

Natural für Großrechner

Parameter-Referenz

Version 4.2.6 für Großrechner

Februar 2010

Dieses Dokument gilt für Natural für Großrechner ab Version 4.2.6 für Großrechner.

Hierin enthaltene Beschreibungen unterliegen Änderungen und Ergänzungen, die in nachfolgenden Release Notes oder Neuausgaben bekanntgegeben werden.

Copyright © 1979-2010 Software AG, Darmstadt, Deutschland und/oder Software AG USA, Inc., Reston, VA, Vereinigte Staaten von Amerika, und/oder ihre Lizenzgeber..

Der Name Software AG, webMethods und alle Software AG Produktnamen sind entweder Warenzeichen oder eingetragene Warenzeichen der Software AG und/oder der Software AG USA, Inc und/oder ihrer Lizenzgeber. Andere hier erwähnte Unternehmens- und Produktnamen können Warenzeichen ihrer jeweiligen Eigentümer sein.

Die Nutzung dieser Software unterliegt den Lizenzbedingungen der Software AG. Diese Bedingungen sind Bestandteil der Produktdokumentation und befinden sich unter <http://documentation.softwareag.com/legal/> und/oder im Wurzelverzeichnis des lizenzierten Produkts.

Diese Software kann Teile von Drittanbieterprodukten enthalten. Die Hinweise zu den Urheberrechten und Lizenzbedingungen der Drittanbieter entnehmen Sie bitte den "License Texts, Copyright Notices and Disclaimers of Third Party Products". Dieses Dokument

ist Bestandteil der Produktdokumentation und befindet sich unter <http://documentation.softwareag.com/legal/> und/oder im Wurzelverzeichnis des lizenzierten Produkts.

Inhaltsverzeichnis

1	Parameter-Referenz	1
2	Profilparameter — Einführung	3
	Benutzung von Profilparametern	4
	Gemeinsame Beschreibungen für Profilparameter und Session-Parameter	4
3	Session-Parameter — Einführung	5
	Benutzung von Session-Parametern	6
	Setzen von Session-Parametern	6
	Verarbeitung von Session-Parametern	8
4	ACIVERS - Define ACI Version for Use with EntireX Broker ACI	11
5	AD - Attribute Definition	13
	AD-Parameter-Syntax	14
	Feldanzeige	15
	Ausrichtung der Feldwerte	16
	Feldeingabe/ausgabe-Charakteristika	16
	Auslegung alphanumerischer Felder	18
	Eingabezwang	18
	Mindestlänge der Eingabewerte	18
	Groß-/Kleinschreibung	19
	Füllzeichen	19
6	ADAMODE - Adabas Call Interface Mode	21
7	ADANAME - Name of Adabas Link Routine	25
8	ADAPRM - REVIEW/DB Support	27
9	ADASBV - Adabas Security by Setting	29
10	AL — Alphanumeric Length for Output	31
11	ASIZE - Entire System Server Auxiliary Buffer	33
12	ASPSIZE (Internal Use)	35
13	ASYNNAM - Output System ID for Asynchronous Processing	37
14	ATTN - Attention Key Interrupt Support	39
15	AUTO - Automatic Logon	41
16	AUTORPC - Automatic Natural RPC Execution	43
17	BPC64 - Buffer Pool Cache Storage Type	45
18	BPCSIZE - Cache Size for Natural Buffer Pool	47
19	BPI - Buffer Pool Initialization	49
	BPI Parameter Syntax	50
	NTBPI Macro Syntax	51
	Keyword Subparameters	51
	Examples of NTBPI Macros	56
	Examples of BPI Parameter	56
20	BPLIST - Name of Preload List for Natural Buffer Pool	59
21	BPMETH - Buffer Pool Space Search Algorithm	61
22	BPNAME - Name of Natural Global Buffer Pool	63
23	BPPROP - Global Buffer Pool Propagation	65
24	BPSFI - Object Search First in Buffer Pool	67

25 BPSIZE - Size of Natural Local Buffer Pool	69
26 BPTXT - Size of Text Segments in Natural Buffer Pool	71
27 BSIZE - Size of EntireX Broker Buffer	73
28 BX — Box Definition (Outlining)	75
29 CANCEL - Session Cancellation with Dump	77
30 CC - Error Processing in Batch Mode	79
31 CCTAB - Printer Escape Sequence Definition	81
CCTAB Parameter Syntax	82
NTCCTAB Macro Syntax	84
String Syntax for OPN, CLS, CODE, CS, CSS or CSE	84
Proportional Fonts	84
Examples of NTCCTAB Macros	85
Examples of CCTAB Parameter	85
32 CD - Color Definition	87
33 CDYNAM - Dynamic Loading of Non-Natural Programs	89
34 CF - Character for Terminal Commands	91
35 CFICU - Unicode and Code Page Support	93
CFICU Parameter Syntax	94
NTCFICU Macro Syntax	95
Keyword Subparameters	95
Example of NTCFICU Macro	98
Example of Dynamic Parameter CFICU	98
36 CFWSIZE (Internal Use)	99
37 CLEAR - Processing of CLEAR Key in NEXT Mode	101
38 CM - Command Mode	103
39 CMPO - Compilation Options	105
CMPO Parameter Syntax	106
NTCMPO Macro Syntax	106
Keyword Subparameters	106
Example of CMPO Parameter	107
Example of NTCMPO Macro	107
40 CMPR - General Default Compression Optimization Algorithm	109
41 COMPR - Set RPC Buffer Compression	111
42 CP - Default Code Page Name	113
43 CPCVERR - Code Page Conversion Error	115
44 CPOBJIN - Code Page of Batch Input File	117
45 CPPRINT - Code Page of Batch Output File	119
46 CPRPC - Define Code Page Name	121
47 CPSYNIN - Code Page of Batch Input File for Commands	123
48 CSIZE - Size of Con-nect/Con-form Buffer Area	125
49 CSTATIC - Programs Statically Linked to Natural	127
Example of CSTATIC Parameter	128
Example of NTCSTAT Macro	129
50 CV - Attribute Control Variable	131
51 CVMIN - Control Variable Modified at Input	133

