a print file destination which is defined by a different logical printer, all print output is routed to this print file. If no printer with the specified name is found, the print output can be routed to any free printer.

CP - Code Page for Print Output

This keyword subparameter defines the code page for the print output. It is assumed that all code page data, for example, Natural sources, contents of A-format fields, etc., are stored in this code page. If no code page is specified with the keyword subparameter **CP**, the code page resulting from the evaluation of the profile parameter **CP** is used.

If Natural code page support is disabled (for example, by parameter **CP=OFF**), any value specified for this parameter is ignored.

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

Value	Meaning
1 - 64 characters	The name of the desired code page. Any character string is possible, but must be predefined by one of the code page parameters CCSID , CCSN , IANA or ALIAS of the macro NTCPAGE in the source module NATCONFIG .

Keyword Subparameters for AM=STD in All Environments

The following keyword subparameters are available: **RECFM** | **BLKSIZE** | **LRECL** | **TRUNC** | **PAD** | **PADCHRO** | **ASA** | **STRIP**

RECFM - Default Record Format of Dataset

RECFM=xxxx determines the default record format of the dataset.

The following formats are supported:

F	Fixed
V	Variable
U	Undefined
B	Blocked
S	Spanned
A	ASA
M	Machine control characters

The following values and also combinations of values are possible:

Possible value:	F, FA, FM, FB, FBA, FBM, V, VA, VM, VB, VBA, VBM, VBS, VBSA, VBSM, U, UA, UM
Default value:	RECFM=VBA (variable blocked with ASA).

The RECFM specification only applies if no record format is predefined in the JCL or (z/OS only) in the dataset DCB.

BLKSIZE - Default Block Size of Dataset

BLKSIZE=nnnnn determines the default block size (in bytes) of the dataset.

Possible values:	0 or 8 to 32767
Default value:	1016

The BLKSIZE specification only applies if no block size is predefined in the JCL or (z/OS only) in the dataset DCB.

LRECL - Default Record Length of Dataset

LRECL=nnn determines the default record length (in bytes) of the dataset.

Possible values:	0 or 5 - 254
Default value:	0

This subparameter is used particularly to check for truncation and padding.

For RECFM=V (B) the LRECL value includes a 4-byte record descriptor word.

If LRECL=0 is defined, the following applies:

- With RECFM=V (B), LRECL defaults to the minimum of BLKSIZE-4 and 254.
- With RECFM=U, LRECL defaults to BLKSIZE.
- With RECFM=F (B), the maximum record length in the Natural program being executed is taken when the file is opened. If no record length from a program is available when the file is opened, for example with OPEN=INIT, a record length of 132 is taken (plus 1 for ASA or a machine control character and/or plus 4 for a record-descriptor word if the record format is variable).

The LRECL specification only applies if no record length is predefined in the JCL or (z/OS only) in the dataset DCB.

TRUNC - Truncation of Output Records

TRUNC=xxx determines whether the output records are truncated:

ON	Output records that are longer than the record length (LRECL) of the dataset will be truncated. This is the default value.
OFF	Error NAT1512 will be issued if an output record is longer than the dataset record length.

PAD - Padding of Output Records

PAD=xxx determines whether the output records are padded or not (applies only to datasets of fixed record length):

ON	Output records that are shorter than the record length (LRECL) of the dataset will be padded with padding characters defined by keyword subparameter PADCHRO . This is the default value.
OFF	Error NAT1510 will be issued if an output record is shorter than the dataset record length.

PADCHRO - Padding Character of Output Records

This subparameter defines the character which is used for padding if PAD=ON is defined for the print file.

Possible values:	'x '	(one character x within single quotes)
	x 'xx '	(one hex character xx)
Default value:	' '	(blank or x'40')

ASA - Use of ASA Record Format

ASA=xxx determines whether the ASA record format is used.

ON	An ASA character is included in the output print records. Under z/OS, this enforces ASA record format, regardless of the RECFM setting in the DCB or the RECFM subparameter. This is the default value.
OFF	No ASA character is included in the output print records. Under z/VSE batch access method (AM=STD), a valid ASA character must be supplied in column one of the output record if the output file is a spool file, otherwise error NAT1530 will be issued.

STRIP - Inhibit Removal of Trailing Blanks

Trailing blanks are stripped off for batch sequential print files (AM=STD) if the dataset is defined with variable record format (RECFM=VB) to reduce disk space. This may cause problems with subsequent applications accessing this dataset due to the missing blanks. These problems can be avoided by setting STRIP=OFF.

ON	Trailing blanks are stripped off. This is the default value.
OFF	Trailing blanks are not stripped off.

Keyword Subparameters for AM=STD in z/OS Environments

The following keyword subparameters are available:

REREAD | **FREE** | **BUFNO** | **DISP** | **VMAX**

REREAD - Closing of Tape File Datasets

REREAD=xxx sets the REREAD option for the closing of the tape file:

ON	The REREAD option is set for the CLOSE SVC. This causes the volume to be repositioned to reprocess the dataset . This is the default value.
OFF	The REREAD option is not set for the CLOSE SVC.

FREE - Dataset De-allocation at File Closure

FREE=xxx determines whether the dataset is de-allocated when the file is closed:

ON	The FREE option is set for the CLOSE SVC, which means that the dataset is de-allocated when it is closed (and not at step termination).
OFF	The FREE option is not set for the CLOSE SVC. This is the default value.

BUFNO - Default Number of z/OS I/O Buffers of Dataset

BUFNO=nnn defines the default number of z/OS I/O buffers of the dataset.

Possible values	0 - 255
Default value	0
	In this case, z/OS allocates five I/O buffers per default.

The number of I/O buffers can improve the performance of print file access dramatically. Note that the storage for I/O buffers is allocated below the 16 MB line.

The BUFNO specification applies only if the BUFNO parameter is not specified in the JCL for the dataset.

DISP - Open Print File for Modification

DISP=xxx determines whether the print file is opened for modification.

This corresponds to the JCL DD statement subparameter DISP=MOD.

MOD	New records are added at the end of the file.
NOMOD	The print file is rewritten from the start. This is the default value.

VMAX - Control LRECL for Variable Record Format

VMAX=xxx controls the LRECL setting for an output file with variable record format (RECFM=V).

ON	Providing a nonzero BLKSIZE value exists for the file, VMAX=ON sets LRECL=BLKSIZE-4 for variable record format, regardless of the LRECL setting in the DCB or the LRECL subparameter.
NAT	LRECL is set to the length +4 of the largest record in the application program if this value is less than LRECL in the DCB for the dataset.
OFF	LRECL from the DCB for the dataset or the LRECL subparameter is used. This is the default value.

Keyword Subparameters for AM=STD in z/VSE Environments

The following keyword subparameters are available:

SYSNR | **LABEL** | **REWIND**

SYSNR - Logical VSE SYS Number

SYSNR=*nn* determines the logical VSE SYS number.

Possible values:	1 - 99
Default value:	By default, the SYS number is print file number plus 40 for print files 1 - 31; for print file 0, that is the hardcopy printer, the default is SYSLST. Example: The z/VSE default SYS number for print file 11 is 11 + 40 >= SYS051.

LABEL - Tape Label Processing

LABEL=*xxx* determines the tape label processing:

ON	The tape is in standard label format. This is the default value.
OFF	The tape is unlabeled with front tape mark.
NOTM	The tape is unlabeled without front tape mark.

REWIND - Action at File Closure

REWIND=*xxx* determines the action to be taken when a tape file is closed:

ON	The tape is rewound when the file is closed. This is the default value.
OFF	The tape is not rewound when the file is closed.
UNLOAD	The tape is unloaded when the file is closed.

Keyword Subparameters for AM=STD in BS2000/OSD Environments

The following keyword subparameter is available: [DISP](#) | [FREE](#)

DISP - File Open Mode

DISP=xxx determines the open mode of the file:

EXT	The open mode is set to EXTEND.
NOEXT	The open mode is set to the default value OUTPUT. This is the default value.

FREE - Release Linkname at File Closure

FREE=xxx determines whether the linkname of the file is released when the destination file is switched over to another one.

ON	The linkname is released.
OFF	The linkname is kept.

Example:

```
DEFINE PRINTER (1) OUTPUT 'P01'  
WRITE (1) 'TEST'  
CLOSE (1)  
DEFINE PRINTER (1) OUTPUT 'FILE=REPORT01.NEW, LINK=LINKP01'
```

If FREE is set to ON, the linkname is released; with FREE=OFF, it is kept.

Keyword Subparameters for AM=CICS

The following keyword subparameters are available:

[TYPE](#) | [DISP](#)

TYPE - Type of CICS Storage Medium

TYPE=xxxx specifies the type of CICS storage medium to be used:

MAIN	Temporary main storage.
AUX	Temporary auxiliary storage.
TD	Transient data.

The default value used depends on the [DEST](#) parameter setting. If the DEST subparameter value matches a valid CICS transient data queue, the TYPE subparameter defaults to TD, otherwise MAIN will be taken as the default value.

DISP - CICS Temporary Storage Queue Disposition

DISP=(xxx,xxx) specifies the CICS temporary storage queue disposition.

Possible value pairs are:

(NEW,KEEP)	The storage queue is deleted when the file is opened. This is the default value.
(NEW,DELETE)	The storage queue is deleted when the file is opened and when it is closed.
(OLD,DELETE)	The storage queue is deleted when the file is closed.
(OLD,KEEP)	The storage queue is not deleted.



Note: The DISP specification does not apply to CICS extra-partition transient data queues.

Keyword Subparameters for AM=COMP (Com-plete)

The following keyword subparameter is available: DRIVER

DRIVER - Name of Com-plete Print Driver

DRIVER=name specifies the *name* of the Com-plete print driver to be used.

Keyword Subparameters for AM=SMARTS (Com-plete)

The following keyword subparameter is available: DEST

DEST - Logical Printer

`DEST=print-server-queue` The environment variable `SAG_APS_LPD_xyz` defines a logical printer under complete, where `xyz` is the name of the print server queue.

If the environment variable `SAG_APS_LPD_xyz` exists for the specified DEST, the output is directly routed to that line printer. For more information, see the *Complete Initialization and Startup Manual*, section *Defining Terminals and Printers*.

`DEST=printer-file-name` If no print server queue for that printer is available, DEST specifies a printer file name. It specifies the location of the output file in the file system. The name of the output file is generated from the `userId` and a sequence number.

Since the DEST clause is restricted to an 8 character maximum, it is useless to define a file with absolute PFS path specification. The name specified in the DEST clause is relative to the print file root directory. The print file root directory is specified with the environment variable `NAT_PRINT_ROOT`.

Example:

```
NAT_PRINT_ROOT=/nat/printer
DEST=printer1
UserId=xyz
```

The first output will be written to file `/nat/printer/printer1/xyz1`.

To specify a file with absolute path definition, the OUTPUT clause of the DEFINE PRINTER statement must be used.

Keyword Subparameters for AM=IMS

The following keyword subparameters are available:

[BLKSIZE](#) | [DRIVER](#)

BLKSIZE - Size of the Print Buffer

BLKSIZE=*nnnnn* specifies the size of the print buffer sent to the IMS TM destination.

DRIVER - Name of Natural IMS Print Driver

DRIVER=*name* specifies the *name* of the Natural IMS print driver to be used.

For possible values, see *NIMPARM Macro Parameters* and *Support of the Natural WRITE (n) Statement* in the section *Natural under IMS TM* in the *TP Monitor Interfaces* documentation.

Keyword Subparameters for DEFINE PRINTER Statement

With the following keyword subparameters, you can set default values for the `DEFINE PRINTER` statement options of the same names (see the *Statements* documentation). When a printer is closed, all `DEFINE PRINTER` statement options are reset to their default values.

The following keyword subparameters are available:

`PROFILE` | `NAME` | `FORMS` | `DISP` | `COPIES` | `CLASS` | `PRTY`

PROFILE - Name of Printer Control Characters Table

PROFILE=*name* specifies the *name* of printer control characters table (`NTCCTAB` macro).

NAME - Name of Listing

NAME=*name* specifies the listing *name*.

FORMS - Name of Listing Forms

FORMS=*name* specifies the listing forms *name*.

DISP - Listing Disposition

DISP=*disposition* specifies the listing *disposition* (HOLD, KEEP, DELETE or LEAVE).

COPIES - Number of Copies

COPIES=*nnn* specifies the number of copies to be printed (1 - 255).

CLASS - Spool Class

CLASS=*class* specifies the spool class (1 byte).

PRTY - Listing Priority

PRTY=*nnn* specifies the listing priority (1 - 255).

183

PROFILE - Activate Dynamic Parameter Profile

This Natural profile parameter can be used to activate a dynamic parameter profile.

When you invoke Natural with dynamic profile parameters, instead of having to specify a whole string of individual parameters each time you invoke Natural, you can specify the string of parameters once, store this string under a profile name and then invoke Natural with only one dynamic parameter as follows:

```
PROFILE=profile-name
```

The parameters defined under this profile are passed to Natural as dynamic profile parameters. You create and maintain these profiles with the utility `SYSPARM` (described in the *Utilities* documentation).

Possible settings	1-8 characters	Profile-name. Or special options, see below.
Default setting		
Dynamic specification	yes	
Specification within session	no	

Special Options

In addition, the `PROFILE` parameter provides the following special options:

PROFILE=AUTO	<p>Natural takes the current TP user ID (as contained in the system variable <code>*INIT-USER</code>) as profile name, which means that the profile defined under the name corresponding to that ID is used.</p> <p>If no such profile is found, a profile named <code>AUTO</code> is used instead (if available). You can define such an <code>AUTO</code> profile as default profile for users without individual profiles.</p>
--------------	---

PROFILE=TERMINAL	<p>Natural takes the current terminal ID (as contained in the system variable *INIT-ID) as profile name, which means that the profile defined under the name corresponding to that ID is used.</p> <p>If no such profile is found, a profile named TERMINAL is used instead (if available). You can define such a TERMINAL profile as default profile for users without individual profiles.</p>
PROFILE=PROGRAM	<p>Natural takes the name of the program currently executing as Natural (as contained in the system variable *INIT-PROGRAM) as profile name, which means that the profile defined under this name is used.</p> <p>If no such profile is found, a profile named PROGRAM is used instead (if available). You can define such a PROGRAM profile as default profile for users without individual profiles.</p>

By default, the profile is read from the current FNAT system file.

To read it from a different system file, you can specify the desired database ID, file number, password and cipher code after the *profile-name* (or after one of the above special options) as follows:

```
PROFILE=(profile-name,dbid,fnr,password,cipher-code)
```

To ensure that all profile parameters are read from the same system file (other than FNAT), specify the following in the parameter module:

```
PROFILE=(,dbid,fnr)
```

If the PROFILE parameter is specified within a parameter module, it is evaluated *after* the other parameters in the parameter module, but *before* any dynamically specified profile parameters are evaluated; this means that parameters specified within the profile can be overridden by individually specified dynamic parameters.

To restrict the use of a profile, you can use the profile parameter [USER](#).

Unlike other parameters, a PROFILE parameter specification cannot be overwritten by another PROFILE. So you can have multiple parameter profiles which are evaluated all in a sequence.

The PROFILE parameter cannot be used with ADARUN MODE=SINGLE.

184

PROGRAM - Non-Natural Program Receiving Control after Termination

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

Possible settings	1-8 characters	Non-Natural back-end program.
	numeric value	Setting a numeric value, for example PROGRAM=0, indicates “no back-end processing”. This is particularly relevant when Natural is invoked by a front-end program, because a default may be taken if PROGRAM is blank or not specified; see <i>Front-End Invoked via XCTL</i> in the <i>TP Monitor Interfaces</i> documentation.
Default setting	none	
Dynamic specification	yes	It can also be set dynamically from within a Natural program by calling the Natural subprogram CMPGMSET which is provided in the library SYSEXTP.
Specification within session	yes	
Application Programming Interface	USR4001N (available on mainframes)	See <i>SYSEXT - Natural Application Programming Interfaces</i> in the <i>Utilities</i> documentation.
	USR6204N (available on all platforms)	

Data for the program specified with the PROGRAM parameter can be supplied with the TERMINATE statement.

For the conventions of calling non-Natural back-end programs, see *Back-End Program Calling Conventions* in the *Operations* documentation.

CICS-Specific Information:

In addition to back-end programs, the Natural CICS interface also supports back-end transactions which may be specified via `RET=XXXX` or `RTI=XXXX` or `STR=XXXX` instead of a program name, with `XXXX` being a valid CICS transaction ID.

- `RET=XXXX` or `RTI=XXXX` indicates that control has to be passed to CICS together with a return transaction ID by a CICS `RETURN TRANSID ('XXXX')` command.
- `RTI=XXXX` indicates that control has to be passed to CICS with a return transaction ID by a CICS `RETURN TRANSID ('XXXX' IMMEDIATE)` command.
- `STR=XXXX` indicates that a new transaction has to be started by a CICS `START TRANSID ('XXXX') TERMID (*INIT-ID)`, before relinquishing control via a CICS `RETURN` command.

185

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. When used as a profile parameter, the `PS` parameter is honored in batch mode only and defines the physical page size. In online mode, the physical page size is always set to the physical screen height.

Possible settings	1 - 250	Maximum number of lines per page.	
	0	<p>The physical page size is to be used.</p> <p>If PS=0 is specified for the first report to be output (Report 0), the physical-device page-size minus 1 will be used.</p> <p>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.</p>	
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 <i>SYSEXT - Natural Application Programming Interfaces</i> in the <i>Utilities</i> documentation.	

See also *Page Size - PS Parameter* in the *Programming Guide*.

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

This Natural profile parameter controls the mode of operation under CICS. When Natural is executing under control of the TP monitor CICS, two modes are possible: conversational and pseudo-conversational.

Possible settings	ON	PSEUDO=ON enables pseudo-conversational mode. In this mode, a Natural session is a sequence of different transactions. After each output to the terminal, all Natural work areas are saved and the transaction is terminated. When the user responds to a message by pressing ENTER (or any other input key), a new transaction is initiated. The Natural work areas are restored, the terminal input is read and the Natural session is continued. The transaction identification of each following transaction can be set dynamically by calling the subroutine CMTRNSET, which is provided in the library SYSEXTP.
	OFF	PSEUDO=OFF disables pseudo-conversational mode and enables conversational mode. In conversational mode, a Natural session is one transaction which is active for as long as the Natural session is active. Note for CICS: A specification of PSEUDO=OFF is ignored for Natural server sessions. See <i>Natural Server Sessions under CICS</i> in the <i>Operations</i> documentation.
Default setting	ON	
Dynamic specification	yes	
Specification within session	no	

For more information, refer to Natural under CICS, section *TYPE - Thread Type for Group* in the *TP Monitor Interfaces* documentation.

187

RCA - Resolve Addresses of Static Non-Natural

Programs

This Natural profile parameter controls the *dynamic* linking of static non-Natural programs to the Natural nucleus during initialization of the Natural session.

Possible settings	ON	At Natural startup, the list of all static non-Natural programs to be linked to Natural is scanned and a load request is issued for all modules whose addresses are unresolved. If a load request fails, no error message is issued. The use of RCA=ON is <i>not</i> recommended, as it causes a lot of processing overhead at Natural startup.
	OFF	No dynamic linking of static non-Natural programs is performed.
	<i>name-list</i>	If RCA= <i>name-list</i> is specified, the list of static non-Natural programs to be linked to Natural is extended by the specified name list. A load request is issued for these modules even if they are already linked. In this way, it is possible to replace linked non-Natural programs. If a load request fails, an error message is issued. If more than one name is specified, each must be separated from the next by a comma and the list must be enclosed within parentheses as shown below: RCA=(PROGRAM1 , PROGRAM2 , PROGRAM3)
Default setting	OFF	
Dynamic specification	yes	
Specification within session	no	

Static non-Natural programs have to be defined for being linked to Natural either internally (by using the macro NTINV within the modules NATPARM and NATCONFIG) or externally (by using the profile parameter [CSTATIC](#)).

If the external name of the non-Natural program is different from the internal one (as used by the `CALL` statement), you can use either the profile parameter `RCALIAS` or the macro `NTALIAS` to define which external name is to be used for the load request.

Under CICS: A PPT entry has to be defined to allow the load request for a non-Natural program. Static non-Natural programs are called via standard linkage conventions rather than `EXEC CICS LINK` requests.

188

RCALIAS - External Name Definition for Non-Natural Programs

■ RCALIAS Parameter Syntax	474
■ NTALIAS Macro Syntax	475
■ Examples of NTALIAS Macro	475
■ Example of RCALIAS Parameter	475

This Natural profile parameter can be used to define the external names of static non-Natural programs which are defined by profile parameter [RCA](#) and loaded for dynamic linking during the initialization of a Natural session. It corresponds to the [NTALIAS](#) macro in the parameter module NATPARM.

Possible settings	<i>internal-program-name</i>	List of name pairs: <i>internal-program-name</i> defines the internal name of a non-Natural program (used by the CALL statement) that must also be defined by the parameter RCA or CSTATIC (only if RCA=ON). <i>external-program-name</i> defines the corresponding external alias name for the load request during session initialization.
	<i>external-program-name</i>	
	OFF	No external names for RCA modules are defined. This value can be specified dynamically only.
Default setting	OFF	
Dynamic specification	yes	This parameter can only be specified dynamically. In the Natural parameter module NATPARM, the macro NTALIAS must be used instead.
Specification within session	no	

The following topics are covered below:

RCALIAS Parameter Syntax

The parameter syntax of RCALIAS is as follows:

```
RCALIAS=(  
internal-program-name1,external-program-name1,internal-program-name2,external-program-name2,...)
```


NTALIAS Macro Syntax

The NTALIAS macro is specified as follows:

```
NTALIAS internal-program-name,external-program-name
```

Examples of NTALIAS Macro

```
NTALIAS PROGRAM1,ALIAS1  
NTALIAS PROGRAM2,ALIAS2
```

Example of RCALIAS Parameter

```
RCA=(PROGRAM1,PROGRAM2),RCALIAS=(PROGRAM1,ALIAS1,PROGRAM2,ALIAS2)
```


189

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 <code>FIND</code> loop will continue by reading the next record.
Default setting	ON	
Dynamic specification	yes	
Specification within session	no	

190

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	

191

RDACT - (Internal Use)

This Natural profile parameter is reserved for internal use by Natural.



Caution: Do not change its setting.

This Natural profile parameter is used to define user exits for the Natural Data Collector of the `SYSRDC` utility and, optionally, a work area size for each exit. If linked, the exit gets control from the Natural Data Collector at certain points within Natural. Specific session information is passed to the exits.

Possible settings	A list of user exit names with a work area size for each user exit.	With <code>RDCEXIT</code> , any exit names can be specified. In the Natural parameter module <code>NATPARM</code> , the exit names are automatically added to the <code>CSTATIC</code> profile parameter's setting list.	
Default setting	none		
Dynamic specification	yes	If <code>RDCEXIT</code> is specified dynamically, the exits must be defined in the profile parameters <code>CSTATIC</code> or <code>RCA</code> (<code>RCA</code> can also be specified dynamically). Optionally, the size of the exit work area can be specified after the exit name.	
		Possible settings	400 - 32760
		Default setting	400
		Example	<code>RDCEXIT=(MYEXIT,2000,RDCEX1)</code>
Specification within session	no		

For details, refer to *Debugging and Monitoring* in the *SYSRDC Utility* documentation.

193

RDCSIZE - Size of Buffer for the Natural Data Collector

This Natural profile parameter specifies the buffer size for the Natural Data Collector which is used by the SYSRDC utility. In addition, it controls the trace recording function of the data collector.

Alternatively, you can use the equivalent Natural profile parameter [DS](#) or macro `NTDS`; see *Using Optional Macros in a Natural Parameter Module* in the *Operations* documentation to specify the buffer size.

Possible settings	2 - 128	Buffer size in KB. To activate the data collector (without trace recording), you specify RDCSIZE=2. To also activate the trace recording, you have to set RDCSIZE to a setting greater than 2. If the requested space is not available, the Natural Data Collector cannot be used.
	0	Deactivates the data collector.
Default setting	0	
Dynamic specification	yes	
Specification within session	no	

For details, refer to the *SYSRDC Utility* documentation.

194

RDNODE (Internal Use)

This parameter is reserved for internal use by Natural.



Caution: Do not change its setting.

195

RDPORT (Internal Use)

This parameter is reserved for internal use by Natural.



Caution: Do not change its setting.

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

It allows you to define up to 10 remote directory servers. For each remote directory server, you specify up to 5 subparameters.

RDS is specified on the client side only.

Possible settings	<i>server-name</i>	The server name (1 - 8 characters). There is no default, the value must be specified.	
	<i>server-node</i>	The server node (1 - 8 characters). There is no default, the value must be specified.	
	<i>subprogram</i>	The name of the subprogram titled CALLNAT, (1 - 8 characters) to be used as interface (default is RDSSCDIR).	
	<i>logon-indicator</i>	A logon indicator. If nothing is specified, blank is the default.	
		L	The client initiates a Natural logon to the server with the library name of the current library on the client.
		blank	No server logon will be executed.
		Note for Windows platforms: instead of L, check the selection box.	
	<i>transport-protocol-name</i>	The transport protocol to be used. ACI is the only possible value and the default.	
Default setting	none	Subparameter defaults, see above.	
Dynamic specification	yes	See below.	

Specification within session	no	
------------------------------	----	--

For dynamic specification the syntax is as follows.

Using 1 server:

RDS=(server-name,server-node-name,subprogram,logon-indicator,transport-protocol-name)

Using 2-10 servers:

RDS=((server-name,server-node
name,subprogram,logon-indicator,transport-protocol-name)(server-name,server-node
name,subprogram,logon-indicator,transport-protocol-name)...(server-name,server-node
name,subprogram,logon-indicator,transport-protocol-name))

For additional information on Natural RPC, see the *Natural Remote Procedure Call (RPC)* document-
ation.

197

READER - z/VSE System Logical Units for Input

This Natural profile parameter specifies the z/VSE system logical units which are to be used by Natural for input.

Possible settings	READER=(<i>n</i> , <i>device</i> , ...)	<i>n</i> is 0 for CMSYNIN and 1 for CMOBJIN <i>device</i> is either SYSRDR or SYSIPT
Default setting	READER=(0, SYSRDR, 1, SYSIPT)	By default, the primary input stream (CMSYNIN) is read from SYSRDR and the input stream (CMOBJIN) is read from SYSIPT (if required). If CMSYNIN or CMOBJIN are disk or tape files, the associated READER subparameter is ignored.
Dynamic specification	yes	
Specification within session	no	

This overwriting of a system logical unit number only applies if the relevant file is a card file.

198

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	<p>Important: This profile parameter only applies to Natural objects of Version 2.3 and above.</p> <p>An error message is issued if an inconsistency concerning a Version 2.2 program and/or global data area is detected.</p> <p>Natural automatically adjusts the object and disables the system commands CATALOG and SAVE.</p> <p>Note: If the adjusted object invokes an object from a steplib library that also has to be adjusted, the object from the steplib library will be copied to the library of the invoking object.</p> <p>Note: RECAT=ON is not possible for programs cataloged with the Natural Optimizer Compiler.</p>
	OFF	Natural issues an error message.
Default setting	OFF	
Dynamic specification	yes	
Specification within session	no	
Application Programming Interface	USR1005N	See <i>SYSEXT - Natural Application Programming Interfaces</i> in the <i>Utilities</i> documentation.

199

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.

By default, Natural automatically issues an internal `REINPUT` statement if invalid data have been entered in response to an `INPUT` statement. With this parameter, you can switch this mechanism off. This will allow you to handle such input errors yourself in your application.

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

Possible settings	ON	An internal REINPUT statement is issued when invalid data have been entered.	
	OFF	An internal REINPUT statement is not issued when invalid data have been entered.	
Default setting	ON		
Dynamic specification	yes		
Specification within session	yes	Applicable Statements:	SET GLOBALS
		Applicable Command:	GLOBALS
Application Programming Interface	USR1005N		

200

RELO - Storage Thread Relocation

This Natural profile parameter controls the relocation of the Natural thread after a terminal I/O in a thread environment (CICS, Com-plete, IMS TM, UTM and Natural as a Server).

Possible settings	ON	The Natural thread and all the buffers contained therein can be relocated to another storage area if the original storage area has been occupied by another user after a terminal I/O.
	OFF	<p>No relocation is performed. The Natural thread and all the buffers therein remain located at the same virtual address after the terminal I/O.</p> <p>This setting applies to CICS, Com-plete and server environments only. In all other thread environments, Natural cannot guarantee that the thread remains located at the same address.</p> <p>Notes for CICS:</p> <ul style="list-style-type: none"> ■ When using TYPE=GETM threads under CICS, RELO=OFF has the same effect as the PSEUDO=OFF setting of the PSEUDO profile parameter. See also TYPE (thread type for group) in the section <i>Natural under CICS</i> in the <i>TP Monitor Interfaces</i> documentation. ■ A specification of RELO=OFF is ignored for <i>Natural Server Sessions under CICS</i> using TYPE=GETM threads.
	FORCE	<p>This will force a relocation of the Natural thread and all the buffers contained therein to another storage area. This can be useful for testing purposes in some environments.</p> <p>This setting does not apply under UTM.</p>
Default setting	ON	
Dynamic specification	yes	
Specification within session	no	

201

RFILE - File for Recordings

This Natural profile parameter specifies where recordings (that is, the data recorded by the Recording function) are stored.

Possible settings	SPAD	Recordings will be stored in the scratch-pad file. (If no scratch-pad file is defined, the recordings will be stored in the system file FUSER.)
	FUSER	Recordings will be stored in the system file FUSER.
	FNAT	Recordings will be stored in the system file FNAT.
Default setting	SPAD	
Dynamic specification	yes	
Specification within session	no	

For details on the Recording function, see *Recording Utility* in the *Utilities* documentation.

202

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 <code>WHERE</code> clause or an <code>ACCEPT/REJECT</code> statement). This reduces the number of ISNs which are contained in the hold queue. This may, however, cause additional performance overhead as an Adabas call is required for each ISN released.
	OFF	The ISN of each record which has been placed in hold status is <i>not</i> released until the end of the transaction.
Default setting	OFF	
Dynamic specification	yes	
Specification within session	no	
Application Programming Interface	USR1005N	See <i>SYSEXT - Natural Application Programming Interfaces</i> in the <i>Utilities</i> documentation.

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.

203

RJESIZE - Initial Size of NATRJE Buffer

This Natural profile parameter specifies the initial size of the NATRJE buffer.

With the Natural utility NATRJE (described in the *Utilities* documentation), JCL jobs can be collected and then submitted all at once. RJESIZE is used to set the initial size of the buffer which holds the JCL jobs before they are submitted.

Alternatively, you can use the equivalent Natural profile parameter [DS](#) or macro [NTDS](#), see *Using Optional Macros in a Natural Parameter Module* in the *Operations* documentation to specify RJESIZE.

Possible settings	1 - 2097151	Buffer size in KB. If the initial size is not sufficient, Natural automatically increases (repeatedly, if necessary) the buffer size in increments of 8 KB.
	0	Disables the NATRJE utility.
Default setting	8	
Dynamic specification	yes	
Specification within session	no	

204

RM - Retransmit Modified Fields

This Natural profile parameter controls the retransmission of modified fields.

Some TP monitors translate input data automatically to upper-case characters. As Natural's screen optimization only retransmits modified data back to the screen, the TP-monitor translation may cause input for a field which has been modified not to be retransmitted.

Possible settings	ON	Natural always sends back all modified fields.
	OFF	Natural sends back modified fields only if they have been changed.
Default setting	OFF	
Dynamic specification	yes	
Specification within session	no	

205

ROSY - Read-Only Access to System Files

This Natural profile parameter disables modifications on the Natural system files [FNAT](#), [FUSER](#), [FDIC](#) and [FSEC](#).

Possible settings	ON	No data can be written to, modified on or deleted from the system files. Natural issues an error message instead of performing any action that would modify any of these system files.
	OFF	Data can be written to, modified on and deleted from the system files.
Default setting	OFF	
Dynamic specification	yes	
Specification within session	no	



Notes:

1. If your system files are specified as read-only (`ROSY=ON`), the Natural utilities/functions Recording and NATPAGE cannot be used, because they write data to the Natural system files FNAT and/or FUSER.
2. Therefore, it is recommended that you allocate and use a so-called scratch-pad file to hold these temporary data. The scratch-pad file is optional and must be defined as recoverable by using the macro [NTLFILE](#) or the profile parameter [LFILE](#). The above functions then write their data to this file instead of FNAT/FUSER.
3. With `ROSY=OFF`, a scratch-pad file should also be defined if you use the Recording and NATPAGE functions with database transaction logic, as that might lead to unpredictable results with FNAT/FUSER.
4. If a system file is specified as read-only in the corresponding profile parameter [FNAT](#), [FUSER](#) or [FSEC](#), it is not possible to enable updates by setting `ROSY=OFF`.

■ RPC Parameter Syntax	512
■ NTRPC Macro Syntax	512
■ Keyword Subparameters	512
■ RPC Parameter Example	513
■ NTRPC Macro Example	513

This Natural profile parameter allows you to specify subparameters which control the handling of Natural RPC. It corresponds to the macro `NTRPC` in the parameter module `NATPARM`.

Possible settings	subparameters	see <i>Keyword Subparameters</i> below.
Default setting	none	
Dynamic specification	yes	This parameter can only be specified dynamically. In the Natural parameter module <code>NATPARM</code> , the macro <code>NTRPC</code> must be used instead.
Specification within session	no	

The following topics are covered below:

RPC Parameter Syntax

The parameter syntax of RPC is as follows:

RPC=(*keyword_subparameter1=value,keyword_subparameter2=value,...*)

For names and values of *keyword_subparameters*, see *Keyword Subparameters* below.

NTRPC Macro Syntax

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

NTRPC *keyword_subparameter1=value,keyword_subparameter2=value,...*

keyword_subparameter - see below.

Keyword Subparameters

There are three groups of keyword subparameters available that apply to

- **both Client and Server**
(`ACIVERS` | `MAXBUFF` | `RPCSIZE` | `SERVER` | `CPRPC`)

- **the Server only**
(LOGONRQ | NTASKS | RPCUCT | SRVCMIT | SRVNAME | SRVNODE | SRVTERM | SRVUSER | SRVWAIT | TRACE | TRANSP)
- **the Client only**
(AUTORPC | COMPR | DFS | RDS | RPCSDIR | TIMEOUT | TRYALT)

RPC Parameter Example

For the client:

```
RPC=(RPCSIZE=80,MAXBUFF=30,AUTORPC=ON,DFS=(MYSERV,MYNODE,,ACI))
```

For the server:

```
RPC=(RPCSIZE=80,MAXBUFF=30,SRVNAME=MYSERV,SRVNODE=MYNODE,SERVER=ON)
```

NTRPC Macro Example

For the client:

```
.....1.....+.....2.....+.....3.....+.....4.....+.....5.....+.....6.....+.....7..
      NTRPC  RPCSIZE=80,                                     *
              MAXBUFF=30,                                     *
              AUTORPC=ON,                                     *
              DFS=(MYSERV,MYNODE1,,ACI),                      *
              RDS=((SRVX,NODEX),(SRVY,NODEY))
```

For the server:

```
.....1.....+.....2.....+.....3.....+.....4.....+.....5.....+.....6.....+.....7..
      NTRPC  RPCSIZE=80,                                     *
              MAXBUFF=30,                                     *
              SRVNAME=MYSERV,                                 *
              SRVNODE=MYNODE,                                 *
              SERVER=ON
```

For additional information on Natural RPC, see the *Natural Remote Procedure Call (RPC)* document-
ation.

207

RPCSDIR - Library for Service Directory

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

It specifies the name of the Natural library (or one of its steplib) used by the client at runtime. This parameter is evaluated by the [SYSRPC](#) utility functions Service Directory Maintenance and Server Command Execution.

`RPCSDIR` is specified on the client side only.

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

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

208

RPCSIZE - Size of Buffer Used by Natural RPC

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

It specifies the size of the buffer used by Natural RPC. If the specified size is not large enough, the buffer will be increased on request.

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

Possible settings	1 - 2097151	Buffer size in KB.
	0	Natural RPC cannot be used.
Default setting	0	
Dynamic specification	yes	
Specification within session	no	

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

209

RPCUCT - Translate Subprogram Name into Upper

Case

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

This parameter specifies whether or not the Natural RPC server will translate the name of the remote CALLNAT to be executed into upper case.

RPCUCT is specified on the server side only.

Possible settings	ON	The name of the remote CALLNAT to be executed by the Natural RPC server will be translated into upper case before the CALLNAT is invoked. With this option, non-Natural RPC clients are supported that use mixed case characters in the subprogram names. Note: On UNIX, OpenVMS and Windows platforms, an implicit upper case translation is already done by Natural itself. RPCUCT=ON is therefore the compatibility mode for Natural RPC servers on mainframes and Natural RPC servers on UNIX, OpenVMS and Windows platforms.
	OFF	The name of the remote CALLNAT to be executed by the Natural RPC server is not changed. If the name contains lower case characters a NAT00082 error message is to be expected.
Default setting	OFF	
Dynamic specification	yes	
Specification within session	no	

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

210

RUNSIZE - Size of Runtime Buffer

This Natural profile parameter specifies the size of the Natural runtime buffer.

Alternatively, you can use the equivalent Natural profile parameter [DS](#) or macro [NTDS](#); see *Using Optional Macros in a Natural Parameter Module* in the *Operations* documentation to specify `RUNSIZE`.

Possible settings	10 - 64	Buffer size in KB.
Default setting	16	
Dynamic specification	yes	
Specification within session	no	

The Natural runtime buffer contains information on the following items:

- defined `STEPLIBS`,
- the file translation table (profile parameter [TF](#)),
- log information of the most recent command,
- the environment stack (for user settings),
- active global data areas,
- invoked subroutines (subroutine name and object name),
- invoked objects (address in the buffer pool for a fast location).

If the specified size of the runtime buffer is exceeded by a Natural user, the size for the invoked objects information is decreased accordingly. However, when this size decreases, the number of possible buffer-pool fast locations decreases, too; if it is about to become 0, an error message is issued.