52 DATSIZE - Minimum Size of Buffer for Local Data	135
53 DB - Database Types and Options	137
DB Parameter Syntax	138
NTDB Macro Syntax	139
Possible Database Options	140
Examples of NTDB Macro	141
Examples of DB Parameter	141
54 DB2SIZE - Natural Buffer Area for DB2 or SQL/DS	143
55 DBCLOSE - Database Close at Session End	145
56 DBGERR - Automatic Start of Debugger at Runtime Error	147
57 DBID - Default Database ID of Natural System Files	149
58 DBOPEN - Database Open without ETID	151
59 DBROLL - Database Calls before Session Suspension	153
60 DBUPD - Database Updating	155
61 DC - Character for Decimal Point Notation	157
62 DD - Day Differential	159
63 DELETE - Deletion of Dynamically Loaded Programs	161
64 DF - Date Format	163
65 DFOUT - Date Format for Output	165
66 DFS - Specify RPC Client's Default Server Address	167
67 DFSTACK - Date Format for Stack	169
68 DFTITLE - Output Format of Date in Standard Report Title	171
69 DL - Display Length for Output	173
70 DLISIZE - Size of Natural Buffer Area for DL/I	175
71 DO - Display Order of Output Data	177
72 DS - Define Size of Storage Buffer	179
DS Parameter Syntax	180
NTDS Macro Syntax	181
Table of Buffer Sizes	181
Examples	182
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-Parametersyntax	195
Beispiele	197
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
Leerzeichen in Editiermasken	219
Standard-Editiermasken	219
Editiermasken für numerische Felder	220
Editiermasken für alphanumerische Felder	223
Editiermasken für binäre Felder – Format B	225
Hexadezimale Editiermasken	225
Editiermasken für Datums- und Zeitfelder (Formate D und T)	227
Editiermasken für logische Felder (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
Ausführung von Helproutinen	308
Beispiele	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
Profilparameter LS	362
Session-Parameter LS	362
Spezifikation in 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 Helproutines	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
Profilparameter 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	446
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	456
Keyword Subparameters for AM=STD in BS2000/OSD Environments	457
Keyword Subparameters for AM=CICS	458
Keyword Subparameters for AM=COMP (Com-plete)	459
Keyword Subparameters for AM=SMARTS (Com-plete)	459
Keyword Subparameters for AM=IMS	460
Keyword Subparameters for DEFINE PRINTER Statement	460
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
Anmerkungen zur Syntax	526
Anmerkungen zur Laufzeitumgebung	527
Funktionen	528
Beschränkungen	530
213 SCTAB - Scanner Characters	531
SCTAB Parameter Syntax	532
NTSCTAB Macro Syntax	533
Example of NTSCTAB Macro	533
Example of SCTAB Parameter	533
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	562
Conversion of Logical Shift-Out/Shift-In Characters	563
Compatibility of SOSI Profile Parameter and Obsolete SO and SI Profile Parameters	563
Automatic Adaptation of Translation Tables	564
SOSI Parameter Examples	564
226 SRETAIN - Retain Source Format	565
227 SRVCMIT - Server Commit Time	567
228 SRVNAME - Name of RPC Server	569
229 SRVNODE - Name of Node	571
230 SRVTERM - Server Termination Event	573
231 SRVUSER - User ID for RPC Server Registry	575
232 SRVWAIT - Wait Time of RPC Server	577
233 SSIZE - Size of Source Area Allocated by the Editors	579
234 STACK - Place Data/Commands on the Stack	581
235 STACKD - Stack Delimiter Character	583
236 STEPLIB - Additional Steplib Library	585

237	SUBSID - Subsystem ID under z/OS and z/VSE	587
238	SYNERR - Control of Syntax Errors	589
239	SYS - Define and Activate a Set of Dynamic Profile Parameters	591
	SYS Parameter Syntax	592
	NTSYS Macro Syntax	593
	Example of NTSYS Macro	593
240	SYSCIP - Adabas Cipher Key for Natural System Files	595
241	SYSPSW - Adabas Password for Natural System Files	597
242	TAB - Standard Output Character Translation	599
	TAB Parameter Syntax	600
	NTTAB Syntax	600
	Example of NTTAB Macro	601
	Example of TAB Parameter	601
243	TAB1 - Alternative Output Translation	603
	TAB1 Parameter Syntax	604
	NTTAB1 Macro Syntax	604
	Example of NTTAB1 Macro	605
	Example of TAB1 Parameter	605
244	TAB2 - Alternative Input Translation	607
	TAB2 Parameter Syntax	608
	NTTAB2 Macro Syntax	608
	Example of NTTAB2 Macro	609
	Example of TAB2 Parameter	609
245	TABA1 - EBCDIC-to-ASCII Translation	611
	TABA1 Parameter Syntax	612
	NTTABA1 Macro Syntax	612
	Example of NTTABA1 Macro	613
	Example of TABA1 Parameter	613
246	TABA2 - ASCII-to-EBCDIC Translation	615
	TABA2 Parameter Syntax	616
	NTTABA2 Macro Syntax	616
	Example of NTTABA2 Macro	617
	Example of TABA2 Parameter	617
247	TABL - SYS Library Output Translation	619
	TABL Parameter Syntax	620
	NTTABL Macro Syntax	620
	Example of NTTABL Macro	621
	Example of TABL Parameter	621
248	TC - Trailing Characters	623
249	TD - Time Differential	625
250	TF - Translation of Database ID/File Number	627
	TF Parameter Syntax	629
	NTTF Macro Syntax	629
	Example of TF Parameter	629
	Example of NTTF Macro	629




251 THSEPCH - Thousands Separator Character	631
252 TIMEOUT - Wait Time for RPC Server Response	633
253 TMODEL - IBM 3270 Terminal Model	635
254 TPF (Internal Use)	637
255 TQ - Translate Quotation Marks	639
256 TRACE - Define Trace Level for Natural RPC Servers	641
257 TRACE - Define Components to be Traced	643
TRACE Parameter Syntax	644
NTTRACE Macro Syntax	645
Example of TRACE Parameter	645
Example of NTTRACE Macro	645
258 TRANSP - Server Transport Protocol	647
259 TRYALT - Try Alternative Server Address	649
260 TS - Translate Output from Programs in System Libraries	651
261 TSIZE - Size of Buffer for Adabas Text Retrieval	653
262 TTYPE - Terminal Type	655
263 UC - Underlining Character	657
264 UDB - User Database ID	659
265 ULANG - User Language	661
266 UPSI - z/VSE User Program Switches	663
267 USER - Restrict Use of Profile Parameter Strings and Modules	665
USER Parameter Syntax	667
NTUSER Macro Syntax	667
Example of NTUSER Macro	667
Example of USER Parameter	667
268 USERBUF (Internal Use)	669
269 UTAB1 - Lower-to-Upper-Case Translation	671
UTAB1 Parameter Syntax	672
NTUTAB1 Macro Syntax	672
Example of NTUTAB1 Macro	673
Example of UTAB1 Parameter	673
270 UTAB2 - Upper-to-Lower-Case Translation	675
UTAB2 Parameter Syntax	676
NTUTAB2 Macro Syntax	676
Example of NTUTAB2 Macro	677
Example of UTAB2 Parameter	677
271 VSIZE - Size of Buffer Area for Natural/VSAM	679
272 WEBIO - Web I/O Interface Screen Rendering	681
WEBIO Parameter Syntax	682
NTWEBIO Macro Syntax	682
Keyword Subparameters	683
273 WH - Wait for Record in Hold Status	685
274 WORK - Work-File Assignments	687
WORK Parameter Syntax	688
NETWORK Macro Syntax	689

Keyword Subparameters for All Environments	690
Keyword Subparameters for AM=STD in All Environments	694
Keyword Subparameters for AM=STD in z/OS Environments	695
Keyword Subparameters for AM=STD in z/VSE Environments	697
Keyword Subparameters for AM=STD in BS2000/OSD Environments	698
Keyword Subparameters for AM=CICS	699
Keyword Subparameters for AM=COMP	700
Keyword Subparameters for AM=SMARTS	700
275 WPSIZE - Sizes of Natural Work Pools	703
276 WSISIZE - Buffer for Natural Workstation Interface	705
277 XML - Activate PARSE XML and REQUEST DOCUMENT Statements	707
XML Parameter Syntax	708
NTXML Macro Syntax	709
Keyword Subparameters	709
Example of NTXML Macro	711
Example of XML Parameter	711
278 XREF - Creation of XRef Data for Natural	713
Extended XRef Data Generation (For Internal Use Only)	715
279 XSIZE - Size of Buffer for User Subsystem	717
280 YD - Year Differential	719
281 YSLW - Year Sliding or Fixed Window	721
282 ZD - Zero-Division Check	723
283 ZP - Zero Printing	725
284 ZSIZE - Size of Entire DB Buffer Area	727
Stichwortverzeichnis	729

1 Parameter-Referenz

Diese Dokumentation enthält ausführliche Informationen zu den Natural-Profil- und Session-Parametern.

Die Beschreibungen der Profilparameter stehen nur in englischer Sprache zur Verfügung.

	Profilparameter – Einführung	Enthält allgemeine Informationen zur Benutzung von Profilparametern sowie eine Liste mit Referenzen auf verwandte Themen.
	Session-Parameter – Einführung	Enthält allgemeine Informationen zur Benutzung und Verarbeitung von Session-Parametern.
	Parameter in alphabetischer Reihenfolge	Beschreibungen sämtlicher Profilparameter und Session-Parameter in alphabetischer Reihenfolge.