211

SA - Sound Terminal Alarm

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

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

Possible settings	ON	The terminal alarm is sounded each time the user is prompted for input by Natural. Note: The use of this feature requires that the terminal alarm hardware feature has been installed for the terminal.	
	OFF	No terminal alarm is used for input prompting, however, the alarm may still be activated with the <code>SOUND ALARM</code> option of the <code>REINPUT</code> statement.	
Default setting	OFF		
Dynamic specification	yes		
Specification within session	yes	Applicable Statements:	SET GLOBALS
		Applicable Command:	GLOBALS
Application Programming Interface	USR1005N	See <i>SYSEXT - Natural Application Programming Interfaces</i> in the <i>Utilities</i> documentation.	

■ Syntactical Considerations	526
■ Runtime Considerations	527
■ Features	528
■ Restrictions	530

Selection boxes in an `INPUT` statement are available on mainframe computers only. For other platforms, selection boxes may be defined in the map editor only.

Selection boxes can be attached to input fields. They are a comfortable alternative to help routines attached to fields, since you can code a selection box direct in your program. You do not need an extra program as with help routines.

You may define a selection box clause for every `INPUT` variable of type alpha, regardless if this field is an input or output field, or both.

The syntax is:

SB=operand1 [,operand1]...

where *operand1* represents a value operand which is used to fill up the selection box with items.

Operand	Possible Structure					Possible Formats												Referencing Permitted	Dynamic Definition
operand1	C	S	A			A												yes	no

With SB, you specify the values to be displayed within the selection box.

To assign a selection box to a field:

Specify the attribute SB for an alpha `INPUT` field in your Natural program using the following example syntax:

INPUT #FLD (SB='value1', #ITEM1, #ITEM2(1:3), #ITEM3(*))

The following topics are covered below:

Syntactical Considerations

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 (mainframe only)	defined with a <code>CONST</code> clause of <code>DEFINE DATA</code> 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=0`. If you use `AD=0`, 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.

Features

How a Selection Box is Displayed

For a field with attached selection box, an indicator “V” is displayed next to the field.

Invoking Selection Boxes

To open a selection box:

There are two ways to open a selection box:

- Enter a question mark (?) in the V-field and press ENTER.

Or position the cursor on the V-field and press the help key (for example, PF1), if you have defined it to do so. See the next section for more details.

To define a help key to invoke the selection box:

You can define a help-key (for example, PF1) to make invoking the selection box much simpler.

This is done by adding the following line of code to your program:

```
SET KEY PF1=HELP
```

Scrolling in a Selection Box

There are two ways to scroll in a selection box:

- By putting the cursor on the `MORE` line and pressing `ENTER`.
- Or by using the terminal commands `%W-` and `%W+` assigned to PF-keys (for example, `PF7/PF8`).

Selecting a Value in a Selection Box

A value is selected from the selection box and copied into the field by putting the cursor on the value and pressing `ENTER`.

Duplicate Lines in a Selection Box

Lines with the same content which directly follow each other are suppressed.

For example, the following code

```
INPUT #FLD (SB='123', '456', 'XYZ', 'XYZ', 'XYZ', 'ABC', 'DEF')
```

produces the following output to the selection box:

```
123
456
XYZ
ABC
DEF
```

In the preceding example, `XYZ` is only displayed once. The other occurrences are considered redundant since they directly follow one another.

However, this line of code

```
INPUT #FLD (SB='123', 'XYZ', '456', 'XYZ', 'ABC', 'XYZ', 'DEF')
```

produces the following output to the selection box:

```
123
XYZ
456
XYZ
ABC
XYZ
DEF
```

In this case, all three occurrences of `XYZ` are displayed, since they do not directly follow one another.

Blank Lines in Selection Boxes

A blank line is only displayed the first time it appears; all subsequent blank lines are suppressed.

Restrictions

The number of operands in the `SB` clause is limited to 20.

The maximum number of values in a selection box is 248. When that limit has been reached, further values are not displayed. No error message is issued when the limit has been exceeded.

213

SCTAB - Scanner Characters

▪ SCTAB Parameter Syntax	532
▪ NTSCTAB Macro Syntax	533
▪ Example of NTSCTAB Macro	533
▪ Example of SCTAB Parameter	534

This Natural profile parameter allows you to overwrite the definitions in the scanner character-type table `NTSCTAB` as contained in the configuration module `NATCONFIG`. The `NTSCTAB` table defines the properties of characters

- used in mask definitions for the `MASK` function,
- recognized as delimiters in the `EXAMINE` and `SEPARATE` statements.

`SCTAB` corresponds to the `NTSCTAB` macro in the Natural parameter module `NATPARM`.

Possible settings	See <i>SCTAB Parameter Syntax</i> below.	
Default setting	As specified within the macro <code>NTSCTAB</code> in <code>NATCONFIG</code> .	
Dynamic specification	yes	This parameter can only be specified dynamically. In the Natural parameter module <code>NATPARM</code> , the macro <code>NTSCTAB</code> must be used instead.
Specification within session	no	

The following topics are covered below:

SCTAB Parameter Syntax

The `SCTAB` parameter is specified as follows:

SCTAB=(**character**1,**attribute-type**1,attribute-type2,...,*character*2,*attribute-type*1,*attribute-type*2,...)

character

You specify a character, and after it its attribute type(s).

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

attribute-type(s)

Attribute types can be:

UPPER	upper-case alphabetical
LOWER	lower-case alphabetical
NUM	numeric
HEX	hexadecimal
ALFANUM	alphanumeric
SPECIAL	special
NDELIM	non-delimiter

It is possible to specify more than one character in the list of values. You must enclose the entire string of character/attribute pairs in parentheses.

NTSCTAB Macro Syntax

The NTSCTAB macro is specified as follows:

```
NTSCTAB character1,attribute-type1,attribute-type2,...
NTSCTAB character2,attribute-type1,attribute-type2,... ..
```

For each character to be overwritten, you have to specify a separate NTSCTAB macro.

Example of NTSCTAB Macro

```
NTSCTAB 5E,LOWER,NDELIM
NTSCTAB 'B',SPECIAL
NTSCTAB 7B,SPECIAL
NTSCTAB 'Ä',UPPER,NDELIM
```

Example of SCTAB Parameter

```
SCTAB=(5E,LOWER,NDELIM,'B',SPECIAL,7B,SPECIAL,'Ä',UPPER,NDELIM)
```

214

SENDER - Screen Output Destination for Asynchronous

Processing

This Natural profile parameter only applies to Natural under CICS, Com-plete, IMS TM and UTM.

It specifies the destination where output from an asynchronous application is to be displayed. The destination specified applies to hardcopy output and primary reports.

Possible settings	1 to 8 characters	Output destination, for example, printer.
Default setting	none	
Dynamic specification	yes	
Specification within session	no	

Any additional reports are sent to the destinations specified with the `DEFINE PRINTER` statement (just as in a synchronous online session).

The following platform-specific characteristics apply:

Platform:	Comment:
CICS	<p>The profile parameter <code>SENDER</code> specifies the CICS transient data (TD) destination and the terminal or printer for terminal output from asynchronous sessions. If the specified destination does not exist, the session output is sent to the specified terminal or printer. If the specified terminal or printer does not exist either, the session terminates abnormally.</p> <p>The default terminal output format for asynchronous sessions is a 3270 data stream. If the <code>SENDER</code> terminal specification is not a 3270 device, the Natural application must switch to the correct terminal type before the first output statement (for example, by specifying <code>SET CONTROL 'T=PRNT'</code> for a printer or by starting with profile parameter <code>TTYTYPE=PRNT</code>).</p> <p>If you are routing all output to a (spool) destination, such as CSSL, the Natural application must be switched to line mode, for example, by specifying <code>SET CONTROL 'T=xxxx'</code> or by starting</p>

Platform:	Comment:
	<p>with profile parameter <code>TTYE=xxxx</code>, where <code>xxxx</code> is <code>BTCH</code> or <code>ASYL</code>. In this case, two other profile parameters are relevant: <code>EJ</code> and <code>INTENS</code>.</p> <p>If you set <code>EJ=ON</code>, all lines are routed with a leading ASA control character.</p> <p>With <code>EJ=OFF</code>, there is no leading ASA control character. <code>INTENS</code> should be set to 1, particularly if you have set <code>EJ=OFF</code>.</p> <p>For further CICS-specific functionality, see <i>Asynchronous Natural Processing under CICS</i> in the <i>TP Monitor Interfaces</i> documentation.</p>
Com-plete	See <i>Asynchronous Natural Processing under Com-plete/SMARTS</i> .
IMS TM	The profile parameter <code>SENDER</code> specifies the default <code>LTERM</code> . This <code>LTERM</code> is always used when no other printer has been specified. You should always dynamically define a <code>SENDER</code> parameter in the <code>NIIBOOT</code> module. This is important when Natural tries to output error messages when starting a session: if no <code>SENDER</code> parameter is specified, there is no valid <code>LTERM</code> and <code>NATIMS</code> terminates the session.
open UTM	<p>The profile parameter <code>SENDER</code> specifies the ID for the initialization of an asynchronous transaction; that is, the ID which identifies the transaction as asynchronous. If output from an asynchronous transaction is to be printed, the setting specified with the <code>SENDER</code> parameter also identifies the printer on which the output is to be printed.</p> <p>For further <i>open UTM</i>-specific functionality, see <i>Asynchronous Transaction Processing under UTM</i> in the <i>TP Monitor Interfaces</i> documentation.</p>

For further information, see also the profile parameter [OUTDEST](#) and Asynchronous Processing in the *Operations* documentation.

215

SERVER - Start Natural Session as an RPC Server

Session

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

It specifies whether or not the Natural session will be started as an RPC server session.

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

Possible settings	ON	The Natural session will be started as an RPC server session.
	OFF	The Natural session will not be started as an RPC server session.
Default setting	OFF	
Dynamic specification	yes	
Specification within session	no	

For further information, see the *Natural Remote Procedure Call (RPC)* documentation.

216

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.

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

Possible settings	1 - 30	Number of spaces. The <code>SF</code> 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 <i>SYSEXT - Natural Application Programming Interfaces</i> in the <i>Utilities</i> documentation.	

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

See also *Column Spacing - SF Parameter and nX Notation* in the *Programming Guide*.

217

SG - Sign Position

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

If the [EM](#) (edit mode) parameter is specified, it overrides the SG parameter.

Possible settings	ON	A sign position will be allocated.	
	OFF	No sign position will be allocated. SG=OFF causes numeric fields with negative values to be output without a minus (-) sign. 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

Example:

```
FORMAT SG=OFF
```

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

218

SI - Shift-In Code for Double-Byte Character Set

This Natural profile parameter is only relevant for Asian countries which use double-byte character sets (DBCS). The parameter is used to specify a shift-in code.



Caution: Note that the profile parameter `SI` will no longer be available with the next release of Natural. The functionality of the profile parameter `SI` is covered by the profile parameter [SOSI](#).

Possible settings	0F	Shift-in code for IBM hardware.
	29	Shift-in code for Fujitsu hardware.
Default setting	none	
Dynamic specification	yes	
Specification within session	no	

The shift-in code is used to indicate the point at which the code of character representation is shifted from double-byte mode back into normal (single-byte) mode. The beginning of the double-byte character representation (shift-out code) is indicated by the setting defined with the profile parameter [S0](#).

219

SKEY - Storage Protection Key

This Natural profile parameter only applies under Com-plete.

It determines whether Natural runs under the same storage protection key as Com-plete.

Possible settings	ON	Natural runs under the same storage protection key as Com-plete.
	OFF	Natural runs under a different storage protection key than Com-plete.
Default setting	ON	
Dynamic specification	yes	
Specification within session	no	

See the Com-plete documentation for details on storage protection keys.

220

SL - Source Line Length

This Natural profile and session parameter specifies the number of characters to be interpreted on each Natural source code line. This also applies to the line mode editor which is activated with the system command EDT.

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

Possible settings	20 - 250	In batch mode:	The number of characters to be processed on each line in the datasets CMSYNIN and CMOBJIN. For details on these datasets, refer to the operating-system-specific parts of the section <i>Natural in Batch Mode</i> in the <i>Operations</i> documentation.
		In online mode:	The number of characters to be interpreted when using the Natural program editor in EDT mode (as activated with the system command EDT).
Default setting	72		
Dynamic specification	yes		
Specification within session	yes	Applicable Statements:	none
		Applicable Command:	GLOBALS
Application Programming Interface	USR1005N	See SYSEXT - <i>Natural Application Programming Interfaces</i> in the <i>Utilities</i> documentation.	

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

221

SLOCK - Source Locking

This Natural profile parameter is used to specify how concurrent updates of Natural source members are to be handled (see also *Locking of Source Objects* in the *Editors* documentation).

Possible settings	PRE	<p>Activates locking of source objects that are edited either locally or in a SPoD environment, or using Natural ISPF, or in mixed environments.</p> <p>This is the recommended setting when working in mixed environments.</p> <p>In order to lock a source member against concurrent updates, a specific record is written to the Natural system file <code>FUSER</code> or <code>FNAT</code> (depending on where the source member to be edited is located).</p>
	SPoD	<p>Locking of source objects occurs only in a remote development environment basing on Natural Single Point of Development (SPoD). This setting provides compatibility with previous Natural versions that supported locking under SPoD.</p> <p>In order to lock a source member against concurrent updates, a specific record is written to the Development Server File (FDIC) system file.</p>
	POST	<p>When setting <code>SLOCK=POST</code>, the source object which is being edited can be read into the source work area and modified by multiple users. However, only the user who saves a modification first can update the source object. This is done by comparing the time stamp of the source object stored in the database with the time stamp of the source object when it is read into the source work area. All other users receive appropriate error messages when trying to save the source. This is not compatible with the SPoD locking concept of previous Natural versions.</p>
	OFF	Deactivates all locking mechanisms.
Default setting	SPoD	
Dynamic specification	yes	
Specification within session	no	

222

SM - Programming in Structured Mode

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

Within a Natural session, the profile parameter setting `SM=OFF` can be overridden by the session parameter `SM=ON`.

Possible settings	ON	Forces the use of structured mode syntax. Note: If structured mode (<code>SM=ON</code>) is specified by profile parameter <code>SM</code> , an attempt to change this setting with system command <code>GLOBALS</code> and session parameter <code>SM</code> will be rejected (Reporting mode not permitted).	
	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 <i>SYSEXT - Natural Application Programming Interfaces</i> in the <i>Utilities</i> documentation.	

If Natural Security is installed:

- the setting of the mode option in the library's security profile determines whether the `SM` profile parameter can be used; see also *Programming mode* in the *Natural Security* documentation.
- this parameter may be disabled by Natural Security to the effect that structured mode is invariably in effect for a given library.

223

SO - Shift-Out Code for Double-Byte Character Set

This Natural profile parameter is only relevant for Asian countries which use double-byte character sets (DBCS). The parameter is used to specify a shift-out code.



Caution: Note that the profile parameter `SO` will no longer be available with the next release of Natural. The functionality of the profile parameter `SO` is covered by the profile parameter [SOSI](#).

Possible settings	0E	Shift-out code for IBM hardware.
	28	Shift-out code for Fujitsu hardware.
Default setting	none	
Dynamic specification	yes	
Specification within session	no	

The shift-out code is used to indicate the point at which the code of character representation is shifted out of normal (single-byte) mode into double-byte mode. The end of the double-byte character representation (shift-in code) is indicated by the setting defined with the profile parameter [SI](#).

224

SORT - Control of Sort Program

■ SORT Parameter Syntax	556
■ NTSORT Macro Syntax	556
■ Keyword Subparameters	557

This Natural profile parameter is used to control the sort program used for the processing of SORT statements. It corresponds to the [NTSORT](#) macro in the Natural parameter module NATPARM.

SORT or NTSORT can be used to specify various options that control the handling of the sort program used when a SORT statement is executed.

The sort program to be used can be either Natural's internal one (the default for all environments) or an external one. The type of sort to be used depends on the setting of the keyword subparameter [EXT](#).

Possible settings		For an explanation of the individual options and their possible settings, see SORT Parameter Syntax below.
Default setting		WRKSIZE=10 , STORAGE=MAIN , EXT=OFF , EXTNAME=Sort , EXTEOJ=OFF
Dynamic specification	yes	This parameter can only be specified dynamically. In the Natural parameter module NATPARM, the macro NTSORT must be used instead.
Specification within session	no	

The following topics are covered below:

SORT Parameter Syntax

The SORT parameter is specified as follows:

SORT=([WRKSIZE](#)=*nnn*,[STORAGE](#)=*medium*,[EXT](#)=ON/OFF,[EXTNAME](#)=*name*,[EXTOPT](#)=(*options*,...),[EXTEOJ](#)=ON/OFF)

NTSORT Macro Syntax

The NTSORT macro is specified as follows:

NTSORT
[WRKSIZE](#)=*nnn*,[STORAGE](#)=*medium*,[EXT](#)=ON/OFF,[EXTNAME](#)=*name*,[EXTOPT](#)=(*options*,...),[EXTEOJ](#)=ON/OFF

The individual keyword subparameters are explained below.

Keyword Subparameters

The following keyword subparameters are available:

[WRKSIZE](#) | [STORAGE](#) | [EXT](#) | [EXTNAME](#) | [EXTOPT](#) | [EXTEOJ](#)

WRKSIZE - Size of Work Buffer Used by Sort Program

WRKSIZE specifies the size *nnnnnnn* (in KB) of the work buffer used by the sort program.

Possible values:	10 to 2097151, or 0.
Default value:	10

If you specify WRKSIZE=0, no sort operations can be performed.

The work buffer specified by WRKSIZE accommodates internal sort control data. The remaining storage is used to collect and sort the records. The size of the sort control data depends on various factors (the WRKSIZE itself, the sort record length, the number of sort keys, their size and format) and can therefore not be calculated in a formal way.

STORAGE - Type of Storage Medium

STORAGE specifies the type of storage *medium* to be used by Natural's internal sort program.

In general, the SORT statement first tries to use the remaining storage in WRKSIZE. If the number of records does exceed this storage, the SORT statement tries to use intermediate storage to additionally process records:

MAIN	Only the remaining storage of WRKSIZE is used, no other intermediate storage is available. This is the default setting.
BP	The SORT buffer pool is used as intermediate storage. See note below.
SD	SD files are used as intermediate storage. This value is honored under Com-plete only.
SMARTS	SMARTS portable file system is used.

Notes Concerning SD Files under Com-plete/SMARTS

- The files are allocated as temporary SD files. They are allocated for a stack level. This means, the name syntax of the SORT SD files is `&&ST snn` with:

<code>&&</code>	Indicator for a temporary SD file
<code>ST</code>	Standard prefix for the SD sort file
<code>s</code>	Stack level
<code>snn</code>	Sequence number within a single SORT run

- SMARTS work files are located in the SMARTS Portable File System. The path must be specified with the SMARTS environment variable `$NAT_WORK_ROOT`. A special directory named SORT for SORT workfiles is added and for each user, a different directory is created named by the user ID. The resulting directory is then `$NAT_WORK_ROOT/sort/userid`. The naming of the SORT work files corresponds to the SD files under Com-plete.