Grundsätzliche Informationen zur Benutzung von Parametern finden Sie in der *Operations*-Dokumentation, siehe *Profile Parameter Usage*. Siehe auch *Using Macros in a Natural Parameter Module*.

2 Profilparameter — Einführung

- Benutzung von Profilparametern 4
- Gemeinsame Beschreibungen für Profilparameter und Session-Parameter 4

Benutzung von Profilparametern

Da die Benutzung der Natural-Profilparameter auf den von Natural unterstützten Plattformen unterschiedlich ist, befinden sich die Informationen zur grundsätzlichen Benutzung der Parameter in den entsprechenden plattformspezifischen Dokumentationen.

Siehe folgende Dokumente in der *Natural Operations*-Dokumentation:

Profile Parameter Usage - Overview

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

Gemeinsame Beschreibungen für Profilparameter und Session-Parameter

Wenn es einen Natural-Session-Parameter mit gleichem Namen und gleicher Funktionalität wie ein Profilparameter gibt, sind die Beschreibungen für diese Parameter in einem gemeinsamen Dokument zusammengefasst.

3 Session-Parameter — Einführung

- Benutzung von Session-Parametern 6
- Setzen von Session-Parametern 6
- Verarbeitung von Session-Parametern 8

Benutzung von Session-Parametern

In Natural werden Session-Parameter verwendet:

- um bestimmte Zeichen festzulegen,
- um Verarbeitungszeitgrenzen festzulegen,
- um Reaktionen auf bestimmte Bedingungen festzulegen,
- um bestimmte Maximal- bzw. Minimalgrößen festzulegen,
- um bestimmte Aspekte der Reportformatierung festzulegen.

Bei der Installation von Natural legt der Natural-Administrator für einige dieser Parameter bestimmte Standardwerte fest, die dann für alle Natural-Benutzer gelten.

Die für Ihre Natural-Session gültigen Parameterwerte können Sie sich ansehen, wenn Sie das Systemkommando `GLOBALS` eingeben. Nähere Einzelheiten hierzu finden Sie in der *Systemkommandos*-Dokumentation).

Setzen von Session-Parametern

Es gibt verschiedene Möglichkeiten, die Werte der Session-Parameter zu setzen:

- über das Standard-Parametermodul (`NATPARM`), das bei der Installation von Natural angelegt wird;
- über dynamische Parameter beim Aufrufen einer Natural-Session (wie in der *Operations*-Dokumentation beschrieben);
- mit dem Systemkommando `GLOBALS`;
- mit dem Statement `SET GLOBALS` (nur im Reporting Mode);
- mit dem Statement `FORMAT`;
- mit Statements wie `INPUT`, `DISPLAY`, `WRITE`, in denen bestimmte Parameter verarbeitet werden;
- über Terminalkommandos.

Statt der Parameterwerte `ON` und `OFF` können Sie wahlweise auch die Werte `T` (True = wahr) bzw. `F` (False = falsch) verwenden.

Ändern von Parameterwerten auf Session-Ebene — Das GLOBALS-Kommando

Einige der vom Natural-Administrator festgelegten Standardwerte können Sie für die Dauer einer Natural-Session ändern.

Sie können die Parameterwerte innerhalb einer Session ändern, indem Sie das folgende Systemkommando verwenden:

```
GLOBALS
```

Wenn Sie das GLOBALS-Kommando ausführen, erhalten Sie einen Schirm, auf dem die für Ihre laufende Session derzeit gültigen Parameterwerte angezeigt werden. Auf diesem Schirm können Sie auch die Werte ändern, die nicht Ihren Anforderungen entsprechen.

Die mit einem GLOBALS-Kommando festgelegten Parameterwerte gelten bis zum Ende der jeweiligen Natural-Session (und gelten für jedes im Laufe der Session gespeicherte Objekt), es sei denn, Sie ändern sie wieder durch ein anderes GLOBALS-Kommando.

Ändern von Parameterwerten auf Programm-Ebene — Das FORMAT-Statement

Einige Parameter-Standardwerte können Sie für die Dauer eines Programms (Reports) ändern. Dies geschieht mit einem FORMAT-Statement im jeweiligen Programm, welches die betreffenden session-weit gültigen Parameter überschreibt.

Beispiel für ein FORMAT-Statement:

```
FORMAT AL=10 HC=R
```

Die mit einem FORMAT-Statement angegebenen Parameterwerte gelten bis zum Ende des ausgeführten Programms, wenn sie nicht vorher im Programm durch ein anderes FORMAT-Statement geändert werden. Sie können allerdings nur einen Teil der Session-Parameter programmspezifisch umsetzen.

Andererseits gibt es eine Reihe von Parametern, die Sie nur auf Programm-Ebene setzen können, nicht aber auf Session-Ebene. Dies sind überwiegend Parameter, die sich auf die Formatierung eines Reports auswirken.

Ändern von Parameterwerten auf Statement-Ebene

Die meisten Parameter, die Sie mit einem `FORMAT`-Statement festlegen können, können Sie auch auf Statement-Ebene setzen, zum Beispiel bei einem `DISPLAY`-, `WRITE`-, `INPUT`- oder `REINPUT`-Statement, und zwar durch Einfügen des Parameters (in Klammern) hinter dem Statement-Namen.

Beispiel:

```
DISPLAY (SF=4) NAME JOB-TITLE CURR-CODE SALARY
```

Ein auf Statement-Ebene angegebener Parameterwert gilt nur für das jeweilige Statement, und zwar vor allen auf anderen Ebenen für diesen Parameter angegebenen Werten.

Ändern von Parameterwerten auf Feld-Ebene

Innerhalb eines `DISPLAY`-, `WRITE`-, `INPUT`- oder `REINPUT`-Statements können Sie bestimmte Parameterwerte noch feldspezifisch festlegen, indem Sie den Parameterwert für ein einzelnes Feld oder Ausgabe-Element unmittelbar hinter dem betreffenden Feld (in Klammern) angeben.

Beispiel:

```
DISPLAY NAME (AL=10) JOB-TITLE CURR-CODE SALARY
```

Dieser Wert gilt dann nur für dieses Feld, und zwar vor allen auf anderen Ebenen für diesen Parameter angegebenen Werten. Die feldspezifische Angabe von Parameterwerten ist allerdings nur für einen Teil der auf Statement-Ebene beeinflussbaren Parameter möglich.

Verarbeitung von Session-Parametern

Session-Parameter, die in den Statements `DISPLAY`, `FORMAT`, `PRINT`, `INPUT`, `REINPUT`, `WRITE`, `WRITE TITLE` oder `WRITE TRAILER` gesetzt werden, werden während der *Programmkompilierung* verarbeitet und sind daher in dem betreffenden Objektmodul enthalten.

Bei der Auswertung der Parameterwerte gilt folgende Hierarchie:

1. für einzelne Felder/Elemente gesetzte Parameter (höchste Priorität)
2. auf Statement-Ebene gesetzte Parameter
3. mit einem `FORMAT`-Statement gesetzte Parameter
4. die Standardwerte der Parameter (niedrigste Priorität)

Mit einem `SET GLOBALS`-Statement gesetzte Parameter bewirken eine Veränderung der Laufzeit-Umgebung. Sie gelten solange, bis sie mit einem weiteren `SET GLOBALS`-Statement (oder `GLOBALS`-Systemkommando) geändert werden.

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	14
▪ Feldanzeige	15
▪ Ausrichtung der Feldwerte	16
▪ Feldeingabe/ausgabe-Charakteristika	16
▪ Auslegung alphanumerischer Felder	18
▪ Eingabezwang	18
▪ Mindestlänge der Eingabewerte	18
▪ Groß-/Kleinschreibung	19
▪ Füllzeichen	19

Attribut-Definition

Mit diesem Session-Parameter definieren Sie Feldattribute auf Feld/Element- oder Statement-Ebene.