Note Concerning Usage of Sort Buffer Pool

If you want to use a sort buffer pool, define the SORT keyword subparameter `STORAGE=BP` to indicate that a sort buffer pool is to be used for any additional storage beyond the defined [WRKSIZE](#). Simultaneously, use the profile parameter `BPI` or the parameter macro `NTBPI` to make a buffer pool of `TYPE=``SORT` and `NAME=`*name* known to Natural, for example: `BPI=(TYPE=``SORT``, NAME=``XYZ``)`. When a name is specified with the `BPI` keyword subparameter `NAME`, reference is made to a global sort buffer pool, whereas a local sort buffer pool can be specified by `NAME=` ' ' (blank).

EXT - Use of External Sort Program

EXT specifies if an external sort program is to be used or not:

ON	An external sort program will be used. The use of an external sort program is possible only in batch environments, including IMS/BMP, TSO, TIAM and CMS.
OFF	The Natural SORT program will be used (this is the default).

EXTNAME - Name of External Sort Program

This subparameter does not apply under BS2000/OSD.

EXTNAME specifies the *name* (1 to 8 characters) of the external sort program to be used. The default name is SORT.

EXTOPT - Additional Options for External Sort Program

This subparameter does not apply under BS2000/OSD.

EXTOPT specifies additional *options* for the external sort program.

Natural generates the necessary field and format parameters and passes them to the external sort program. With EXTOPT, you can specify additional parameters to be passed to the external sort program. You can only specify parameters that are part of the control statement syntax of your external sort program.

You can specify up to two option strings which are delimited by a slash (/). The first option string is appended to the SORT control statement, the second option string is used to build an OPTION control statement. You may omit the option string before or after the slash. If the option string after the slash is omitted no OPTION control statement is generated at all.

The whole option string must be enclosed in single quotes ('...'). For compatibility reasons, it is still possible to have the option string enclosed in brackets instead.

For compatibility reasons, a single option string without a leading or trailing slash is handled differently. Depending on the underlying operating system, the options are appended to the following control statements:

z/OS and VM/CMS:	SORT control statement
z/VSE:	OPTION control statement

EXTOPT Example:

The additional parameters can be specified as in the following example:

```
EXTOPT=(SIZE=E2000000,NOEQUALS,DYNALLOC=(3350,8))
EXTOPT='SIZE=E2000000,NOEQUALS,DYNALLOC=(3350,8) '
EXTOPT='SIZE=E2000000,NOEQUALS,DYNALLOC=(3350,8)/NOCHECK '
EXTOPT='/NOCHECK '
EXTOPT='WORK=4/ '
```

EXTEOJ - Action in the Event of an Error

EXTEOJ specifies the action to be taken if an error is detected during the execution of the external sort program:

ON	If an error is detected, SORT processing is terminated. ON requires that the SORT program used is able to detect a return code of 16 from both the E15 and E35 SORT exit routines.
OFF	If an error is detected, Natural withholds further calls to the sort program and ignores each record as it is passed to the E35 SORT exit routine (this is the default).

225 SOSI - Shift-Out/Shift-In Codes for Double-Byte

Character Set

▪ SOSI Parameter Syntax	562
▪ Positional Subparameters	563
▪ Conversion of Logical Shift-Out/Shift-In Characters	564
▪ Automatic Adaptation of Translation Tables	564
▪ Compatibility of SOSI Profile Parameter and Obsolete SO and SI Profile Parameters	564
▪ SOSI Parameter Examples	565

This Natural profile parameter is relevant for Asian countries which use double-byte character sets (DBCS).

SOSI replaces the profile parameters [SI](#) and [SO](#) which will cease to be available with the next version of Natural.

If the profile parameter [CP](#) is set to a multi-byte code page (MBCS), the logical shift-in and shift-out characters will be supplied with the code page and therefore SOSI will be ignored.

Possible settings	subparameters	See Positional Subparameters below.
Default setting	none	
Dynamic specification	yes	
Specification within session	no	

The following topics are covered below:

SOSI Parameter Syntax

The parameter syntax of SOSI is as follows:

SOSI=([logical-shift-out](#),[[physical-shift-out](#)],[logical-shift-in](#),
[[physical-shift-in](#)],[[SO/SI-display-length](#)])

A shift-out code is used to indicate the point at which the code of character representation is shifted out of normal (single-byte) mode into double-byte mode.

A shift-in code is used to indicate the point at which the code of character representation is shifted from double-byte mode back into normal (single-byte) mode.

Positional Subparameters

The positional subparameters are described below:

<i>logical-shift-out</i>	<p>The logical shift-out character must be a single character. Specify the hexadecimal representation of the logical shift-out character.</p> <p>Usually, the value 0E is used for IBM hardware and the value 28 is used for Fujitsu hardware.</p>
<i>physical-shift-out</i>	<p>The value of the physical shift-out character must be chosen depending on the screen hardware that is used.</p> <p>The length of the physical shift-out character may be one or two bytes. Specify the hexadecimal representation of the physical shift-out character.</p> <p>The default value is the logical shift-out character.</p>
<i>logical-shift-in</i>	<p>The logical shift-in character must be a single character. Specify the hexadecimal representation of the logical shift-in character.</p> <p>Usually, the value 0F is used for IBM hardware and the value 29 is used for Fujitsu hardware.</p>
<i>physical-shift-in</i>	<p>The value of the physical shift-in character must be chosen depending on the screen hardware that is used.</p> <p>The length of the physical shift-in character may be one or two bytes. Specify the hexadecimal representation of the physical shift-in character.</p> <p>The default value is the logical shift-in character.</p>
<i>SO/SI-display-length</i>	<p>The number of bytes occupied on the screen by the physical shift-out/shift-in characters.</p> <p>Possible values are 0 and 1. The default value is 1.</p> <p>For IBM hardware, the value 1 must be used. For Fujitsu hardware, the value 0 must be used.</p>

Conversion of Logical Shift-Out/Shift-In Characters

Logical shift-out/shift-in characters are converted into the corresponding physical shift-out/shift-in characters before data is transferred to the screen.

Physical shift-out/shift-in characters are converted into the corresponding logical shift-out/shift-in characters before data entered on the screen is transferred to the Natural application.

Automatic Adaptation of Translation Tables

If code page support is disabled (that is, the profile parameter `CP` is set to `CP=OFF`), the entries for the logical shift-out/shift-in characters are updated in the translation tables provided by the following macros and profile parameters:

Table	Macro	Profile Parameter
Standard (primary) output translation table	<code>NTTAB</code>	<code>TAB</code>
Alternative (secondary) output translation table	<code>NTTAB1</code>	<code>TAB1</code>
Secondary input translation table used when the session parameter <code>PM</code> is set to <code>C</code> .	<code>NTTAB2</code>	<code>TAB2</code>
<code>SYS*</code> output translation table	<code>NTTABL</code>	<code>TABL</code>

If the characters into which the logical shift-out/shift-in characters are to be translated still have their default value (`? = X'6F'`) in the respective translation table at Natural startup (that is, they have not been modified by one of the macros or profile parameters mentioned above), they will be updated so that logical shift-out/shift-in characters will not be not translated for input and output.

For detailed information on the translation tables, see *Translation Tables* in the *Operations* documentation.

Compatibility of SOSI Profile Parameter and Obsolete SO and SI Profile Parameters

The subparameter logical-shift-out corresponds to the profile parameter `S0` and the subparameter logical-shift-in corresponds to the profile parameter `SI`.

The obsolete profile parameters `S0` and `SI` are still valid, but must be used mutually exclusive with the profile parameter `SOSI`. It is strongly recommended that you use the `SOSI` profile parameter instead of the profile parameters `S0` and `SI`.

Specifying $S0=xx, SI=yy$ is equivalent to specifying $SOSI=(xx, xx, yy, yy, 1)$.

SOSI Parameter Examples

For IBM hardware, you should use $SOSI=(0E, 0E, 0F, 0F, 1)$, which is equivalent to $SOSI=(0E, , 0F, , 1)$.

For Fujitsu hardware, you should use $SOSI=(28, 28, 29, 29, 0)$, which is equivalent to $SOSI=(28, , 29, , 0)$.

To execute an application that has been created for IBM hardware (with the parameter setting $SOSI=(0E, 0E, 0F, 0F, 1)$ applied) on Fujitsu hardware without changing the application, use $SOSI=(0E, 4028, 0F, 2940, 1)$.

226

SRETAIN - Retain Source Format

This Natural profile parameter specifies the encoding format for new and existing Natural sources when they are saved.

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

Possible settings	ON	<p>The original code page of an existing Natural source is retained.</p> <p>If an existing Natural source without code page information is saved, it will not receive any code page information.</p> <p>When a new Natural source is created, it will be saved in the default code page format as defined with the profile parameter CP.</p>
	OFF	Natural sources will be saved in the default code page format.
	(ON,EXCEPTNEW)	<p>The original code page of an existing Natural source is retained.</p> <p>If an existing Natural source without code page information is saved, it will not receive any code page information.</p> <p>When a new Natural source is created, it will be saved without code page information.</p> <p>(ON,EXCEPTNEW) retains the compatibility of newly created Natural sources with existing applications that have been created with earlier Natural versions that did not provide code page support.</p> <p>Note: The value (ON,EXCEPTNEW) is supported on mainframe computers only.</p>
Default setting	ON	
Dynamic specification	yes	
Specification within session	no	

For the code page of a Natural source that is saved or stowed, the resulting encoding depends on the settings of the profile parameters SRETAIN and CP. See *Code Page Support for Editors, System Commands and Utilities on the Mainframe* in the *Unicode and Code Page Support* documentation.

227

SRVCMIT - Server Commit Time

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

It specifies the time at which a Natural RPC server automatically commits an RPC conversation or a non-conversational RPC request. This parameter is only evaluated if the profile parameter [ETEOP](#) is set to ON.

SRVCMIT is specified on the server side only.

Possible settings	B	The Natural RPC server automatically commits a database transaction before the reply is sent to the client. 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. If the reply fails, the database transaction is rolled back.
Default setting	B	
Dynamic specification	yes	
Specification within session	no	

For further information, see the *Natural Remote Procedure Call (RPC)* documentation.

228

SRVNAME - Name of RPC Server

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

It specifies the name of the RPC server, with which it registers on the node specified with the profile parameter [SRVNODE](#).

SRVNAME is specified on the server side only.

Possible settings	1 - 192 characters	Valid server name.
Default setting	none	
Dynamic specification	yes	
Specification within session	no	

You may either specify a physical server name of up to 32 characters or a logical service name of up to 192 characters. In case of a logical service name, an asterisk (*) must be specified with the [SRVNODE](#) parameter(intentionally left empty).

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 broker attribute file, as below:

```
CLASS=RPC, SERVICE=CALLNAT, SERVER=srvname
```

Example:

```
SRVNAME='PRODUCTION_SERVER'      /* physical server name */  
SRVNAME='MY_LOGICAL_SERVICE,MY_SET' /* logical server name */
```

For more details about Location Transparency and logical service names, refer to the EntireX documentation.

For further information, see the *Natural Remote Procedure Call (RPC)* documentation.

229

SRVNODE - Name of Node

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

It specifies the name of the node upon which an RPC server registers.

SRVNODE is specified on the server side only.

Possible settings	1 - 192 characters	Node name.
Default setting	none	
Dynamic specification	yes	
Specification within session	no	

You may either specify a physical node name of up to 32 characters, a logical node name of up to 192 characters or an asterisk (*) (intentionally left empty) to indicate that [SRVNAME](#) contains a logical service name.

In case of an EntireX Broker node, a physical node name may refer to an Entire Net-Work node or to a TCP/IP address. Note that the broker stub in use must support the naming notation. For details about the structure of node names and their support by the broker stubs, refer to the EntireX documentation.

The examples below are based on the EntireX notation.

```
SRVNODE=ETB001                /* Entire Net-Work node */
SRVNODE=PCBROKER              /* host name for a TCP/IP address */
SRVNODE='157.189.160.95:1958:TCP' /* TCP/IP address with port number */
SRVNODE='tcpip://host.com:1958' /* host name for a TCP/IP address
with port number */
SRVNODE='LOGBROKER=MY_LOGICAL_NODE,MY_SET' /* logical node name */
SRVNODE='*'                  /* logical service name in SRVNAME */
```

If a host name is used for the TCP/IP address, the name must either be known to your DNS server or it must be defined in the hosts file of your TCP/IP configuration.

If the port number is 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.

For more details about Location Transparency and logical node names, refer to the EntireX documentation.

For further information, see the *Natural Remote Procedure Call (RPC)* documentation.

230

SRVTERM - Server Termination Event

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

It specifies the event at which a Natural RPC server is automatically terminated.

SRVTERM is specified on the server side only.

Possible settings	NEVER	A Natural RPC server is never automatically terminated. To terminate a Natural RPC server refer to <i>Terminating a Natural RPC Server</i> and <i>Terminating an EntireX Broker Service</i> in the <i>Natural Remote Procedure Call (RPC)</i> 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. 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	

For further information, see the *Natural Remote Procedure Call (RPC)* documentation.

231

SRVUSER - User ID for RPC Server Registry

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

It specifies the user ID needed to register an RPC server on the node specified with the profile parameter [SVRNODE](#).

In case of an EntireX Broker node, [SRVUSER](#) is also used to logon to the EntireX Broker. A password is either taken from Natural Security (see [*NSC](#) below) or specified via the application programming interface [USR2072N](#).

[SRVUSER](#) is specified on the server side only.

Possible settings	<i>user-ID</i>	Valid user ID, *USER or *NSC. 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	<i>timestamp</i>	If the user ID is omitted, the timestamp will be used.
Dynamic specification	yes	
Specification within session	no	

For further information, see the *Natural Remote Procedure Call (RPC)* documentation.

232

SRVWAIT - Wait Time of RPC Server

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

It specifies the number of seconds the server is to wait for an RPC client request. If this time is exceeded, the RPC server is informed by the transport layer. The RPC server writes a corresponding message to the Natural RPC server trace file and continues to wait for an RPC client request.

SRVWAIT is specified on the server side only.

Possible settings	0 - 32767	Wait time in seconds.
Default setting	0	Unlimited wait time.
Dynamic specification	yes	
Specification within session	no	

For further information, see the *Natural Remote Procedure Call (RPC)* documentation.

233

SSIZE - Size of Source Area Allocated by the Editors

This Natural profile parameter determines the size of the buffer used by the Software AG Editor.

Alternatively, you can use the equivalent Natural profile parameter [DS](#) or the macro [NTDS](#) to specify the `SSIZE` value. See *Using Optional Macros in a Natural Parameter Module* in the *Operations* documentation.

Possible settings	40 - 512	Buffer size in KB.
	0	If <code>SSIZE=0</code> or if the required space is not available, the Software AG Editor cannot be used.
Default setting	64	
Dynamic specification	yes	
Specification within session	no	

If you have defined an Editor work file with a record length greater than 4 KB (default), you should use an `SSIZE` value greater than 64 KB. There are two work file record buffers allocated within the `SSIZE`. Therefore you should add two times your work file record buffer size minus 4 KB to your `SSIZE`. Example: Your Editor work file has a record length of 10 KB. Then use at least `SSIZE=76` (that is, $64+2*(10-4)$).

For further information about the SAG Editor work file, see *Operating the Software AG Editor, Editor Work File* in the *Operations* documentation.

234

STACK - Place Data/Commands on the Stack

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



Note: If `STACK` is used, a colon (:) must be specified with the profile (or session) parameters `DC`, `HI`, `IA`, `ID` and `STACKD`.

Possible settings	any character string	See below.
Default setting	HELLO	
Dynamic specification	yes	
Specification within session	no	

The stack can contain a sequence of Natural commands and/or user-specified commands, together with their data, for execution at the beginning of the Natural session.

The command stack is processed before the user is prompted for input on the screen (TP mode) or data are read from `CMSYNIN/CMOBBIN` files; see *Natural in Batch Mode* in the *Operations* documentation.

If an `INPUT` statement is encountered during stack processing, the corresponding input screen is generated only if the required input data were not supplied with the command when the stack was created. Any reports generated during stack processing are displayed as usual.

Each system or user-defined command can be optionally followed by data which are used to satisfy requests for information required during the processing of the command. The character string provided as data for the `STACK` parameter must be enclosed in parentheses. If the command is a user command (that is, the name of a user program), any data provided resolve the data requirements of `INPUT` statements within the user program.

Conventions:

- Multiple settings for one `INPUT` statement are separated by a comma (,).
- Data for multiple `INPUT` statements are separated by a colon (:).

- Commands are separated by the stack delimiter character defined by profile parameter [STACKD](#). The default setting is a semicolon (;).

Examples:

```
STACK=(LOGON USER1;UCMD1 A,B;UCMD2 C,D:E;FIN)
STACK=OFF                                     No STACK data.
STACK=UCMND Execute command UCMND           No embedded blanks.
STACK=(CMD DATA:DATA;CMD...)               Place commands/data on stack.
```

Since some commands (for example, `GLOBALS`) do not read parameters by `INPUT`, a blank character should be used rather than a colon to delimit a command from the first parameter data element.

```
STACK='LOGON SYSTEM'
```

Because the macro assembler does not allow embedded blanks within parentheses, the character string must be enclosed in apostrophes when specified as static parameter.

235

STACKD - Stack Delimiter Character

This Natural profile parameter specifies the character to be used as the command delimiter for the `STACK` parameter and for command input under the Natural Development Server (product code: NDV) in a Natural Single Point of Development environment.

To avoid that the value specified for the `STACK` parameter or the data passed as command input under the Natural Development Server is not interpreted as intended, the `STACKD` parameter value should be set to a character that is not contained in the data passed if the data contains the default value of the stack delimiter character (see example below). The `STACKD` parameter should be changed to a character other than the default character if the `ID` parameter has been set to the semicolon. For downward compatibility reasons, this restriction does not apply to `STACKD=;` (the default setting).

Possible settings	any special character	The character must not be the same as the one specified with the <code>ID</code> profile/session parameter (input delimiter character), <code>DC</code> profile/session parameter (decimal character) or <code>IA</code> profile/session parameter (input assign character).
Default setting	; (semicolon)	
Dynamic specification	yes	
Specification within session	no	

The character specified may be enclosed with single quotes. If the input delimiter character is to be a comma, it must be specified as `ID= ' , '`, because the comma (,) separates individual parameters.

Example:

```
STACKD='/',ID=';' STACK=(DUMP IOB;+100/FIN)
```

To avoid that the semicolon after DUMP IOB is interpreted as a command delimiter, STACKD is set to '/ '.

236

STEPLIB - Additional Steplib Library

This Natural profile parameter specifies the name of an additional Natural steplib (concatenated library) to be used.

Possible settings	1 to 8 characters	Steplib name.
Default setting	SYSTEM	
Dynamic specification	yes	
Specification within session	no	
Application Programming Interface	USR1005N	See <i>SYSEXT - Natural Application Programming Interfaces</i> in the <i>Utilities</i> documentation.

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

237

SUBSID - Subsystem ID under z/OS and z/VSE

This Natural profile parameter is available under z/OS and z/VSE only. It identifies the Natural subsystem to be used.