Verwandter Session-Parameter: [CD](#) - Farbdefinition.

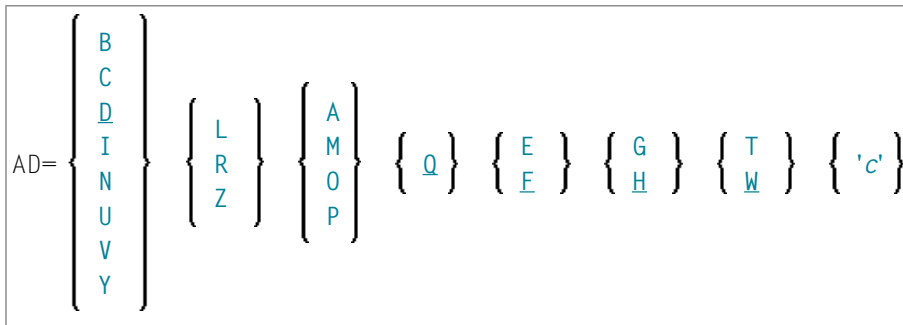
Mögliche Werte	Siehe unten.	Sie können mehrere Attribute in beliebiger Reihenfolge angeben.
Standard-Einstellung	Siehe unten.	
Gültige Statements	FORMAT	
	DISPLAY INPUT NEWPAGE WITH TITLE PRINT REINPUT WRITE WRITE TITLE WRITE TRAILER	Parameter können auf Statement-Ebene und/oder Element-Ebene angegeben werden.
	ASSIGN CALLNAT CALLDBPROC COMPUTE MOVE PERFORM SEND METHOD	Parameter können auf Element-Ebene angegeben werden, allerdings können nur die in der betreffenden Statement-Beschreibung angegebenen Attribute benutzt werden.
Gültiges Kommando	Keines.	

Die Parameter-Syntax und die Bedeutung der Attribute und die einzelnen Werte werden in den folgenden Abschnitten erklärt.

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]
```

Sie können mehrere Attribute in beliebiger Reihenfolge angeben. Mögliche Werte sind:



Die Bedeutung der Attribute und die einzelnen Werte werden in den folgenden Abschnitten erklärt.

Beispiele:

```
DISPLAY #FIELDA (AD=R)
INPUT #FIELDB (AD=M)
INPUT (AD=IM) #FIELDA #FIELDB
```

Feldanzeige

Wert	Bedeutung	Statements	Erklärung
B	blinkend (*)	ASSIGN	Der Feldwert wird blinkend angezeigt.
C	kursiv (*)	COMPUTE	Der Feldwert wird kursiv angezeigt.
D	nicht intensiviert	MOVE DISPLAY FORMAT	Der Feldwert wird normal angezeigt, d.h. weder intensiviert noch sonstwie hervorgehoben. Dies ist die Voreinstellung.
I	intensiviert	INPUT	Der Feldwert wird intensiviert, d.h. hell leuchtend angezeigt.
N	nicht sichtbar	PRINT	Ein eingegebener Feldwert wird nicht angezeigt.
U	unterstrichen	REINPUT WRITE	Der Feldwert wird unterstrichen angezeigt.
V	invers (*)		Der Feldwert wird invers angezeigt, d.h. in farblicher Umkehrung von Hintergrund und Feldwert.
Y	dynamische Attribute	INPUT DISPLAY PRINT WRITE	Feldattribute werden dynamisch über eine Kontrollvariable (Format C) zugewiesen.

* Die mit einem Stern (*) markierten Feldanzeige-Attribute sind an entsprechende Hardware-Voraussetzungen gebunden und werden zur Laufzeit ignoriert, falls diese Voraussetzungen nicht gegeben sind.

Ausrichtung der Feldwerte

Wert	Bedeutung	Statements	Erklärung
L	linksbündig	DISPLAY FORMAT	Feldwerte werden linksbündig im Feld ausgegeben. Dies ist die Voreinstellung für alphanumerische Felder.
R	rechtsbündig	INPUT PRINT REINPUT	Feldwerte werden rechtsbündig im Feld ausgegeben. Dies ist die Voreinstellung für numerische Felder.
Z	vorangestellte Nullen	WRITE	Feld werte werden rechtsbündig im Feld ausgegeben, der Rest des Feldes wird mit Nullen aufgefüllt.

Feldeingabe/ausgabe-Charakteristika

Wert	Bedeutung	Statements	Erklärung
A	Eingabefeld, ungeschützt	INPUT FORMAT	Ein Feldwert wird in Antwort auf ein INPUT-Statement eingegeben. Dies ist die Voreinstellung.
	nur Eingabe	CALLNAT CALLDBPROC PERFORM SEND METHOD	<p>Markieren Sie einen Parameter mit AD=A, wird sein Wert nicht an das aufgerufene Objekt übergeben (Subprogramm, Stored Procedure, Subroutine, Dialog, Methode), sondern er erhält einen Wert von dem aufgerufenen Objekt.</p> <p>Bei einem mit BY VALUE in der Parameter Data Area des aufgerufenen Objekts definierten Feld kann das aufrufende Objekt keinen Wert erhalten. In diesem Fall bewirkt AD=A nur, dass das Feld auf den niedrigeren Wert von dem betreffenden Format zurückgesetzt wird (Leerzeichen für alphanumerisch, binäre Nullen für binär und Nullen für numerische Felder), bevor das Objekt aufgerufen wird.</p> <p>Bei einem CALLNAT-Statement kann AD=A für abgesetzt betriebene Subprogramme nützlich sein, die über Natural RPC in einer Client/Server -Umgebung ausgeführt werden, um die Belastung der gesendeten Daten zu reduzieren. Wenn ein Subprogramm lokal ausgeführt wird, werden mit AD=A markierte Felder auf den niedrigen Wert des betreffenden Formats zurückgesetzt, bevor das Objekt aufgerufen wird.</p> <p>Ist bei einem SEND METHOD-Statement keine Methode in Natural implementiert, ist das Verhalten von der Methoden- Implementierung abhängig. Der Parameter wird dann als initialisierte Variante übergeben. Ob die externe Komponente dann einen Wert zurückgeben kann, ist in der Dokumentation der externen Komponente beschrieben. Sie kann auch im Natural Komponenten- Browser eingesehen werden.</p>

Wert	Bedeutung	Statements	Erklärung
M	Ausgabefeld, änderbar	INPUT FORMAT	Der Wert des Feldes soll bei der Ausführung des INPUT-Statements angezeigt werden, und der ausgegebene Wert kann vom Benutzer überschrieben werden. Das Feld ist ein Ausgabefeld und kann geändert werden.
	änderbar	CALLNAT CALLDBPROC PERFORM SEND METHOD	<p>Standardmäßig kann der übergebene Wert eines Parameters im aufgerufenen Objekt (Subprogramm, Stored Procedure, Subroutine, Dialog, Methode) geändert werden, und der geänderte Wert wird wieder zurück an das aufrufende Objekt übergeben, wo es den ursprünglichen Wert überschreibt.</p> <p>Bei einem mit BY VALUE in der Parameter Data Area des aufgerufenen Objekts definierten Feld wird kein Wert zurückgegeben.</p> <p>Ist bei einem SEND METHOD-Statement keine Methode in Natural implementiert, ist das Verhalten von der Methoden-Implementierung abhängig. Der Parameter wird dann BY REFERENCE übergeben. Ob die externe Komponente einen By Reference- oder By Value-Parameter akzeptiert, ist in der Dokumentation der externen Komponente beschrieben. Dies kann auch im Natural Komponenten-Browser eingesehen werden.</p>
0	Ausgabefeld, geschützt	INPUT FORMAT	Der Wert des Feldes soll bei der Ausführung des INPUT-Statements angezeigt werden. Das Feld ist ein reines Ausgabefeld und kann nicht geändert werden.
	nicht änderbar	CALLNAT CALLDBPROC PERFORM SEND METHOD	<p>Wenn Sie einen Parameter mit AD=0 markieren, kann der übergebene Wert im aufgerufenen Objekt (Subprogramm, Stored Procedure, Subroutine, Dialog, Methode) geändert werden, aber der geänderte Wert kann nicht an das aufrufende Objekt zurückgegeben werden, d.h. das Feld im aufrufenden Objekt behält seinen ursprünglichen Wert bei.</p> <p>Intern wird AD=0 genauso verarbeitet wie BY VALUE (siehe den Abschnitt <i>Definition von Parameterdaten</i> in der Beschreibung des DEFINE DATA-Statements).</p> <p>Wenn bei SEND METHOD eine Methode in Natural implementiert wird, wird der Parameter behandelt, so wie er von BY VALUE in der Parameter Data Area der Methode definiert wurde (siehe den Abschnitt <i>PARAMETER-Klausel</i> in der Beschreibung des INTERFACE-Statements).</p> <p>Wenn bei SEND METHOD keine Methode in Natural implementiert ist, ist das Verhalten abhängig von der Implementierung der Methode. Der Parameter wird dann BY VALUE übergeben. Ob die externe Komponente einen By Reference- oder By Value-Parameter akzeptiert, ist in der Dokumentation der externen Komponente beschrieben. Dies kann auch im Natural Komponenten-Browser eingesehen werden.</p>

Wert	Bedeutung	Statements	Erklärung
P	vorübergehend geschützt (P = protected)	INPUT REINPUT	Wird in Verbindung mit einer Kontrollvariablen (Format C), dem DY -Parameter (dynamische Attribute) und dem REINPUT-Statement verwendet.

Auslegung alphanumerischer Felder

Wert	Bedeutung	Statements	Erklärung
Q	Alphanumerisches Feld so anzeigen als sei es ein numerisches Feld	ASSIGN COMPUTE MOVE DISPLAY FORMAT INPUT PRINT REINPUT WRITE	Dieses Attribut ist nur auf Großrechnern verfügbar. Es ist eine entsprechende Hardware-Funktion erforderlich. Ein alphanumerisches Feld wird interpretiert, als ob es ein numerisches Feld wäre. Wenn das Feld im Bereich des Profil- oder Session-Parameters PM angezeigt wird, wird der Wert des Feldes von links nach rechts anstatt von rechts nach links interpretiert.

Eingabezwang

Wert	Bedeutung	Statements	Erklärung
E	Eingabe erforderlich	INPUT FORMAT	Als Antwort auf ein INPUT-Statement muss ein Feldwert eingegeben werden; andernfalls erscheint eine Fehlermeldung. Dies ist nur bei reinen Eingabefeldern (AD=A) relevant.
F	Wert optional	INPUT FORMAT	Als Antwort auf ein INPUT-Statement kann ein Feldwert eingegeben werden, muss aber nicht. Dies ist die Voreinstellung.

Mindestlänge der Eingabewerte

Wert	Bedeutung	Statements	Erklärung
G	Wertlänge	INPUT FORMAT	Ein als Antwort auf ein INPUT-Statement eingegebener Wert muss genauso lang sein wie das Feld, d.h. das Feld muss vollständig gefüllt werden. Dies ist nur bei reinen Eingabefeldern (AD=A) relevant.
H	Wertlänge	INPUT FORMAT	Ein als Antwort auf ein INPUT-Statement eingegebener Wert darf kürzer sein als das Feld, d.h. das Feld muss nicht vollständig gefüllt werden. Dies ist die Voreinstellung.

Groß-/Kleinschreibung

Wert	Bedeutung	Statements	Erklärung
T	Kleinbuchstaben werden umgesetzt	INPUT FORMAT	Ein eingegebener Wert wird automatisch in Großbuchstaben umgesetzt, d.h. es ist egal, ob ein Wert in Klein- oder Großbuchstaben eingegeben wird.
W	Kleinbuchstaben werden akzeptiert	INPUT FORMAT	Es erfolgt keine Umsetzung von Klein- in Großbuchstaben, d.h. ein Wert wird so interpretiert, wie er eingegeben wird. Um AD=W einzuschalten, müssen Sie den Wert ON für den Natural-Profilparameter LC angeben. Dies ist die Voreinstellung.

Füllzeichen

Wert	Bedeutung	Statements	Erklärung
' c '	filler character	INPUT FORMAT	Der unbeschriebene Teil eines Feldes (für reine Eingabefelder) wird bei der Anzeige mit dem Zeichen c gefüllt, wenn AD=A (reines Eingabefeld, ungeschützt) oder AD=M (Ausgabefeld, änderbar) angegeben wird.

Bevor der Wert für ein änderbares Feld (AD=M) angezeigt wird, werden nicht von dem Wert belegte Feldstellen mit dem angegebenen Füllzeichen wie folgt gefüllt:

- Vorangestellte oder nachfolgende Zeichen werden (abhängig von der Feldausrichtung) bei Feldern mit dem Format I, N und P gefüllt.
- Nachfolgende Zeichen werden für Felder des Formats A gefüllt.

Wenn der Benutzer einen Wert als Antwort auf ein INPUT-Statement eingibt, bevor der Wert dem Feld zugewiesen wurde,

- werden sowohl vorangestellte als auch nachfolgende Füllzeichen bei Feldern mit den Formaten I, N und P entfernt
- werden nachfolgende Füllzeichen bei Feldern des Formats A entfernt.



Vorsicht: Füllzeichen, die auftreten können als Teil des Wertes entweder an vorangestellten oder nachfolgenden Positionen sollten vermieden werden, um ungewünschte Ergebnisse zu verhindern.

Wenn beispielsweise das Füllzeichen 0 (Null) für ein Feld des Formats N5 definiert ist und der Wert 00100 als Eingabedaten eingegeben wird, werden vorangestellte und nachfolgende Nullen entfernt, so dass nur der Wert 1 übrig bleibt und dem Feld zugewiesen wird. Aus demselben

Grund sollte das Minuszeichen (-) als Füllzeichen für numerische Felder vermieden werden, wenn negative Werte eingegeben werden sollen.

Wenn das Füllzeichen auf Leerzeichen (x'40') gesetzt wird, werden auffüllende Leerzeichen durch x'00' ersetzt, damit eine Einfügung von Zeichen ermöglicht wird, ohne dass zuvor der Rest im Eingabefeld gelöscht werden muss.

In BS2000/OSD-Umgebungen werden x'00'-Zeichen als Punkte auf Terminals des Typs 97xx angezeigt. Ihre Erscheinungsform kann mittels der SIDA-Utility oder mit der Konfigurations-Utility der entsprechenden Terminal-Emulation geändert werden.

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.



Vorsicht: 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.



Anmerkung: 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.



Anmerkung: 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

Alphanumerische Länge der Ausgabe

Mit diesem Session-Parameter geben Sie die standardmäßige Ausgabelänge für ein alphanumerisches Feld an, d.h. wenn es kürzer als die Feldlänge ist, wird das Feld rechts abgeschnitten.

Mögliche Werte	1 bis n	n = Wert des LS -Parameters (Zeilenlänge) minus 1).
Standard-Einstellung	none	
Gültige Statements	FORMAT	Parameter können dynamisch mit dem <code>FORMAT</code> -Statement angegeben werden.
	DISPLAY INPUT PRINT WRITE	Parameter können auf Statement-Ebene und/oder Element-Ebene angegeben werden.
Gültiges Kommando	Keines.	



Anmerkungen:

1. Es empfiehlt sich nicht, den Session-Parameter `AL` für Eingabefelder ([Attributdefinition](#) `AD=A` oder `AD=M`) in `INPUT`-Statements zu verwenden.
2. Eine für ein Feld definierte Editiermaske (siehe Session-Parameter [EM](#)) setzt den Session-Parameter `AL` für dieses Feld außer Kraft.

Beispiel:

```
FORMAT AL=20
```

See also *Parameter zur Beeinflussung der Ausgabe von Feldern im Leitfaden zur Programmierung*.

11 ASIZE - Entire System Server Auxiliary Buffer

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



Vorsicht: 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 <code>ASIZE=0</code> 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.



Vorsicht: 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.



Anmerkung: 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=0N is converted into: BPI=(TYPE=NAT,SEQ=0,C64=0N)

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 <code>BPCSIZE=0</code> 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 <code>C64=OFF</code> .
	100 to 58720256	(that is, 56 GB) for cache „above the bar“, that is, with <code>C64=ON</code> .
	Anmerkung: 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 50
- NTBPI Macro Syntax 51
- Keyword Subparameters 51
- Examples of NTBPI Macros 56
- Examples of BPI Parameter 56

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



Anmerkung: 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`, `BPTXT`, `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.



Anmerkung: * 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.



Anmerkung: 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.



Anmerkung: 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.



Anmerkung: 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 **BPTTEXT** (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 **C64** or profile parameter **BPC64** is set to a non-zero value. The C64 specification can be overridden dynamically by profile parameter **BPC64**.

Examples of NTBPI Macros

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

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

Examples of BPI Parameter

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


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

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

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

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

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

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

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

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

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>Vorsicht: 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.



Anmerkungen:

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

BPTTEXT - 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 `BPTTEXT` 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 `BPTTEXT` parameter only affects the very first Natural session, which initializes the local buffer pool.

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

Example:

```
BPTTEXT=4
```

is converted into:

```
BPI=(TYPE=NAT,SEQ=0,XTSIZE=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	0 - 64	Buffer size in KB.
Default setting	0	
Dynamic specification	yes	
Specification within session	no	

28

BX — Box Definition (Outlining)

Box-Definition (Feldumrahmung)

Mit diesem Parameter bestimmen Sie, welcher Teil des Rahmens angezeigt werden soll:

„Outlining (Boxing)“ ist die Möglichkeit, bestimmte Felder auf dem Bildschirm eingerahmt anzuzeigen. Diese Form der Anzeige ist eine weitere Möglichkeit, dem Benutzer Länge und Position von Feldern auf dem Schirm deutlich zu machen.

Die Outlining-Funktion steht nur bei bestimmten Terminal-Typen zur Verfügung, gewöhnlich bei denen, die auch die Anzeige von Doppelbyte-Zeichensätzen unterstützen. Wenn der benutzte Terminal kein Outlining unterstützt, wird dieser Parameter zur Ausführungszeit ignoriert.

Mögliche Werte	T	Oberer waagerechter Rand. Siehe Anmerkung 1.
	B	Unterer waagerechter Rand. Siehe Anmerkung 1.
	L	Linker senkrechter Rand. Siehe Anmerkungen 1 und 2.
	R	Rechter senkrechter Rand. Siehe Anmerkungen 1 und 2.
	ON	Entspricht BX=TBLR.
	OFF	Bewirkt, dass keine Kästchen um die betreffenden Felder gezogen werden.
Standard-Einstellung	Keine.	
Gültige Statements	FORMAT	
	DISPLAY INPUT WRITE	Parameter können auf Statement-Ebene und/oder Element- Ebene angegeben werden.
Gültiges Kommando	Keines.	

Anmerkung:

1. Sie können die Werte T, B, L, R in beliebiger Reihenfolge angeben.

2. Wenn Sie die Einstellungen des Session-Parameters `BX=L` oder `BX=R` benutzen, sollten Sie die Bildschirm-Optimierung von Natural mittels der Profilparametereinstellung `DSC=OFF` oder des Natural Terminalkommandos `%R0` ausschalten.

Beispiel:

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

Vgl. Terminalkommando `%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

Fehlerverarbeitung im Batch-Modus

Mit diesem Profil- und Session-Parameter bestimmen Sie, was geschehen soll, wenn bei der Kompilierung oder Ausführung eines Natural-Programms im Batch-Betrieb ein Fehler auftritt.

Er gilt nicht, wenn vom Benutzer geschriebene Fehlerbehandlungsroutinen benutzt werden.

Innerhalb einer Natural-Session kann der Profilparameter CC vom Session-Parameter CC überschrieben werden.

Mögliche Werte	ON	Die Eingabedaten für die Batch-Eingabedateien CMSYNIN und CMOBJIN werden gelöscht, und zwar bis zu einer Zeile, die mit %% an den beiden ersten Stellen beginnt, oder einer „End-of-File“-Bedingung. Folgen weitere Eingabedaten, so liest Natural nach der %%-Zeile weiter.	
	OFF	Natural versucht dann, das nächste Programm (oder Kommando) im Eingabedatenstrom auszuführen.	
Standard-Einstellung	OFF		
Dynamische Spezifikation			
Spezifikation in Session	ja	Gültige Statements:	SET GLOBALS
		Gültiges Kommando:	GLOBALS
Programmierschnittstelle (API)	USR1005N	Siehe SYSEXT - <i>Natural Application Programming Interfaces</i> in der <i>Utilities</i> -Dokumentation.	

Wenn die Natural-Session beendet wird, so wird — falls ein Fehler auftritt — Return Code 4 über Register 15 an das aufrufende Programm übergeben (und zwar unabhängig von der Einstellung des CC-Parameters).

31 CCTAB - Printer Escape Sequence Definition

- CCTAB Parameter Syntax 82
- NTCCTAB Macro Syntax 84
- String Syntax for OPN, CLS, CODE, CS, CSS or CSE 84
- Proportional Fonts 84
- Examples of NTCCTAB Macros 85
- Examples of CCTAB Parameter 85

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

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

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

2nd Variant

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

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

3rd Variant

```
NTCCTAB name, ATR=nnnn, CSS='xxx', CSE='yyy'
```

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

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

```
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

Farbdefinition

Mit diesem Session-Parameter bestimmen Sie die Farbe, in der Felder angezeigt werden. Falls kein Farb-Bildschirm verwendet wird, wird dieser Parameter zur Laufzeit ignoriert.

Verwandter Session-Parameter: [AD](#) - Attribute Definition.

Mögliche Werte	BL	blau
	GR	grün
	NE	neutral
	PI	rosa
	RE	rot
	TU	türkis
	YE	gelb
Standard-Einstellung	NE	
Gültige Statements	FORMAT	
	DISPLAY INPUT PRINT WRITE	Parameter können auf Statement-Ebene und/oder Element-Ebene angegeben werden.
	ASSIGN MOVE REINPUT	Parameter können auf Statement-Ebene angegeben werden.
Gültiges Kommando	Keines	

Beispiel:

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

34 CF - Character for Terminal Commands

Steuerzeichen für Terminalkommandos

Mit diesem Profil- und Session-Parameter bestimmen Sie das Steuerzeichen für Natural-Terminalkommandos, d.h. das Zeichen, mit dem die Terminalkommandos beginnen.

Innerhalb einer Natural-Session kann der Profilparameter CF vom Session-Parameter CF überschrieben werden.

Mögliche Werte	beliebiges Sonderzeichen	Ein Terminalkommando muss mit dem angegebenen Zeichen anfangen. Das mit dem CF-Parameter angegebene Zeichen <ul style="list-style-type: none"> ■ darf nicht dasselbe sein wie das mit dem HI-Parameter (Hilfezeichen) oder IA-Parameter (Input-Zuweisungszeichen) angegebene Zeichen. ■ sollte nicht dasselbe sein wie das mit dem DC-Parameter (Dezimalzeichen) oder ID-Parameter (Input-Delimiterzeichen) angegebene Zeichen. ■ Im Map-Editor ist das Steuerzeichen für Terminalkommandos stets „%“ (um Konflikte mit Delimiterzeichen zu vermeiden, die in Maps benutzt werden), egal welches Zeichen mit dem CF-Parameter definiert ist. 	
	OFF	Es steht kein Steuerzeichen für Terminalkommandos zur Verfügung. Mit SET CONTROL angegebene Terminalkommandos können jedoch nach wie vor ausgeführt werden.	
Standard-Einstellung	%	Ein Terminalkommando muss mit dem Zeichen % anfangen.	
Dynamische Spezifikation	ja		
Spezifikation in Session	ja	Gültige Statements:	SET GLOBALS
		Gültiges Kommando:	GLOBALS

Programmierschnittstelle (API)	USR0350N, USR1005N *	Siehe <i>SYSEXT - Natural Application Programming Interfaces</i> in der <i>Utilities-Dokumentation</i> . * Empfohlen.
---------------------------------------	-------------------------	--

Unter Natural Security: Die Einstellung dieses Parameters kann von der *Session Parameters*-Option des Library-Profiles überschrieben werden.

35

CFICU - Unicode and Code Page Support

▪ CFICU Parameter Syntax	94
▪ NTCFICU Macro Syntax	95
▪ Keyword Subparameters	95
▪ Example of NTCFICU Macro	98
▪ Example of Dynamic Parameter CFICU	98

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 <i>Keyword Subparameters</i> , 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,
```

Or:

```
CFICU=(OFF,COLLATE=value,COLNORM=value,CNVNORM=value,LOCALE=value,CPOPT=value,DATFILE=value,
```

Or:

```
CFICU=[ { (ON) } ]
```

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, *
        BPNLY=value

```

Or:

```
NTCFICU ON
```

Or:

```
NTCFICU OFF
```

Keyword Subparameters

[COLLATE](#) | [COLNORM](#) | [CNVNORM](#) | [LOCALE](#) | [CPOPT](#) | [DATFILE](#) | [BPNLY](#)

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 subparameter `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.



Vorsicht: 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	The default action can be overridden by supplying a character which, when appended to the terminal-command control character (as specified with the CF parameter), forms a valid Natural terminal command. Example: <pre>CF=% CLEAR=R</pre> Natural executes the terminal command %R when the CLEAR key is pressed in NEXT mode.
Default setting	%	By default, when the CLEAR key is pressed, Natural responds as if the user had entered the terminal command %%.
Dynamic specification	yes	
Specification within session	no	

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	

39

CMPO - Compilation Options

- CMPO Parameter Syntax 106
- NTCMPO Macro Syntax 106
- Keyword Subparameters 106
- Example of CMPO Parameter 107
- Example of NTCMPO Macro 107

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 <code>NATPARM</code> , the macro <code>NTCMPO</code> 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-subparameter=va lue,...)
```

NTCMPO Macro Syntax

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

```
NTCMPO keyword-subparameter=va lue,...
```

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` | `TSENABL` | `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

```
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	<i>code-page</i>	<p>The name of the desired code page. Length: 1 - 64 characters.</p> <p>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.</p> <p>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.</p>										
	ON	<p>Set the default code page for IBM and Siemens mainframes as follows:</p> <p>In case of Siemens mainframes, the code page is EDF03IRV.</p> <p>In case of IBM mainframes, it depends on the setting of Natural profile parameter ULANG:</p> <table border="1"> <thead> <tr> <th>ULANG Setting:</th> <th>Code Page Used:</th> </tr> </thead> <tbody> <tr> <td>ULANG=1 (English)</td> <td>IBM01140</td> </tr> <tr> <td>ULANG=2 (German)</td> <td>IBM01141</td> </tr> <tr> <td>ULANG=3 (French)</td> <td>IBM01147</td> </tr> <tr> <td>ULANG=4 (Spanish)</td> <td>IBM01145</td> </tr> </tbody> </table> <p>For other languages, IBM01140 is used as default code page.</p>		ULANG Setting:	Code Page Used:	ULANG=1 (English)	IBM01140	ULANG=2 (German)	IBM01141	ULANG=3 (French)	IBM01147	ULANG=4 (Spanish)
ULANG Setting:	Code Page Used:											
ULANG=1 (English)	IBM01140											
ULANG=2 (German)	IBM01141											
ULANG=3 (French)	IBM01147											
ULANG=4 (Spanish)	IBM01145											

		Anmerkung: 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. Anmerkung: 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

Codepage-Umsetzungsfehler

Mit diesem Profil- und Session-Parameter legen Sie fest, ob eine Fehlermeldung angezeigt wird, wenn ein Umsetzungsfehler bei folgenden Umsetzungen auftritt:

- von Unicode nach Codepage oder
- von Codepage nach Unicode oder
- von einer Codepage in eine andere Codepage

Unabhängig davon enthält der Zieloperand das Ergebnis der Umsetzung, wobei alle Zeichen, die nicht umgesetzt werden können, durch ein für die betreffende Codepage durch die International Components for Unicode (ICU) festgelegtes Ersatzzeichen ersetzt werden. Dieser Parameter wird nicht bei der Umsetzung von Natural-Sourceen berücksichtigt, wenn diese in den Editierbereich geladen oder katalogisiert werden.

Auf Großrechnern wird nicht berücksichtigt, ob ein Unicode-Feld vor einer Eingabe/Ausgabe über eine Terminalemulation in die Codepage umgesetzt wird. In diesem Fall wird das Ersatzzeichen durch das in NATCONFIG definierte Platzhalterzeichen ersetzt.

Siehe auch:

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

Mögliche Werte	ON	Ein Natural-Fehler NAT3413 wird ausgegeben, wenn während der ICU-Umsetzung mindestens ein Code Point nicht korrekt umgesetzt werden konnte. Bei Ausgabe-Statements erfolgt keine Fehlermeldung.	
	OFF	Es wird kein Fehler ausgegeben, wenn ein oder mehrere Code Points während der ICU-Umsetzung nicht korrekt umgesetzt werden konnten.	
Standard-Einstellung	ON		
Dynamische Spezifikation	ja		
Spezifikation in Session	ja	Gültige Statements:	SET GLOBALS
		Gültiges Kommando:	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 `CPPRINT` (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 128
- Example of NTCSTAT Macro 129

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.

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

Kontrollvariable

Mit diesem Session-Parameter wird eine Kontrollvariable referenziert. Eine Kontrollvariable wird mit Format C (siehe *Spezielle Formate* im Leitfaden zur Programmierung) definiert und dient folgenden Zwecken:

- Feldern dynamisch Attribute zuweisen und/oder
- den Status "modifiziert" eines Feldes in Verbindung mit einem INPUT- oder PROCESS PAGE-Statement prüfen; siehe auch *Logische Bedingungen, MODIFIED-Option* im Leitfaden zur Programmierung.

Mögliche Werte	B, C, D, I, N, U, V	Felddarstellungsattribute (siehe Session-Parameter AD).
	P	Feldschutz (siehe Session-Parameter AD).
	BL, GR, NE, PI, RE, TU, YE	Farbe (eine Erläuterung der Farbcodes entnehmen Sie dem Session-Parameter CD).
Standard-Einstellung	Keine	
Gültige Statements	DISPLAY INPUT PRINT PROCESS PAGE WRITE	Parameter können auf Statement- und/oder Element-Ebene angegeben werden.
Gültiges Kommando	Keine	

Beispiel:

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

Mittels einer Kontrollvariablen und der MODIFIED-Option des IF-Statements kann auch überprüft werden, ob während der Ausführung eines INPUT- oder PROCESS PAGE-Statements Feldinhalte verändert worden sind.