Possible settings	1 to 4 characters	Natural subsystem. If you specify less than 4 characters, blanks will be appended so as to get a 4-byte setting.
Default setting	NAT4	
Dynamic specification	yes	
Specification within session	no	

For the purposes of the Natural CICS Interface (see `ROLLSRV`, `SIPSERV`, `SUBSID`), the Natural profile parameter `SUBSID` is ignored if it is specified in a parameter string by a profile parameter `SYS` or `PROFILE` or in an alternate parameter module (as specified with the profile parameter `PARM`).

For information on the Natural subsystem, see *Natural Subsystem under z/OS* or *Natural Subsystem under z/VSE* in the *Operations* documentation.

238

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 <i>SYSEXT - Natural Application Programming Interfaces</i> in the <i>Utilities</i> documentation.

The error transaction program is defined either with the profile parameter [ETA](#) or within the Natural Security library profile.

239

SYS - Define and Activate a Set of Dynamic Profile

Parameters

■ SYS Parameter Syntax	594
■ NTSYS Macro Syntax	595
■ Example of NTSYS Macro	595

This Natural profile parameter enables you to activate a set of dynamic profile parameters which is predefined in the Natural parameter module. This avoids the repeated specification of long sequences of profile parameters for the Natural session start. Alternatively, a similar functionality is provided by the profile parameter [PROFILE](#).

In the parameter module (NATPARM), you use [NTSYS](#) macros to predefine sets of dynamic profile parameters. You identify such a set of parameters by giving it a unique set name.

Possible settings	<i>set-name</i>	set-name (1 to 8 characters) defined by the NTSYS macro in the Natural parameter module (NATPARM).
Default setting	none	
Dynamic specification	yes	This parameter can only be specified dynamically.
Specification within session	no	

The specified parameter set must be defined in the Natural parameter module (NATPARM) currently active, e.g. in an alternative parameter module, if it is specified by the [PARM](#) parameter before the SYS parameter.

A parameter set is evaluated right in its position of SYS in the parameter string, as you would have included the defined parameter string at this position instead.

The following topics are covered below:

SYS Parameter Syntax

The parameter syntax of SYS is as follows:

SYS=[set-name](#)

NTSYS Macro Syntax

The NTSYS macro is specified for each set of parameters as follows:

```
NTSYS set-name,'parameter-string1','parameter-string2',...
```

Syntax Element Description:

<i>set-name</i>	The <i>set-name</i> identifies the subsequent set of parameters, it can be 1 to 8 characters long and must begin with an alphabetical character.
<i>parameter-string</i>	<p>After the <i>set-name</i>, you specify individual profile parameters and their values. For the profile parameters you can specify, see Profile Parameters in the <i>Parameter Reference</i> documentation.</p> <ul style="list-style-type: none"> ■ The entire set of parameters you specify with an NTSYS macro must constitute a valid string of dynamic parameters. The specified parameter string is not checked for validity during the NATPARM assembly. ■ If <i>parameter-string1</i> exceeds 255 characters, you must define a second parameter string <i>parameter-string2</i>, etc. ■ All parameter strings of one NTSYS macro are concatenated to one set of parameters. ■ An apostrophe within a substring is represented by two apostrophes.

Example of NTSYS Macro

```
NTSYS SET1,'FUSER=(,50),LC=ON,NC=ON,ULANG=2,TQ=OFF','STACK=(LOGON ULIB1)'
NTSYS SET2,'FUSER=(,51),ULANG=4,WH=ON,KC=ON,STACK=(LOGON ULIB2)'
```


240

SYSCIP - Adabas Cipher Key for Natural System Files

This Natural profile parameter only applies to Adabas databases. It provides a default Adabas cipher key for access to Natural system files (FNAT, FUSER, FDIC, FSEC, FSP00L) which have been loaded with the ciphered option.

Possible settings	8 characters	The cipher code specified with the SYSCIP parameter applies to all Natural system files for which no individual cipher codes are specified.
	blanks	If the Natural system files are not ciphered, set SYSCIP to blanks.
Default setting	blanks	
Dynamic specification	yes	If you specify the SYSCIP parameter dynamically in conjunction with any of the individual system file parameters FNAT , FUSER , FDIC , FSEC and FSP00L , you must specify the SYSCIP parameter <i>before</i> any individual system file parameter.
Specification within session	no	



Note: Cipher codes for individual system files can be specified with the parameters FNAT, FUSER, FDIC, FSEC and FSP00L.

241

SYSPSW - Adabas Password for Natural System Files

This Natural profile parameter only applies to Adabas databases.

It provides a default Adabas password for access to Natural system files (**FNAT**, **FUSER**, **FDIC**, **FSEC**, **FSP00L**) which have been password-protected.

Possible settings	8 characters	<p>If a Natural system file is password-protected, a password which permits updates to the file must be specified.</p> <p>The password specified with the SYSPSW parameter applies to all Natural system files for which no individual passwords are specified.</p> <p>If the OPRB parameter is specified, the SYSPSW password is used for the initial Adabas open call and must permit access and/or update to all the files specified in OPRB as required.</p>
	blanks	If the Natural system files are not password-protected, set SYSPSW to blanks.
Default setting	blanks	
Dynamic specification	yes	If you specify SYSPSW dynamically in conjunction with any of the individual system file parameters FNAT , FUSER , FDIC , FSEC and FSP00L , you must specify SYSPSW <i>before</i> any individual system file parameter.
Specification within session	no	



Note: Passwords for individual system files can be specified with the profile parameters **FNAT**, **FUSER**, **FDIC**, **FSEC** and **FSP00L**.

242

TAB - Standard Output Character Translation

▪ TAB Parameter Syntax	602
▪ NTTAB Syntax	602
▪ Example of NTTAB Macro	603
▪ Example of TAB Parameter	603

This Natural profile parameter allows you to overwrite the definitions in the translation table `NTTAB` as contained in the configuration module `NATCONFIG`. The `NTTAB` table is the standard output translation table.

TAB corresponds to the `NTTAB` macro in the Natural parameter module `NATPARM`.

Possible settings		See <i>TAB Parameter Syntax</i> below.
Default setting		As specified within the macro <code>NTTAB</code> in <code>NATCONFIG</code> .
Dynamic specification	yes	This parameter can only be specified dynamically. In the Natural parameter module <code>NATPARM</code> , the macro <code>NTTAB</code> must be used instead.
Specification within session	no	

The following topics are covered below:

TAB Parameter Syntax

The `TAB` parameter is specified as follows:

TAB=(a1,a2,b1,b2,c1,c2,...)

You specify pairs of characters, the first character of a pair being the character to be translated, the second character of a pair being the character into which the first character is to be translated.

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

NTTAB Syntax

The `NTTAB` macro is specified as follows:

NTTAB a1,a2,b1,b2,c1,c2,...

Example of NTTAB Macro

```
NTTAB 5E,'Ä','ö',78,FF,00,'ü','Ü'
```

In this example, the character represented by H'5E' is translated into 'Ä', 'ö' into the character represented by H'78', the character represented by H'FF' into the character represented by H'00', and 'ü' into 'Ü'.

Example of TAB Parameter

With the TAB parameter, you must enclose the entire string of character pairs in parentheses, for example:

```
TAB=(5E,'Ä','ö',78,FF,00,'ü','Ü')
```

243

TAB1 - Alternative Output Translation

■ TAB1 Parameter Syntax	606
■ NTTAB1 Macro Syntax	606
■ Example of NTTAB1 Macro	607
■ Example of TAB1 Parameter	607

This Natural profile parameter allows you to overwrite the definitions in the translation table NTTAB1 as contained in the configuration module NATCONFIG. The NTTAB1 table is the alternative output translation table for the secondary character set used when the profile/session parameter `PM=C` is set.

TAB1 corresponds to the NTTAB1 macro in the Natural parameter module NATPARM.

Possible settings		See <i>TAB1 Parameter Syntax</i> below.
Default setting		As specified within the macro NTTAB1 in NATCONFIG.
Dynamic specification	yes	This parameter can only be specified dynamically. In the Natural parameter module NATPARM, the macro NTTAB1 must be used instead.
Specification within session	no	

The following topics are covered below:

TAB1 Parameter Syntax

The TAB1 parameter is specified as follows:

TAB1=(a1,a2,b1,b2,c1,c2,...)

You specify pairs of characters, the first character of a pair being the character to be translated, the second character of a pair being the character into which the first character is to be translated.

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

NTTAB1 Macro Syntax

The NTTAB1 macro is specified as follows:

NTTAB1 a1,a2,b1,b2,c1,c2,...

Example of NTTAB1 Macro

```
NTTAB1 5E,'Ä','ö',78,FF,00,'ü','Ü'
```

In this example, the character represented by H'5E' is translated into 'Ä', 'ö' into the character represented by H'78', the character represented by H'FF' into the character represented by H'00', and 'ü' into 'Ü'.

Example of TAB1 Parameter

With the TAB1 parameter, you must enclose the entire string of character pairs in parentheses, for example:

```
TAB1=(5E,'Ä','ö',78,FF,00,'ü','Ü')
```

244

TAB2 - Alternative Input Translation

■ TAB2 Parameter Syntax	610
■ NTTAB2 Macro Syntax	610
■ Example of NTTAB2 Macro	611
■ Example of TAB2 Parameter	611

This Natural profile parameter allows you to overwrite the definitions in the translation table NTTAB2 as contained in the configuration module NATCONFIG. The NTTAB2 table is the alternate input translation table for the secondary character set used when the profile/session parameter PM is set to PM=C.

TAB2 corresponds to the NTTAB2 macro in the Natural parameter module NATPARM.

Possible settings		See TAB2 Parameter Syntax below.
Default setting		As specified within the macro NTTAB2 in NATCONFIG.
Dynamic specification	yes	This parameter can only be specified dynamically. In the Natural parameter module NATPARM, the macro NTTAB2 must be used instead.
Specification within session	no	

The following topics are covered below:

TAB2 Parameter Syntax

The TAB2 parameter is specified as follows:

TAB2=(a1,a2,b1,b2,c1,c2,...)

You specify pairs of characters, the first character of a pair being the character to be translated, the second character of a pair being the character into which the first character is to be translated.

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

NTTAB2 Macro Syntax

The NTTAB2 macro is specified as follows:

NTTAB2 a1,a2,b1,b2,c1,c2,...

Example of NTTAB2 Macro

```
NTTAB2 5E,'Ä','ö',78,FF,00,'ü','Ü'
```

In this example, the character represented by H'5E' is translated into 'Ä', 'ö' into the character represented by H'78', the character represented by H'FF' into the character represented by H'00', and 'ü' into 'Ü'.

Example of TAB2 Parameter

With the TAB2 parameter, you must enclose the entire string of character pairs in parentheses, for example:

```
TAB2=(5E,'Ä','ö',78,FF,00,'ü','Ü')
```


245

TABA1 - EBCDIC-to-ASCII Translation

■ TABA1 Parameter Syntax	614
■ NTTABA1 Macro Syntax	615
■ Example of NTTABA1 Macro	615
■ Example of TABA1 Parameter	615

This Natural profile parameter allows you to overwrite the definitions in the translation table NTTABA1 as contained in the configuration module NATCONFIG. This table is used for EBCDIC-to-ASCII translation.

TABA1 corresponds to the NTTABA1 macro in the Natural parameter module NATPARM.

Possible settings		See TABAI Parameter Syntax below.
Default setting		As specified within the macro NTTABA1 in NATCONFIG.
Dynamic specification	yes	This parameter can only be specified dynamically. In the Natural parameter module NATPARM, the macro NTTABA1 must be used instead.
Specification within session	no	

The following topics are covered below:

TABA1 Parameter Syntax

The TABA1 parameter is specified as follows:

```
TABA1=(a1,a2,b1,b2,c1,c2,...)
```

You specify pairs of characters, the first character of a pair being an EBCDIC character to be translated, the second character of a pair being the ASCII character into which the EBCDIC character is to be translated.

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

NTTABA1 Macro Syntax

The NTTABA1 macro is specified as follows:

```
NTTABA1 a1,a2,b1,b2,c1,c2,...
```

Example of NTTABA1 Macro

```
NTTABA1 5E,'Ä','ö',78,FF,00,'ü','Ü'
```

In this example, the character represented by H'5E' is translated into 'Ä', 'ö' into the character represented by H'78', the character represented by H'FF' into the character represented by H'00', and 'ü' into 'Ü'.

Example of TABA1 Parameter

With the TABA1 parameter, you must enclose the entire string of character pairs in parentheses, for example:

```
TABA1=(5E,'Ä','ö',78,FF,00,'ü','Ü')
```

■ TABA2 Parameter Syntax	618
■ NTTABA2 Macro Syntax	619
■ Example of NTTABA2 Macro	619
■ Example of TABA2 Parameter	619

This Natural profile parameter allows you to overwrite the definitions in the translation table NTTABA2 as contained in the configuration module NATCONFIG. This table is used for ASCII-to-EBCDIC translation.

TABA2 corresponds to the [NTTABA2](#) macro in the Natural parameter module NATPARM.

Possible settings		See TABA2 Parameter Syntax below.
Default setting		As specified within the macro NTTABA2 in NATCONFIG.
Dynamic specification	yes	This parameter can only be specified dynamically. In the Natural parameter module NATPARM, the macro NTTABA2 must be used instead.
Specification within session	no	

The following topics are covered below:

TABA2 Parameter Syntax

The TABA2 parameter is specified as follows:

```
TABA2=(a1,a2,b1,b2,c1,c2,...)
```

You specify pairs of characters, the first character of a pair being an ASCII character to be translated, the second character of a pair being the EBCDIC character into which the ASCII character is to be translated.

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

NTTABA2 Macro Syntax

The NTTABA2 macro is specified as follows:

```
NTTABA2 a1,a2,b1,b2,c1,c2,...
```

Example of NTTABA2 Macro

```
NTTABA2 5E,'Ä','ö',78,FF,00,'ü','Ü'
```

In this example, the character represented by H'5E' is translated into 'Ä', 'ö' into the character represented by H'78', the character represented by H'FF' into the character represented by H'00', and 'ü' into 'Ü'.

Example of TABA2 Parameter

With the TABA2 parameter, you must enclose the entire string of character pairs in parentheses, for example:

```
TABA2=(5E,'Ä','ö',78,FF,00,'ü','Ü')
```

- TABL Parameter Syntax 622
- NTTABL Macro Syntax 622
- Example of NTTABL Macro 623
- Example of TABL Parameter 623

This Natural profile parameter allows you to overwrite the definitions in the translation table `NTTABL` as contained in the configuration module `NATCONFIG`. The `NTTABL` table is used to translate output produced by programs located in `SYS . . .` libraries.

`TABL` corresponds to the `NTTABL` macro in the Natural parameter module `NATPARM`.

Possible settings		See TABL Parameter Syntax below.
Default setting		As specified within the macro <code>NTTABL</code> in <code>NATCONFIG</code> .
Dynamic specification	yes	This parameter can only be specified dynamically. In the Natural parameter module <code>NATPARM</code> , the macro <code>NTTABL</code> must be used instead.
Specification within session	no	

The following topics are covered below:

TABL Parameter Syntax

The `TABL` parameter is specified as follows:

TABL=(a1,a2,b1,b2,c1,c2,...)

You specify pairs of characters, the first character of a pair being the character to be translated, the second character of a pair being the character into which the first character is to be translated.

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

NTTABL Macro Syntax

The `NTTABL` macro is specified as follows:

NTTABL a1,a2,b1,b2,c1,c2,...

Example of NTTABL Macro

```
NTTABL 5E,'Ä','ö',78,FF,00,'ü','Ü'
```

In this example, the character represented by H'5E' is translated into 'Ä', 'ö' into the character represented by H'78', the character represented by H'FF' into the character represented by H'00', and 'ü' into 'Ü'.

Example of TABL Parameter

With the TABL parameter, you must enclose the entire string of character pairs in parentheses, for example:

```
TABL=(5E,'Ä','ö',78,FF,00,'ü','Ü')
```


248

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.

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*.

Possible settings	any character	Up to 10 characters may be specified. Trailing characters may optionally be specified enclosed within apostrophes, in which case any characters can be specified. Any character string specified which contains a closing parenthesis or a quotation mark must be enclosed within apostrophes.	
Default setting	none		
Specification within session	yes	Applicable Statements:	DISPLAY FORMAT
		Applicable Command:	none

Examples:

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

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

249

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

Possible settings	AUTO	Natural compares the physical (store clock) and logical (system environment) machine times and uses the difference between the two as the setting for the TD parameter. For a time change to take effect for Natural (for example, to change time to summer time or back to winter time), it is therefore sufficient to reset the logical machine time.
	+/- hh (+/- hh, mm) (+/- hh, mm, ss)	Hours, minutes and seconds from (-23,59,59) to (+23,59,59). A plus (optional) or minus sign indicates, whether the TD value is to be added or subtracted. The specified time is added to or subtracted from the physical machine time to set the time/date to be used by Natural.
	1 to 32 characters	Name of the time zone to be used. This must be defined as a valid time zone in the NTTZ macro of the NATCONFIG module, see <i>Configuration Tables - Module NATCONFIG</i> .
Default setting	0	
Dynamic specification	yes	
Specification within session	no	
Application Programming Interface	USR1005N	See SYSEXT - <i>Natural Application Programming Interfaces</i> in the <i>Utilities</i> documentation.

Examples:

```
TD=6           (6 hours ahead)
TD=(5,30)      (5 hours and 30 minutes ahead)
TD=(-6,12,30)  (6 hours, 12 minutes and 30 seconds behind)
TD='USA-EST'   (eastern time zone as defined in NTTZ macro)
```

z/VSE-Specific Information: With VSE-type operating systems, `// ZONE` and `//DATE` JCL statements are honored with `TD=AUTO`. This can also affect the setting of the profile parameter `DD`. See also the profile parameters [YD](#) and [DD](#).

250

TF - Translation of Database ID/File Number

■ TF Parameter Syntax	631
■ NTTF Macro Syntax	631
■ Example of TF Parameter	631
■ Example of NTTF Macro	631



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

This Natural profile parameter can be used to translate a database ID/file number to another database ID/file number during the execution of an application. It corresponds to the macro `NTTF` in the parameter module `NATPARM`.

Possible settings	<i>production-DBID</i>	Must be in the range of 0-254, or 256-65535, or can be an asterisk (*) which stands for all DBIDs. Database ID 255 is reserved for logical system files for Software AG products, see profile parameter LFILE .
	<i>production-FNR</i>	Must be in the range of 1-65535, or an asterisk (*) which stands for all FNRs.
	<i>test-DBID</i>	Must be in the range of 0-254, or 256-65535, or can be an asterisk (*) which leaves the DBID unchanged.
	<i>test-FNR</i>	Must be in the range of 1- 65535, or an asterisk (*) which leaves the FNR unchanged.
Default setting	none	
Dynamic specification	yes	This parameter can only be specified dynamically. In the Natural parameter module <code>NATPARM</code> , the macro <code>NTTF</code> must be used instead.
Specification within session	no	
Application Programming Interface	USR1034N	See <i>SYSEXT - Natural Application Programming Interfaces</i> in the <i>Utilities</i> documentation.
	USR2005N *	
		* Recommended.

This feature is relevant when developing an application in a production environment. It enables you to develop an application in a test database and then transfer the finished application to the production database without having to change or re-compile the application. The Natural objects are cataloged with the production DBID/FNR, but whenever a database access is executed, the production DBID/FNR is translated into the test DBID/FNR according to the `TF` parameter specifications; that is, the test database is used. This means that testing can take place in the actual production environment, but not with production data.

The `TF` parameter or the `NTTF` macro can be specified several times so as to specify different combinations of file numbers.

The following topics are covered below:

TF Parameter Syntax

The TF profile parameter is specified as follows:

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

NTTF Macro Syntax

The NTTF macro is specified as follows:

```
NTTF production-DBID,production-FNR,test-DBID,test-FNR
```

Example of TF Parameter

```
TF=(777,39,17,88),TF=(251,*,9,*)
```

Example of NTTF Macro

Equivalent specification in the Natural parameter module:

```
NTTF 777,39,17,88  
NTTF 251,*,9,*
```

251

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.



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

Possible settings	any character	At runtime, the dynamic thousands separator is replaced with this character. 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. 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)	By default, a comma is used as thousands separator.	
Dynamic specification	yes		
Specification within session	yes	Applicable Statements:	none
		Applicable Command:	GLOBALS

See also:

- Option THSEP of system command COMPOPT in the *System Commands* documentation.
- Keyword subparameter THSEP of profile parameter CPMO or macro NTCPMO.
- *Customizing Separator Character Displays* in the *Programming Guide*.

252

TIMEOUT - Wait Time for RPC Server Response

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

This specifies the number of seconds the client is to wait for an RPC server response. If this time is exceeded, the remote procedure call will be terminated with a corresponding error message.

TIMEOUT is specified on the client side only.

Possible settings	0 - 32767	Seconds.
Default setting	55	
Dynamic specification	yes	
Specification within session	yes	At runtime, this value can be overwritten using the Parameter Maintenance function of the SYSRPC utility.

For further information, see the *Natural Remote Procedure Call (RPC)* documentation.

253

TMODEL - IBM 3270 Terminal Model

This Natural profile parameter is for IBM mainframes only or for the Natural Web I/O Interface.

Under CICS, this parameter is ignored for terminal bound sessions, because the terminal screen size is defined by the CICS terminal control table.

TMODEL controls the IBM 3270 terminal model number for online environments, for example, under IMS TM. It is used to determine the number of lines and columns of the terminal screen. It can be also used under the Natural Development Server (NDV) and for the Natural Web I/O Interface (NWO) server in any operating system environment for defining the terminal screen size for the Natural Web I/O Interface. For further information, refer to the Natural Development Server documentation, Configuring the Natural Development Server, or to the *Natural Web I/O Interface* documentation.

Possible settings	0	The screen size is determined by the environment-dependant driver module. If possible, it gets the screen size information from its subsystem. Otherwise, the definitions of default Model 2 are used, for example, under IMS TM. When the Natural Web I/O Interface is used, the default screen size is 43 lines and 132 columns.
	2	The screen size is 24 lines and 80 columns.
	3	The screen size is 32 lines and 80 columns.
	4	The screen size is 43 lines and 80 columns.
	5	The screen size is 27 lines and 132 columns.
	(<i>lines</i> , <i>cols</i>)	This syntax is allowed for NWO server terminals only. The number of lines (<i>lines</i>) can be 24 through 250, and the number of columns (<i>cols</i>) can be 80 through 250.
Default setting	0	
Dynamic specification	yes	

Specification within session	no	
------------------------------	----	--

**Notes:**

1. If your TMODEL specification is incompatible with the physical terminal screen size, the output data may be displayed incorrectly or hardware errors may occur.
2. The terminal screen size has a direct influence on the storage required for the terminal I/O buffers used by Natural.

254

TPF (Internal Use)

This parameter is reserved for internal use by Natural.



Caution: Do not change its setting.

255

TQ - Translate Quotation Marks

This parameter has been replaced by subparameter TQMARK of profile parameter **CMPO**.

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

It activates the RPC trace facility and determines the trace level *n* to be used. For further information, see *Using the Server Trace Facility* p.p. in the *Natural Remote Procedure Call (RPC)* documentation.

TRACE is specified on the server side only.

Possible settings	0	Nothing is traced.
	1	Only messages (inclusive Natural errors) are traced.
	(1 , E)	Messages are traced in the event of an error only.
	2	All messages and data from/to client are traced.
	(2 , E)	Messages and data from/to client are traced in the event of an error only.
Default setting	0	
Dynamic specification	yes	
Specification within session	no	


The values 3 - 9 are also accepted. These values are for future use and behave like TRACE=2.

For further information see the *Natural Remote Procedure Call (RPC)* documentation.

■ TRACE Parameter Syntax	646
■ NTTRACE Macro Syntax	647
■ Example of TRACE Parameter	647
■ Example of NTTRACE Macro	647

This Natural profile parameter is intended primarily for Software AG internal use for debugging purposes. It can be used to define the components for which trace data are to be written. It does not activate trace recording.

Trace recording can be activated by the profile parameters [ITRACE](#) (internal trace) and [ETRACE](#) (external trace) or during the session by the corresponding terminal commands %TRI and %TRE.

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

TRACE corresponds to the macro [NTTRACE](#) in the Natural parameter module NATPARM.

Possible settings	list of <i>trace-IDs</i>	<i>trace-IDs</i> (each 1-8 bytes) define the names of the Natural components to be traced. Component names have to be entered in upper case.
Default setting	none	
Dynamic specification	yes	This parameter can only be specified dynamically. In the Natural parameter module NATPARM, the macro NTTRACE must be used instead.
Specification within session	no	

The setting lists of multiple TRACE parameter specifications are not concatenated; that is, a TRACE parameter overrides any previously specified TRACE parameter and any NTTRACE macro definitions.

The following topics are covered below:

TRACE Parameter Syntax

The TRACE parameter is specified as follows:

TRACE=([trace-ID1](#),[trace-ID2](#),...)

NTTRACE Macro Syntax

The NTTRACE macro is specified as follows:

```
NTTRACE trace-ID1,trace-ID2,...
```

Multiple specifications of the NTTRACE macro are concatenated to one trace list.

Example of TRACE Parameter

```
TRACE=(NATGETM,NATFREM,DYNPARMS)
```

This defines traces to be written for the Natural nucleus components “storage acquisition”, “storage release” and “dynamic parameter evaluation”.

Example of NTTRACE Macro

Equivalent specification in the Natural parameter module:

```
NTTRACE NATGETM,NATFREM,DYNPARMS
```


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

It determines which server transport protocol is used. If ACI is used, you can additionally specify the transport method.

TRANSP is specified on the server side only.

Possible settings	ACI	ACI is used. The transport method is defined by the EntireX Broker.
	(ACI , TCP)	ACI is used with TCP/IP.
	(ACI , NET)	ACI is used with Entire Net-work, i.e. using the Adabas protocol.
	(ACI , TCP - NET)	Trying to use ACI with TCP. If not available, ACI is used with NET.
	(ACI , NET - TCP)	Trying to use ACI with NET. If not available, ACI is used with TCP.
Default setting	ACI	
Dynamic specification	yes	
Specification within session	no	

The use of TRANSP is no longer required as you may now specify the full node name with [SRVNODE](#). It is still supported for compatibility reasons.

For further information, see the *Natural Remote Procedure Call (RPC)* documentation.

259

TRYALT - Try Alternative Server Address

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

It determines whether an RPC client should try to execute an RPC request on an alternative server (ON) or not (OFF). For further information, see *Specifying RPC Server Addresses* in the *Natural Remote Procedure Call (RPC)* documentation.

TRYALT is specified on the client side only.

Possible settings	ON	If a request could not be executed on the node you specified, the RPC client tries to find an alternative server address to send that request to.
	OFF	No such attempt will be made.
Default setting	OFF	
Dynamic specification	yes	
Specification within session	yes	At runtime, this value can be overwritten using the Parameter Maintenance function of the SYSRPC utility.

For further information, see the *Natural Remote Procedure Call (RPC)* documentation.

260

TS - Translate Output from Programs in System

Libraries



Important: The TS parameter applies only to primary output (CMPRINT, see Natural in Batch Mode in the *Operations* documentation).

This Natural profile and session parameter is used to translate output from Natural system libraries (that is, libraries whose names begin with SYS) using a translation table. This may be necessary for locations which have non-standard lower-case usage (for example, Middle East or Far East countries).

Error messages or warnings are translated if the English version of the text is displayed. If the text is displayed in the local language (for example, Hebrew), it is not translated into upper-case characters. The translation of messages and warnings does not depend on the library from where the program is executed.

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

Possible settings	ON	Output is translated. With TS=ON, the profile parameter LC=OFF and the session parameter AD=T, both of which translate input to upper case, are ignored, as they would cause undesired character translation for special character sets.	
	OFF	Output is not translated.	
Default setting	OFF		
Dynamic specification	yes		
Specification within session	yes	Applicable Statements:	SET GLOBALS
		Applicable Command:	GLOBALS
Application Programming Interface	USR1005N	See SYSEXT - <i>Natural Application Programming Interfaces</i> in the <i>Utilities</i> documentation.	



Note: The translation table can be modified with the `NTTABL` macro or the corresponding dynamic profile parameter `TABL`.

Support of TS=ON for Natural under IMS TM Messages

All Natural under IMS TM messages are translated into upper case if `TS=ON` is specified in the Natural session.

Support of TS=ON for RPC Server Trace

All messages in the Natural RPC server trace are translated into upper case if `TS=ON` is specified in the Natural RPC server session. The trace of the data from/to the client is not affected by `TS=ON` and remains unchanged.

Other Parameters to Provide Upper Case Translation

In addition to honoring `TS=ON`, several Natural components provide an `UCTRAN` parameter to provide for translation of messages into upper case, even if the setting of the `TS` parameter is not (or not yet) available. These components are:

- Authorized Services Manager
- Roll Server
- Global Buffer Pool Manager under z/OS and z/VSE
- Natural Com-plete/SMARTS Interface
- Natural Remote Procedure Call

See *Startup Parameters* in z/OS Batch Mode and *Startup Parameters* under CICS in the *Natural Remote Procedure Call (RPC)* documentation.

For the Natural Development Server, the configuration parameter `UPPERCASE_SYSTEMMESSAGES` with similar functionality is available, for details, see *Configuring the Natural Development Server* in the *Natural Development Server* documentation.

261

TSIZE - Size of Buffer for Adabas Text Retrieval

This Natural profile parameter specifies the size of the buffer to be used for the Adabas Text Retrieval facility.

Alternatively, you can use the equivalent Natural profile parameter [DS](#) or macro [NTDS](#), see *Using Optional Macros in a Natural Parameter Module* in the *Operations* documentation to specify the `TSIZE` value.

Possible settings	1 - 2097151	Buffer size in KB. If the requested space is not available, the Adabas Text Retrieval facility cannot be used.
	0	Adabas Text Retrieval facility is not used.
Default setting	0	
Dynamic specification	yes	
Specification within session	no	

262

TTYTYPE - Terminal Type

This Natural profile parameter allows you to specify the terminal type used - in TP environments in which this information is not supplied automatically - so that Natural can activate the appropriate converter routine for attribute sequences to operate that type of terminal.

Possible settings	1-4 characters	The setting specified with the TTYTYPE parameter must be defined as a valid terminal device type in the NTDVCE macro of the NATCONFIG module, see <i>Configuration Tables - Module NATCONFIG</i> .
Default setting	IBM	3270
	Siemens	The setting defined in PDN, unless overridden by the parameter T975X (see <i>Natural TP Monitor Interfaces, Natural under TIAM, Parameters in Macro NAMTIAM</i>).
Dynamic specification	yes	
Specification within session	yes	The TTYTYPE parameter has the same function as the terminal command %T=.



Note: If you use the TTYTYPE parameter, it is no longer necessary to execute a program containing a SET CONTROL 'T=...' statement at the start of the session in order to set the terminal type.

263

UC - Underlining Character

This session parameter determines the character that is used as underlining character for the following:

- column headings generated by `DISPLAY` statements;
- page titles/trailers produced by `WRITE TITLE`/`WRITE TRAILER` statements with `UNDERLINED` option.

Possible settings	any character	See also Note below.	
	OFF		
Default setting	- (hyphen)		
Specification within session	yes	Applicable Statements:	DISPLAY FORMAT 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=+)
```

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

264

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 - 65535, except 255	Valid database ID. Database ID 255 is reserved for logical system files for Software AG products, see profile parameter LFILE .
Default setting	database ID applicable for FUSER	
Dynamic specification	yes	
Specification within session	no	
Application Programming Interface	USR1005N	See <i>SYSEXT - Natural Application Programming Interfaces</i> in the <i>Utilities</i> documentation. * Recommended.
	USR1040N *	



Notes:

1. The DBID 0 and the databases selected with the `UDB` parameter must be of the same type (ADA/ADA, SQL/SQL or XML/XML for example).
2. If no DBID is specified in the DDM used, the DBID specified with the `UDB` profile parameter determines which database is accessed. Thus it is possible to have different user environments without multiple `FUSER` files being required.
3. If no DBID is specified in the DDM and the `UDB` profile parameter is not specified, the DBID that applies to the `FUSER` system file is used.

265

ULANG - User Language

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



Note: See also the note on language code related adaptation of profile parameter [CP](#) when set to ON.

Possible settings	1 - 60	Language code. For example, 1 is assigned to English, 2 is assigned to German, 3 is assigned to French. For a detailed list of language codes, see the table in the documentation of the *LANGUAGE variable.
Default setting	1	
Dynamic specification	yes	
Specification within session	no	
Application Programming Interface	USR1005N	See SYSEXT - <i>Natural Application Programming Interfaces</i> in the <i>Utilities</i> documentation.

Within the session, the language code can be specified using the terminal command %L=.

See also:

- *Configuration Tables - Module NATCONFIG* in the *Operations* documentation for additional information about language indicators and possible settings.
- *Screen Design, Skill-Sensitive User Interfaces* in the *Programming Guide*.

This Natural profile parameter is used for debugging in Natural under z/VSE.

See also *Debugging Facilities for Natural under z/VSE* in the *Operations* documentation.

It specifies UPSI settings for the Natural z/VSE Interface corresponding to the z/VSE UPSI system control statement. The UPSI profile parameter is in particular relevant in cases where UPSI system control statement settings have produced side effects in the sense that they have a different meaning for other programs such as for front-end Natural or for programs called by Natural.

Possible settings	1-8 characters	Any combination of the characters 0, 1, X.
Default setting	XXXXXXXX	
Dynamic specification	yes	
Specification within session	no	

The syntax for the UPSI string is the same as for the z/VSE UPSI system control statement.

The Natural z/VSE batch interface takes the UPSI settings in JCL and merges the UPSI profile settings into it according to the following rules:

0	The corresponding bit is 0.
1	The corresponding bit is 1.
X	The corresponding bit remains unchanged.

267

USER - Restrict Use of Profile Parameter Strings and Modules

■ USER Parameter Syntax	669
■ NTUSER Macro Syntax	669
■ Example of NTUSER Macro	669
■ Example of USER Parameter	669

This Natural profile parameter can be used to restrict the use of dynamic parameter strings as specified in a SYSPARM profile, **NTSYS** macro or parameter dataset (CMPRMIN) or to restrict an alternative parameter module (NATPARM).

Possible settings	list of user IDs	The IDs of the users who will be allowed to use the subsequently specified string of profile parameters. Only the specified users will then be allowed to use that parameter string.
Default setting	none	
Dynamic specification	yes	This parameter can only be specified dynamically. To restrict the use of an alternative parameter module (NATPARM), the corresponding macro NTUSER must be used instead.
Specification within session	no	

The **USER** parameter applies only to the string of dynamic parameters specified *after* it. The **NTUSER** macro applies to the parameter module in which it is specified. The default Natural parameter module linked to the environment-dependent Natural nucleus cannot be restricted.

When the dynamic profile parameters are evaluated and the **USER** parameter is encountered, Natural checks if the current user ID (that is, the current setting of the system variable *INIT-USER) is contained in the list of user IDs specified with the **USER** parameter. If it is not, the user receives a corresponding error message, and the processing of dynamic profile parameters is terminated immediately.

When an alternative parameter module is to be used, Natural loads the alternative parameter module specified by the **PARM** parameter and checks if the current user ID (that is, the current setting of the system variable *INIT-USER) is contained in the list of user IDs specified by the **NTUSER** macro in the alternative parameter module. If it is not, the user receives a corresponding error message, and the alternative parameter module is discarded.

To restrict the use of:

- a SYSPARM profile,
you specify the **USER** parameter as the first parameter in the profile. The subsequent string of profile parameters in the profile, that is, the entire profile, can then only be used by the user specified with the **USER** parameter.
- a parameter string defined by an **NTSYS** macro or in a CMPRMIN dataset,
you specify the **USER** parameter as the first parameter in the parameter string.
- an alternative parameter module,
you specify the macro **NTUSER** in the alternative parameter module.

The following topics are covered below:

USER Parameter Syntax

The parameter syntax of USER is as follows:

```
USER=(user-id1,user-id2,...)
```

NTUSER Macro Syntax

The NTUSER macro is specified in a Natural parameter module as follows:

```
NTUSER user-id1,user-id2,user-id3,...  
NTUSER user-id4,user-id5,...  
...
```

Example of NTUSER Macro

The following is an example of protecting a Natural parameter macro:

```
NTPRM ...  
...  
NTUSER ADMIN1,ADMIN2
```

Example of USER Parameter

```
USER=(ADMIN1,ADMIN2),FNAT=(12,177,SECPASSW,74832055)
```

268 USERBUF (Internal Use)

This parameter is reserved for internal use by Natural.



Caution: Do not change its setting.

■ UTAB1 Parameter Syntax	674
■ NTUTAB1 Macro Syntax	674
■ Example of NTUTAB1 Macro	675
■ Example of UTAB1 Parameter	675

This Natural profile parameter allows you to overwrite the definitions in the translation table NTUTAB1 as contained in the configuration module NATCONFIG. The NTUTAB1 table is used for lower-to-upper-case translation.

UTAB1 corresponds to the NTUTAB1 macro in the Natural parameter module NATPARM.

Possible settings		See UTAB1 Parameter Syntax below.
Default setting		As specified within the macro NTUTAB1 in NATCONFIG.
Dynamic specification	yes	This parameter can only be specified dynamically. In the Natural parameter module NATPARM, the macro NTUTAB1 must be used instead.
Specification within session	no	

The following topics are covered below:

UTAB1 Parameter Syntax

The UTAB1 parameter is specified as follows:

UTAB1=(a1,a2,b1,b2,c1,c2,...)

You specify pairs of characters, the first character of a pair being a lower-case character to be translated, the second character of a pair being the upper-case character into which the lower-case character is to be translated.

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

NTUTAB1 Macro Syntax

The NTUTAB1 macro is specified as follows:

NTUTAB1 a1,a2,b1,b2,c1,c2,...

Example of NTUTAB1 Macro

```
NTUTAB1 5E,'Ä','ö',78,FF,00,'ü','Ü'
```

In this example, the character represented by H'5E' is translated into 'Ä', 'ö' into the character represented by H'78', the character represented by H'FF' into the character represented by H'00', and 'ü' into 'Ü'.

Example of UTAB1 Parameter

With the UTAB1 parameter, you must enclose the entire string of character pairs in parentheses, for example:

```
UTAB1=(5E,'Ä','ö',78,FF,00,'ü','Ü')
```

■ UTAB2 Parameter Syntax	678
■ NTUTAB2 Macro Syntax	678
■ Example of NTUTAB2 Macro	679
■ Example of UTAB2 Parameter	679

This Natural profile parameter allows you to overwrite the definitions in the translation table NTUTAB2 as contained in the configuration module NATCONFIG. The NTUTAB2 table is used for upper-to-lower case translation.

UTAB2 corresponds to the [NTUTAB2](#) macro in the Natural parameter module NATPARM.

Possible settings		See UTAB2 Parameter Syntax below.
Default setting		As specified within the macro NTUTAB2 in NATCONFIG.
Dynamic specification	yes	This parameter can only be specified dynamically. In the Natural parameter module NATPARM, the macro NTUTAB2 must be used instead.
Specification within session	no	

The following topics are covered below:

UTAB2 Parameter Syntax

The UTAB2 parameter is specified as follows:

UTAB2=(a1,a2,b1,b2,c1,c2,...)

You specify pairs of characters, the first character of a pair being a upper-case character to be translated, the second character of a pair being the lower-case character into which the upper-case character is to be translated.

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

NTUTAB2 Macro Syntax

The NTUTAB2 macro is specified as follows:

NTUTAB2 a1,a2,b1,b2,c1,c2,...

Example of NTUTAB2 Macro

```
NTUTAB2 5E,'Ä','ö',78,FF,00,'ü','Ü'
```

In this example, the character represented by H'5E' is translated into 'Ä', 'ö' into the character represented by H'78', the character represented by H'FF' into the character represented by H'00', and 'ü' into 'Ü'.

Example of UTAB2 Parameter

With the UTAB2 parameter, you must enclose the entire string of character pairs in parentheses, for example:

```
UTAB1=(5E,'Ä','ö',78,FF,00,'ü','Ü')
```

271

VSIZE - Size of Buffer Area for Natural/VSAM

This Natural profile parameter applies only if the Natural VSAM interface is installed.

It sets the maximum size of the buffer area required by Natural for VSAM. If set to 0 or if the requested space is not available, the Natural for VSAM Interface cannot be used.

Possible settings	1 - 512	Buffer size in KB. The size actually required depends on the specifications in the NVSPARM macro (described in the <i>Natural for VSAM</i> documentation). If the requested space is not available, the Natural VSAM interface cannot be used. An appropriate error message at the initialization of the Natural VSAM interface tells you which buffer specified in NVSPARM does not fit into the VSIZE area; you can then either reduce individual buffer sizes in NVSPARM or increase the size of the VSIZE area.
	0	With VSIZE=0, the Natural VSAM interface cannot be used.
Default setting	0	
Dynamic specification	yes	
Specification within session	no	



Note: If Natural is installed for VSAM, the corresponding Natural buffers are requested at the initialization of the Natural session. If you do not need VSAM support during a Natural session, it is recommended that you invoke Natural with VSIZE=0 to avoid overhead caused by handling of unused buffers.

■ WEBIO Parameter Syntax	684
■ NTWEBIO Macro Syntax	684
■ Keyword Subparameters	685

This Natural profile parameter allows you to individually enable or disable the rendering of certain features of the Natural Web I/O Interface display on the basis of a style sheet. It corresponds to the [NTWEBIO](#) macro in the parameter module NATPARM.

For further information, see the corresponding sections in *Using Style Sheets* in the *Natural Web I/O Interface* documentation.

Possible settings	See Keyword Subparameters , below.	Possible subparameter keywords: ML KEYS WIN
Default setting	ML=OFF , KEYS=OFF , WIN=OFF	By default, the style sheet based rendering of the message line, PF key buttons and Natural window objects is disabled.
Dynamic specification	yes	The parameter WEBIO can only be specified dynamically. In NATPARM, use the macro NTWEBIO.
Specification within session	no	

The following topics are covered below:

WEBIO Parameter Syntax

The WEBIO parameter is specified as follows:

```
WEBIO=({ML=[ON|OFF],}{KEYS=[ON|OFF],}{WIN=[ON|OFF]})
```

For names and values of keyword subparameters, see [Keyword Subparameters](#) below.

NTWEBIO Macro Syntax

The NTWEBIO macro is specified as follows:

```
NTWEBIO {ML=[ON|OFF],}{KEYS=[ON|OFF],}{WIN=[ON|OFF]}
```

Keyword Subparameters

ML | KEYS | WIN

ML - Message Line

Enables/disables the style sheet based rendering of the message line. See also *Modifying the Message Line* in the *Natural Web I/O Interface* documentation. Possible values are:

ON	The style sheet based rendering of the message line is enabled.
OFF	The style sheet based rendering of the message line is disabled.

KEYS - PF Keys

Enables/disables the style sheet based rendering of the PF key buttons. See also *Modifying the Style of the PF Key Buttons* in the *Natural Web I/O Interface* documentation. Possible values are:

ON	The style sheet based rendering of the PF key buttons is enabled.
OFF	The style sheet based rendering of the PF key buttons is disabled.

WIN - Window Objects

Enables/disables the style sheet based rendering of Natural window objects. See also *Modifying the Natural Windows* in the *Natural Web I/O Interface* documentation. Possible values are:

ON	The style sheet based rendering of Natural window objects is enabled.
OFF	The style sheet based rendering of Natural window objects is disabled.

273

WH - Wait for Record in Hold Status

This Natural profile and session parameter applies to Adabas databases only.

It specifies the action to be taken if a required record is not available for processing, because it has been placed in hold status by another user.

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

Possible settings	ON	The user is placed in wait status until either the requested record becomes available, or an error message is issued due to Adabas exceeding a time limit or other limit while attempting to place the record in hold status.	
	OFF	An error message is returned if any of these records cannot be placed in hold status.	
Default setting	OFF		
Dynamic specification	yes		
Specification within session	yes	Applicable Statements:	SET GLOBALS
		Applicable Command:	GLOBALS
Application Programming Interface	USR1005N	See <i>SYSEXT - Natural Application Programming Interfaces</i> in the <i>Utilities</i> documentation.	



Note: When a Natural statement is executed which results in Adabas records being read and an update/delete operation could follow, Natural requests that Adabas places these records in hold status. See the Adabas *Command Reference* documentation for further information on hold processing.

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

■ WORK Parameter Syntax	690
■ NETWORK Macro Syntax	691
■ Keyword Subparameters for All Environments	692
■ Keyword Subparameters for AM=STD in All Environments	696
■ Keyword Subparameters for AM=STD in z/OS Environments	697
■ Keyword Subparameters for AM=STD in z/VSE Environments	699
■ Keyword Subparameters for AM=STD in BS2000/OSD Environments	700
■ Keyword Subparameters for AM=CICS	701
■ Keyword Subparameters for AM=COMP	702
■ Keyword Subparameters for AM=SMARTS	702

This Natural profile parameter allows you to define the maximum number of work files to be used during the session. Within a session, up to 32 logical work files (numbered 1 to 32) can be used.

WORK corresponds to the [NETWORK](#) macro in the parameter module NATPARM. To provide different work file definitions, WORK or NETWORK can be specified multiple times.

Possible settings		See <i>Keyword Subparameters</i> below.
Default setting		See below. Depending on the access method and the environment, there may be different default settings.
Dynamic specification	yes	The parameter WORK can only be specified dynamically. In NATPARM, the macro NETWORK must be used.
Specification within session	no	

The software components for accessing work files in different environments are called access methods. For the duration of a Natural session, each logical work file can be assigned to one access method only. The access method for a work file is determined by the keyword subparameter AM (see below).

In z/OS under TSO and in batch mode, work files need not be predefined in the JCL. Provided they are defined by subparameter AM=STD, they can be allocated dynamically during the session by a Natural program using the DEFINE WORK FILE statement or the application programming interface USR2021, which is located in library SYSEXT.

This document covers the following topics:

See also *Print and Work File Handling with External Datasets in a Server Environment* in the *Operations* documentation).

WORK Parameter Syntax

With the WORK parameter, you first specify one or more logical work file numbers, and then several keyword subparameters, which define the characteristics for these work files:

WORK=((*work-file-numbers*),*keyword-subparameters*,...)

work-file-numbers

The file numbers must be specified first and enclosed in parentheses. The numbers can be from 1 to 31. They can be specified in any sequence. Multiple numbers must be separated from one another by commas or blanks. To specify a range of numbers, you can use a hyphen (-).

keyword-subparameters

The various types of keyword subparameters are described below.

For work files with different characteristics, you specify different **WORK** parameters. If any previous definition (or default) for the same work file exists, only the values for the specified keyword subparameters are overwritten, all other values remain unchanged.

Examples:

```
WORK=((2,12,18),AM=STD,DEST='WORK**')
WORK=((1,3,6-11,15),AM=COMP,OPEN=INITOBJ,CLOSE=CMD)
```

NETWORK Macro Syntax

With an **NETWORK** macro, you first specify one or more logical work file numbers, and then several keyword subparameters, which define the characteristics for these work files:

```
NETWORK (work-file-numbers),keyword-subparameters,...
```

work-file-numbers

The file numbers must be specified first and enclosed in parentheses. The numbers can be from 1 to 31. They can be specified in any sequence. Multiple numbers must be separated from one another by commas. To specify a range of numbers, you can use a hyphen (-).

keyword-subparameters

The various types of keyword subparameters are described below.

For work files with different characteristics, you specify different **NETWORK** macros. If any previous definition (or default) for the same work file exists, only the values for the specified keyword subparameters are overwritten, all other values remain unchanged.

Examples:

```
NETWORK (2,12,18),AM=STD,DEST='WORK**'
NETWORK (1,3,6-11,15),AM=COMP,OPEN=INITOBJ,CLOSE=CMD
```

Keyword Subparameters for All Environments

The following keyword subparameters are available: [AM](#) | [DEST](#) | [OPEN](#) | [CLOSE](#) | [LRECL](#) | [TRUNC](#) | [PAD](#) | [PADCHRO](#) | [PADCHRI](#)

AM - Type of Access Method

AM=*xxx* specifies the type of access method to be used.

For an online session, all work files to be used have to be assigned to a specific access method.

For a batch session, any work files not assigned to a specific access method will be automatically detected and assigned by the standard batch access method (AM=STD), provided that they have been predefined in the JCL. See also [FAMSTD - Overwriting of Print and Work File Access Method Assignments](#).

STD	Standard sequential files (batch, TSO, TIAM, CMS OS simulation).
COMP	Com-plete work files.
SMARTS	SMARTS work files. Work file on a SMARTS Portable File System (PFS).
CICS	CICS transient data or temporary storage.
CMS	CMS Disk and SFS files.
PC	Entire Connection.
USER	Third-party vendor work-file interface.
OFF	Unassigned. No automatic assignments if FAMSTD=OFF is set.
0	Unassigned. Automatic assignments if FAMSTD=OFF is set. This is the default value.



Note: WORK=OFF is equivalent to: WORK=((1 - 32)) , AM=OFF). It does not affect any of the other keyword subparameter specifications.

DEST - External Dataset Name

DEST=*name* specifies the external dataset name (1 - 8 characters).

This corresponds to the *operand1* of the DEFINE WORK FILE statement (and can be overwritten by a DEFINE WORK FILE specification).

The meaning of this keyword subparameter depends on the access method.

AM=STD	<p>DEST is the logical dataset name (DDNAME, LINK name, DTF name).</p> <p>If the destination is to be for multiple files, two asterisks (**) have to be specified for the file number. These will be replaced by the corresponding logical file number for each work file. A DEST value including two asterisks must be enclosed in apostrophes when using it as a dynamic parameter.</p> <p>The default value is DEST='CMWKF**' for IBM and DEST='W**' for SIEMENS environments.</p> <p>Under z/VSE, only 7-character names are supported.</p>
AM=CICS	<p>There is no default value for work files under CICS. Here, the DEST subparameter is mandatory; that is, CICS work files defined without a valid DEST specification are ignored.</p> <p>The Natural CICS interface also supports a variable (see TERMVAR parameter in the NCIPARM generation macro; &TID is the default) as part of the DEST value which, when being specified, is replaced by the actual CICS terminal ID; see also <i>Natural Print and Work Files under CICS</i> in the <i>TP Monitor Interfaces</i> documentation).</p>
AM=CMS	For usage of DEST under CMS, refer to <i>Natural under VM/CMS</i> in the <i>Operations</i> documentation).
AM=COMP	DEST defines the name of the Com-plete SD-file. The length is restricted to a maximum of 8 characters. If the file is defined with TYPE=TID, the DEST value is appended by the Com-plete stack level. The length is restricted to a maximum of 7 characters accordingly. SD-file names starting with '&&' are treated as temporary files which are deleted automatically after Natural termination.

OPEN - Time of File Opening

OPEN=xxx determines when the file is to be opened:

Value	The file is opened
INIT	for output at session initialization.
OBF	according to the default OPEN value for the different environments (Batch, CICS, Com-plete, TSO).
OBJ	when the execution of the first object which accesses the file starts. This is the default value.
INITOBF	for output at session initialization. Any subsequent re-opening of the file sets the default OPEN value for the different environments (Batch, CICS, Com-plete, TSO).
OBJ1	when the execution of the first object on level 1 which accesses the file starts. Otherwise, it is opened when it is first accessed.
ACC	when it is first accessed by a statement.
INITOBJ	for output at session initialization. Any subsequent re-opening of the file will be performed when the execution of the first object which accesses the file starts.
INITOBJ1	when the execution of the first object on level 1 which accesses the file starts. Otherwise, it is opened when it is first accessed.

Value	The file is opened
INITACC	for output at session initialization. Any subsequent re-opening of the file will be performed when it is first accessed by a statement.

CLOSE - Time of File Closure

CLOSE=xxx determines when the file is to be closed:

Value	The file is closed
OBJ	either when processing of the object in which it was first accessed is finished, or when command mode, NEXT mode or MAINMENU is reached.
CMD	when command mode, NEXT mode or MAINMENU is reached. This is the default value.
FIN	at session end. With CLOSE=FIN, a DEFINE WORK FILE statement causes an error if the work file was opened already. A CLOSE WORK FILE statement for the work file is ignored. When the end-of-file condition occurs during the READ WORK FILE statement, Natural closes the work file immediately.
USER	This value specifies that a work file is closed only if the file is open and one of the following conditions is true: <ul style="list-style-type: none">■ a CLOSE WORK FILE statement is issued,■ a DEFINE WORK FILE statement is issued,■ at session termination.

LRECL - Default and Maximum Record Length of Dataset

LRECL=nnn determines the record length (in bytes) of the dataset.

Possible values:	0 or 5 - 32767
Default value:	0

This subparameter is used particularly to check for truncation and padding. For more information on AM=STD, see the keyword subparameter LRECL in the section [WORK Keyword Subparameters for AM=STD in All Environments](#) below.

TRUNC - Truncation of Output Records

TRUNC=xxx determines whether the output records are truncated or not:

ON	Output records that are longer than the record length (LRECL) of the dataset will be truncated.
OFF	Error NAT1512 will be issued if an output record is longer than the dataset record length. This is the default value.

PAD - Padding of Output Records

PAD=xxx determines whether the output records are padded or not (applies only to datasets of fixed record length):

ON	Output records that are shorter than the record length (LRECL) of the dataset will be padded with padding characters defined by keyword subparameter PADCHRO . This is the default value.
OFF	Error NAT1510 will be issued if an output record is shorter than the dataset record length.

PADCHRO - Padding Character of Output Records

This subparameter defines the character which is used for padding of output records if PAD=ON is defined for the work file.

Possible values:	'x '	(one character x within single quotes)
	x'xx'	(one hex character xx)
Default value:	x'00'	

PADCHRI - Padding Character of Input Records

This subparameter defines the character which is used for padding of input records.

Possible values:	'x '	(one character x within single quotes)
	x'xx'	(one hex character xx)
Default value:	x'40'	(blank)

Keyword Subparameters for AM=STD in All Environments

The following keyword subparameters are available: [RECFM](#) | [BLKSIZE](#) | [LRECL](#)

RECFM - Default Record Format of Dataset

RECFM=xxxx determines the default record format of the dataset.

The following formats are supported:

F	Fixed
V	Variable
U	Undefined
B	Blocked
S	Spanned
A	ASA
M	Machine control characters

The following values and also combinations of values are possible:

Possible value:	F, FA, FM, FB, FBA, FBM, V, VA, VM, VB, VBA, VBM, VBS, VBSA, VBSM, U, UA, UM
Default value:	RECFM=VB (variable blocked).

The RECFM specification only applies if no record format is predefined in the JCL or (z/OS only) in the dataset DCB.

BLKSIZE - Default Block Size of Dataset

BLKSIZE=nnnnn determines the default block size (in bytes) of the dataset.

Possible values:	0 or 8 - 32767
Default value:	4628

The BLKSIZE specification only applies if no block size is predefined in the JCL or (z/OS only) in the dataset DCB.

LRECL - Default and Maximum Record Length of Dataset

LRECL=nnn determines the record length (in bytes) of the dataset.

Possible values:	0 or 5 - 32767
Default value:	0

This subparameter is used particularly to check for truncation and padding.

- For RECFM=V (B) the LRECL value includes a 4-byte record descriptor word.
- If LRECL=0 is defined, the following applies:
 - With RECFM=V (B), LRECL defaults to BLKSIZE-4.
 - With RECFM=U, LRECL defaults to BLKSIZE.
 - With RECFM=F (B), the maximum record length in the Natural program being executed is taken when the file is opened. If no record length from a program is available when the file is opened, for example with OPEN=INIT, this leads to an error.

The LRECL specification only applies if no record length is predefined in the JCL or (z/OS only) in the dataset DCB.

Keyword Subparameters for AM=STD in z/OS Environments

The following keyword subparameters are available: [REREAD](#) | [FREE](#) | [BUFNO](#) | [DISP](#) | [VMAX](#)

REREAD - Closing of Tape File Datasets

REREAD=xxx sets the REREAD option for the closing of the tape file:

ON	The REREAD option is set for the CLOSE SVC. This causes the volume to be repositioned to reprocess the dataset. This is the default value.
OFF	The REREAD option is not set for the CLOSE SVC.

FREE - Dataset De-allocation at File Closure

FREE=xxx determines whether the dataset is de-allocated when the file is closed:

ON	The FREE option is set for the CLOSE SVC, which means that the dataset is de-allocated when it is closed (and not at step termination).
OFF	The FREE option is not set for the CLOSE SVC. This is the default value.

BUFNO - Default Number of z/OS I/O Buffers of Dataset

BUFNO=nnn defines the default number of z/OS I/O buffers of the dataset.

Possible values	0 - 255
Default value	0 In this case, z/OS allocates five I/O buffers per default.

The number of I/O buffers can improve the performance of work file access dramatically. Note that the storage for I/O buffers is allocated below the 16 MB line.

The BUFNO specification applies only if the BUFNO parameter is not specified in the JCL for the dataset.

DISP - Open Work File for Modification

DISP=xxx determines that the work file is opened for modification.

This corresponds to the JCL DD statement subparameter DISP=MOD.

MOD	New records are added at the end of the file.
NOMOD	The work file is rewritten from the start. This is the default value.

VMAX - Control LRECL for Variable Record Format

VMAX=xxx controls the LRECL setting for an output file with variable record format (RECFM=V).

ON	Providing a nonzero BLKSIZE value exists for the file, VMAX=ON sets LRECL=BLKSIZE - 4 for variable record format, regardless of the LRECL setting in the DCB or the LRECL subparameter.
NAT	LRECL is set to the length +4 of the largest record in the application program if this value is less than LRECL in the DCB for the dataset.
OFF	LRECL from the DCB for the dataset is used. This is the default value.

Keyword Subparameters for AM=STD in z/VSE Environments

The following keyword subparameters are available: [SYSNR](#) | [LABEL](#) | [REWIND](#) | [BLOCKS](#) | [DISP](#)

SYSNR - Logical VSE SYS Number

SYSNR=nn determines the logical VSE SYS number.

Possible values:	1 - 99
Default value:	By default, the SYS number is identical to the work file number.

LABEL - Tape Label Processing

LABEL=xxx d determines the tape label processing:

ON	The tape is in standard label format. This is the default value.
OFF	The tape is unlabeled with front tape mark.
NOTM	The tape is unlabeled without front tape mark.

REWIND - Action at File Closure

REWIND=xxx determines the action to be taken when a tape file is closed:

ON	The tape is rewound when the file is closed. This is the default value.
OFF	The tape is not rewound when the file is closed.
UNLOAD	The tape is unloaded when the file is closed.

BLOCKS - Number of Storage Blocks

BLOCKS=nnnn specifies the number of file blocks or file tracks to be allocated for a dynamic NATVSE work file.

Possible values:	1 - 9999
Default value:	20

See *NATVSE Dynamic Work File Allocation (DYNALLOC) Support* in the *Operations* documentation.

DISP - Work File Disposition for VSAM/SAM

DISP=(xxx,xxx) specifies the disposition of a dynamic NATVSE work file controlled by VSAM/SAM.

Possible value pairs are:

(NEW,KEEP)	File is to be reset at OPEN and to be kept at CLOSE. This is the default value.
(NEW,DELETE)	File is to be reset at OPEN and to be made inaccessible at CLOSE.
(OLD,DELETE)	File is not to be reset at OPEN and to be made inaccessible at CLOSE.
(OLD,KEEP)	File is not to be reset at OPEN and to be kept at CLOSE.

See *NATVSE Dynamic Work File Allocation (DYNALLOC) Support* in the *Operations* documentation.

Keyword Subparameters for AM=STD in BS2000/OSD Environments

The following keyword subparameter is available: DISP

DISP - File Open Mode

DISP=xxx determines the open mode of the file:

EXT	The open mode is set to EXTEND.
NOEXT	The open mode is set to the default value OUTPUT. This is the default value.

Keyword Subparameters for AM=CICS

The following keyword subparameters are available: [TYPE](#) | [DISP](#)

TYPE - Type of CICS Storage Medium

TYPE=xxxx specifies the type of CICS storage medium to be used:

MAIN	Temporary main storage.
AUX	Temporary auxiliary storage.
TD	Transient data.

The default value used depends on the [DEST](#) keyword subparameter setting. If the [DEST](#) subparameter value matches a valid CICS transient data queue, the [TYPE](#) subparameter defaults to TD, otherwise MAIN will be taken as the default value.

DISP - CICS Temporary Storage Queue Disposition

DISP=(xxx,xxx) specifies the CICS temporary storage queue disposition.

Possible value pairs are:

(NEW,KEEP)	The storage queue is deleted when the file is opened. This is the default value.
(NEW,DELETE)	The storage queue is deleted when the file is opened and when it is closed.
(OLD,DELETE)	The storage queue is deleted when the file is closed.
(OLD,KEEP)	The storage queue is not deleted.



Note: The [DISP](#) specification does not apply to CICS extra-partition transient data queues.

Keyword Subparameters for AM=COMP

The following keyword subparameters are available: [TYPE](#) | [BLOCKS](#) | [BLKSIZE](#)

TYPE - Type of Storage Access

TYPE=xxx specifies the type of storage access to be used:

SHR	Shared access, that is, the work file is accessible by all users.
TID	The work file is only available to the current Com-plete terminal ID.
DYN	The work file is only available to the current terminal stack level.

BLOCKS - Number of Storage Blocks

BLOCKS=nnnn specifies the number of storage blocks to be allocated.

Possible values:	1 - 9999
Default value:	20

BLKSIZE - Size of Storage Blocks

BLKSIZE=nnnn determines the default block size (in bytes) of the dataset.

Possible values:	0, or 8 - 32767
Default value:	4628

Keyword Subparameters for AM=SMARTS

The following keyword subparameters are available: [DEST](#) | [TYPE](#) | [DISP](#)

DEST - Work File Name

DEST=*name* specifies the workfile name (1-8 characters).

Since the DEST clause is restricted to an 8 character maximum, it is useless to define a file with absolute PFS path specification.

The name specified in the DEST clause is relative to the workfile root directory. The work file root directory is specified with the environment variable NAT_WORK_ROOT.

To specify a file with absolute path definition, the DEFINE WORK FILE statement must be used.

TYPE - Type of Storage Access

TYPE=*xxx* specifies the type of storage access to be used. Possible values are:

BIN	Each line is written to the work file without terminating end-of-line character. This is the default value.
TXT	Each line is written to the work file with a terminating end-of-line character (x'15').

DISP - File Open Mode

DISP=(*Disp1,Disp2,Disp3*) specifies the mode of the work file. Possible values are:

<i>Disp1=xxx</i>	Specifies whether an existing file should be deleted or new data should be appended to the file.	
	NEW	An existing file will be deleted if the file is opened for writing. This is the default value.
	OLD or MOD	New data written are appended at the end of the file.
<i>Disp2=xxx</i>	Specifies whether a file should be kept or removed after access.	
	KEEP	Permanent file that will be kept after close. This is the default value.
	DELETE	Temporary file that will be removed after close.
<i>Disp3=xxx</i>	Specifies whether a user has exclusive access to the file or not.	
	SHR	Shared access, that is, the work file is accessible by all users. This is the default value.
	OWN	Exclusive access, the work file is accessible to the current Comp-lete user ID. Files with exclusive access are located in an additional directory which has the name of the current user ID.

275

WPSIZE - Sizes of Natural Work Pools

This Natural profile parameter specifies the sizes of the Natural work pools below and above the 16 MB line for one Natural session.

Natural uses work pools below and above the 16 MB line. In these work pools, all temporary buffers physical storage requests are satisfied.

Natural uses physical storage in special situations only, for example, for passing parameter areas outside the thread (while the thread is released) during the execution of the `CALL` statement with the “call by value option” indicated by a `SET CONTROL 'P=V'` statement under CICS.

The advantage of work pools is that, if there are many requests for physical storage, Natural can satisfy these requests by itself rather than by passing it to the operating system.

Possible settings	<i>size-below</i>	<i>size-below</i> (0-1024) is the size of one work pool in KB below the 16 MB line. If the work pool is exhausted, another work pool with the same size is allocated. The value 0 means that no work pool is allocated, i.e. all requests for physical storage below 16 MB are passed directly to the operating system.
	<i>size-above</i>	<i>size-above</i> (0-16384) is the size of one work pool in KB above the 16 MB line. If the work pool is exhausted, another work pool with the same size is allocated. The value 0 means that no work pool is allocated, that is, all requests for physical storage above 16 MB are passed directly to the operating system.
	<i>maximum-below</i>	<i>maximum-below</i> (0-2097151) limits the total physical storage in KB which can be allocated below the 16 MB line. The value 0 means no physical storage can be allocated below the 16 MB line.
	<i>maximum-above</i>	<i>maximum-above</i> (0-2097151) limits the total physical storage in KB which can be allocated above the 16 MB line.

		The value 0 means no physical storage can be allocated above the 16 MB line.
Default setting	(32,128,2097151,2097151)	
Dynamic specification	yes	
Specification within session	no	

The WPSIZE parameter is specified as follows:

WPSIZE=(size-below,size-above,maximum-below,maximum-above)

Subparameters not to be changed can be omitted, e.g. you can specify WPSIZE=(,1000) if you want to set the work pool size only above 16MB to 1000 KB.

Natural allocates the work pools outside the Natural storage thread according to the specified settings. A work pool is allocated during the first request for physical storage and is released during the next terminal I/O.

For non-thread environments (e.g. batch, TSO), the recommended setting is WPSIZE=(0,0). This may save virtual storage.

276

WSISIZE - Buffer for Natural Workstation Interface

This Natural profile parameter only applies if Natural Workstation Interface is installed.

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

Possible settings	10 - 256	Size of buffer area in KB. If the required space is not available, the Natural Workstation Interface cannot be used.
	0	The Natural Workstation Interface cannot be used.
Default setting	0	
Dynamic specification	yes	
Specification within session	no	

Statements

▪ XML Parameter Syntax	710
▪ NTXML Macro Syntax	711
▪ Keyword Subparameters	711
▪ Example of NTXML Macro	713
▪ Example of XML Parameter	713

This Natural profile parameter is used to activate/deactivate the statements `REQUEST DOCUMENT` and `PARSE XML`.

This profile parameter corresponds to the `NTXML` macro in the parameter module `NATPARM`.



Note: As a prerequisite for using the XML profile parameter, the profile parameter `CFICU` must be set to `CFICU=ON`.

See also *Statements for Internet and XML Access* in the *Programming Guide*.

Possible settings	ON	Enable XML support according to the subparameter settings below. For the <code>PARSE XML</code> and <code>REQUEST DOCUMENT</code> statement usage, the subparameters <code>RDOC</code> and <code>PARSE</code> must be set to ON as well.
	OFF	Disable XML support. Any subparameter settings are ignored.
	See Keyword Subparameters below.	In addition, the following keyword subparameters are available: <code>RDOC</code> <code>PARSE</code> <code>RDCP</code> <code>RDP</code> <code>RDPPORT</code> <code>RDPS</code> <code>RDSPORT</code> <code>RDNOP</code>
Default setting	OFF	
Dynamic specification	yes	The parameter XML can only be specified dynamically. In <code>NATPARM</code> , use the macro <code>NTXML</code> .
Specification within session	no	

The following topics are covered below:

XML Parameter Syntax

The XML parameter is specified as follows:

$$\text{XML} = \left(\begin{Bmatrix} \text{ON} \\ \text{OFF} \end{Bmatrix} , \text{keyword_subparameter1} = \text{value}, \text{keyword_subparameter2} = \text{value}, \dots \right)$$

For names and values of *keyword_subparameters*, see [Keyword Subparameters](#) below.

NTXML Macro Syntax

The NTXML macro is specified as follows:

```

.....1.....2.....3.....4.....5.....6.....7..
      NTXML  ON/OFF
              RDOC=ON/OFF,
              PARSE=ON/OFF,
              RDCP=code-page-name,
              RDP=url,
              RDPPORT=port-number,
              RDPS=url,
              RDSPORT=port-number,
              RDNOP=domain-name

```



Note: The keyword subparameters RDPPORT and RDPS are currently for z/OS only.

Keyword Subparameters

[RDOC](#) | [PARSE](#) | [RDCP](#) | [RDP](#) | [RDPPORT](#) | [RDPS](#) | [RDSPORT](#) | [RDNOP](#)

RDOC - Support of REQUEST DOCUMENT Statement

Possible values are:

ON	Use of the REQUEST DOCUMENT statement is supported.
OFF	Use of the REQUEST DOCUMENT statement is not supported. This is the default value.

PARSE - Support of PARSE XML Statement

Possible values are:

ON	Use of the PARSE XML statement is supported.
OFF	Use of the PARSE XML statement is not supported. This is the default value.

RDCP - Name of the Default HTML/XML Code Page

Specifies the default code page which is assumed if *operand15* in the REQUEST DOCUMENT statement contains only spaces.

Possible values	<i>code-page-name</i>
Default value	ISO 8859-1:1987

RDP - URL of Proxy Server

Specifies the URL of the proxy server through which all internet (not intranet) HTTP requests have to be routed.

Possible values	<i>url</i>
Default value	OFF

Blanks are not allowed. The value OFF means that no URL is defined.

RDPPORT - Proxy Port Number

Specifies the port number of the proxy, if any is set.

Possible values	0 - 65535
Default value	80

RDPS - URL of SSL Proxy Server

Specifies the URL of the SSL proxy server through which all internet (not intranet) HTTPS requests have to be routed.



Note: This keyword subparameter is currently for z/OS only.

Possible values	<i>url</i>
Default value	OFF

Blanks are not allowed. The value OFF means that no URL is defined.

RDSPORT – SSL Proxy Port Number

Specifies the port number of the SSL proxy, if any is set.



Note: This keyword subparameter is currently for z/OS only.

Possible values	0 - 65535
Default value	443

RDNOP - Name of Local Domain

Specifies local domain(s) which are to be addressed directly, not via the proxy.

Possible values	<i>domain-name(s)</i>
Default value	OFF

Blanks are not allowed. The value OFF means that no URL is defined.

Wildcard notation for prefixes can only be used in the form *.xxx and not in the form .xxx.

Multiple entries are separated by a semicolon.

Example of NTXML Macro

```
NTXML
ON,RDP=HTTPPROXY.MYCOMPANY.COM,RDPPORT=8080,RDPS=SSLPROXY.MYCOMPANY.COM,RDSPORT=443,RDNOP=*.MYCOMPANY.COM,RDOC=ON,PARSE=ON
```



Note: The keyword subparameters RDPPORT and RDPS are currently for z/OS only.

Example of XML Parameter

```
XML=(ON,RDP='HTTPPROXY.MYCOMPANY.COM',RDPPORT=8080,RDPS='SSLPROXY.MYCOMPANY.COM',RDSPORT=443,RDNOP='*.MYCOMPANY.COM',RDOC=ON,PARSE=ON)
```



Note: The keyword subparameters RDPPORT and RDPS are currently for z/OS only.

278

XREF - Creation of XRef Data for Natural

■ Extended XRef Data Generation (For Internal Use Only)	717
---	-----

This Natural profile parameter is used to enable/disable the creation of XRef data for Natural. These are 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.

This parameter controls the compilation in two aspects:

- generation of XRef data in the cases described above and
- to fulfil 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).

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	<code>ON</code>	XRef data are generated in the cases described above. Documentation premise is not checked.	
	<code>OFF</code>	XRef data are not generated. Documentation premise is not checked.	
	<code>FORCE</code>	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.	
	<code>DOC</code>	A Natural object can only be cataloged if a documentation object already exists for this object. XRef data are not generated.	
Default setting	<code>OFF</code>		
Dynamic specification	<code>yes</code>		
Specification within session	<code>yes</code>	Applicable Statements:	<code>none</code>
		Applicable Commands:	<code>XREF</code>

There are different ways to set the Natural `XREF` parameter:

- In the Natural parameter module.
- 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.

Extended XRef Data Generation (For Internal Use Only)



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

The extended `XREF` parameter has the following syntax:

```
XREF=(normal-xref-setting,extended-xref-setting)
```

279

XSIZE - Size of Buffer for User Subsystem

This Natural profile parameter specifies the size of the buffer area to be used for user subsystems called by Natural programs.

Alternatively, you can use the equivalent Natural profile parameter [DS](#) or the macro [NTDS](#), see *Using Optional Macros in a Natural Parameter Module* in the *Operations* documentation to specify the XSIZE value.



Caution: If Natural Connection is installed and asynchronous lines are used, the XSIZE profile parameter is reserved for internal use by Natural Connection and must not be used otherwise.

Possible settings	1 - 64	Size of the buffer area in KB.
	0	If XSIZE=0 or if the required space is not available, the user subsystem cannot be used.
Default setting	0	
Dynamic specification	yes	
Specification within session	no	

280

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 1583 through 2699. If the profile parameter [MAXYEAR](#) is set to 9999, the upper year limit is 9999.

281

YSLW - Year Sliding or Fixed Window

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

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	<p>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.</p> <p>Example:</p> <p>If the current year is 2002 and you specify YSLW=40, the sliding window will cover the years 1962 to 2061. A 2-digit year setting <i>nn</i> from 62 to 99 is then interpreted accordingly as 19<i>nn</i>, while a 2-digit year setting <i>nn</i> from 00 to 61 is interpreted as 20<i>nn</i>.</p>
	Fixed Window	1582 - 2600	<p>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.</p> <p>Example:</p> <p>If you specify YSLW=1985, the fixed window will cover the years 1985 to 2084. A 2-digit year setting <i>nn</i> from 85 to 99 is then interpreted accordingly as 19<i>nn</i>, while a 2-digit year setting <i>nn</i> from 00 to 84 is interpreted as 20<i>nn</i>.</p>
Default setting	0		No sliding or fixed-window mechanism is used.

Dynamic specification	yes	
Specification within session	no	

The `YSLW` parameter is evaluated at runtime when an alphanumeric date setting with a 2-digit year component is moved into a date variable. This applies to data settings which are:

- used with the mathematical function `VAL`;
- used with the `IS(D)` option in a logical condition;
- read from the stack as input data;
- or entered in a map as input data.

See also the section *Processing of Date Information* in the *Programming Guide*.

282

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.

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

Possible settings	ON	Natural issues an error message if a division by 0 is attempted.	
	OFF	Natural returns a result of 0 for any division operation in which the divisor is 0.	
Default setting	ON		
Dynamic specification	yes		
Specification within session	yes	Applicable Statements:	SET GLOBALS
		Applicable Command:	GLOBALS
Application Programming Interface	USR1005N	See SYSEXT - <i>Natural Application Programming Interfaces</i> in the Utilities documentation.	

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

283

ZP - Zero Printing

This Natural profile and session parameter specifies how a field which contains a setting of all zeros is to be output; that is, it is used to suppress the display of a numeric field (format N, I, P or F) or time field (format T) which contains a value of all zeros.

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

Possible settings	ON	Each field value which consists of all zeros is output as one zero, right justified (for numeric fields) or all zeros (for time fields).	
	OFF	Each field value which consists of all zeros is suppressed.	
Default setting	ON		
Dynamic specification	yes		
Specification within session	yes	Applicable Statements:	DISPLAY FORMAT INPUT PRINT REINPUT SET GLOBALS WRITE
		Applicable Command:	GLOBALS
Application Programming Interface	USR1005N	See SYSEXT - <i>Natural Application Programming Interfaces</i> in the <i>Utilities</i> documentation.	

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

284

ZSIZE - Size of Entire DB Buffer Area

This Natural profile parameter only applies to Entire DB. It specifies the size of the buffer area required by Entire DB.

Alternatively, you can use the equivalent Natural profile parameter [DS](#) or macro [NTDS](#), see *Using Macros in a Natural Parameter Module* in the *Operations* documentation to specify the `ZSIZE` value.

Possible settings	1 - 64	Size of the buffer area in KB.
	0	If <code>ZSIZE=0</code> or if the required space is not available, the Entire DB Interface cannot be used.
Default setting	0	
Dynamic specification	yes	
Specification within session	no	

Index

P

- parameter
 - overview, 1
- profile parameter
 - overview, 1

S

- session parameter
 - overview, 1