```
IF #ATTR MODIFIED ...
```

Eine einzelne Attribut-Kontrollvariable kann auch zu mehreren Eingabefeldern zugewiesen werden, indem man sie einmal auf Statement-Ebene und mehrere Male auf Element-Ebene (Feldebene) angibt. In diesem Fall wird die Modified-Statusanzeige gesetzt, wenn eines der Felder, die die Kontrollvariable referenzieren, geändert worden ist.

Wenn Sie den CV-Parameter auf Statement-Ebene und auf Feldebene angeben und die Kontrollvariable für das betreffende Feld leer ist, wird die Attribut-Kontrollvariable für das Statement auch für das Feld genommen.

Die Attribut-Kontrollvariable kann auf maximal drei Dimensionen erweitert werden, zum Beispiel, CONTR(*), CONTR(*,*), CONTR(*,*,*), je nach Rang des entsprechenden 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 138
- NTDB Macro Syntax 139
- Possible Database Options 140
- Examples of NTDB Macro 141
- Examples of DB Parameter 141

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

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	Natural closes only those databases which had been opened by an explicit open command. An explicit open command will be issued in the following cases: <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. Anmerkung: 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 ETDB, 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

Dezimalstellenzeichen

Mit diesem Profil- und Session-Parameter bestimmen Sie, welches Zeichen als Dezimalkomma bzw. Dezimalpunkt verwendet wird.

Innerhalb einer Natural-Session kann der Profilparameter DC vom Session-Parameter DC überschrieben werden.

Mögliche Werte	beliebiges Zeichen (außer numerischen Zeichen)	<p>Geben Sie den DC-Parameter als DC=' c ' an, wobei das als Dezimalkomma (-punkt) zu benutzende Zeichen darstellt. Das mit dem DC-Parameter angegebene Zeichen</p> <ul style="list-style-type: none"> ■ darf nicht dasselbe sein wie das mit dem Profil/Session-Parameter IA (Input-Zuweisungszeichen), oder auch dem Profil/Session-Parameter ID (Input-Delimiterzeichen) angegebene Zeichen. ■ sollte nicht dasselbe sein wie das mit dem Profil/Session-Parameter CF (Steuerzeichen für Terminalkommandos) oder Profilparameter HI (Hilfezeichen) angegebene Zeichen. 		
Standard-Einstellung	. (Punkt)			
Dynamische Spezifikation	ja			
Spezifikation in Session	ja	Gültige Statements:	SET GLOBALS	Parameter wird zur Laufzeit ausgewertet.
		Gültiges Kommando:	GLOBALS	Parameter kann dynamisch mit dem Systemkommando GLOBALS angegeben werden.
Programmierschnittstelle (API)	USR0350N, USR1005N *	Siehe SYSEXT - <i>Natural Application Programming Interfaces</i> in der <i>Utilities</i> -Dokumentation.		

		* Empfohlen.
--	--	--------------

Unter Natural Security: Die Einstellung dieses Parameters kann von der Session Parameters-Option des Library-Profiles außer Kraft gesetzt werden.

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 in the 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 in the Operations</i> documentation.

64 DF - Date Format

Datumsformat

Mit dem DF-Session-Parameter bestimmen Sie die Länge eines in alphanumerische Darstellung umgesetzten Datums, wenn hierfür keine Editiermaske angegeben ist. Der DF-Parameter wird zur Kompilierungszeit ausgewertet.

Die Reihenfolge der Tages-, Monats- und Jahreskomponenten sowie die Delimiterzeichen werden durch den DTFORM-Profilparameter bestimmt.

Mögliche Werte	S	8-Byte-Darstellung mit 2-stelliger Jahreskomponente und Delimitern (<i>yy-mm-dd</i>). Bei DF=S stehen nur 2 Stellen für die Jahres-Informationen zur Verfügung, d.h. dass wenn der Datumswert das Jahrhundert enthalten würde, gingen diese Informationen bei der Konvertierung verloren.
	I	8-Byte-Darstellung mit 4-stelliger Jahreskomponente und keinen Delimitern (<i>yyyymmdd</i>). * Siehe Anmerkung.
	L	10-Byte-Darstellung mit 4-stelliger Jahreskomponente und Delimitern (<i>yyyy-mm-dd</i>). * Siehe Anmerkung.
Standard-Einstellung	S	
Gültige Statements	FORMAT	
	INPUT DISPLAY WRITE PRINT	Parameter kann dynamisch mit dem FORMAT-Statement angegeben werden.
	MOVE COMPRESS STACK RUN FETCH	Parameter kann auf Element-Ebene angegeben werden.

Gültiges Kommando	Keines
--------------------------	--------

Wenn der Wert eines Datumsfeldes in alphanumerisches Format umgesetzt wird (z.B. in einem MOVE-, DISPLAY-, WRITE- oder INPUT-Statement) und für die Umsetzung keine Editiermaske angegeben ist, wird das durch den Profilparameter `DTFORM` bestimmte Standarddatumsformat als Editiermaske genommen.

Dasselbe gilt bei der Eingabeauswertung einer in einem INPUT-Statement verwendeten Datumsvariablen: Wenn keine Editiermaske angegeben ist, wird die Eingabe entsprechend des durch den Profilparameter `DTFORM` bestimmten Datumsformats ausgewertet.



Anmerkung: Mit `DF=I` bzw. `DF=L` können Sie Ihre Anwendungen nach und nach auf 4-stellige Jahresdarstellung umstellen und dabei weiterhin die durch den `DTFORM`-Parameter gebotene Flexibilität ausnutzen.

Siehe auch den Abschnitt *Datumsformat für alphanumerische Darstellung – der DF-Parameter im Leitfaden zur Programmierung*.

65 DFOUT - Date Format for Output

Datumsformat für Ausgabe

Dieser Natural Profil- und Session-Parameter bestimmt die Form, in dem die Einstellungen der Datumsvariablen von INPUT-, DISPLAY-, PRINT- und WRITE-Statements angezeigt werden.

In einer Natural-Session kann der Profilparameter DFOUT durch den Session-Parameter DFOUT überschrieben werden.

Mögliche Werte	S	Datumsvariablen werden mit einer 2-stelligen Jahreskomponente und Delimitern angezeigt, wie vom Profilparameter DTFORM festgelegt. Beispiel: <i>yy-mm-dd</i>	
	I	Datumsvariablen werden mit einer 4-stelligen Jahreskomponente ohne Delimiter angezeigt. Beispiel: <i>yyyymmdd</i>	
Standard-Einstellung	S		
Dynamische Spezifikation	ja		
Spezifikation in Session	ja	Gültige Statements:	SET GLOBALS
		Gültiges Kommando:	GLOBALS
Programmierschnittstelle (API)	USR1005N	Siehe <i>SYSEXT - Natural Application Programming Interfaces</i> in der <i>Utilities</i> -Dokumentation.	



Anmerkungen:

1. Der *Profilparameter* DFOUT wird zur Laufzeit ausgewertet. Er gilt für Datumsfelder in INPUT-, DISPLAY-, PRINT- und WRITE-Statements, für die weder explizit Editiermasken angegeben sind noch der *Session-Parameter* DF gesetzt ist.
2. Bei beiden DFOUT-Einstellungen wird die Reihenfolge der Tages-, Monats- und Jahreskomponenten in den Datumswerten durch den Profilparameter DTFORM bestimmt.

Siehe auch *Datumsinformationen verarbeiten* im *Leitfaden zur Programmierung*.

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 <i>USR2007N</i> .	
Application Programming Interface	USR2007N	See <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 for *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

Datumsformat für Stack

Dieser Natural Profil- und Session-Parameter bestimmt die Form, in der die Werte von Datumsvariablen mit einem STACK-, RUN- oder FETCH-Statement auf dem Stack abgelegt werden.

In einer Natural-Session kann der Profilparameter DFSTACK durch den Session-Parameter DFSTACK überschrieben werden.

Mögliche Werte	S	Datumsvariablen werden mit 2-stelliger Jahreskomponente und mit durch den Profilparameter DTFORM bestimmten Delimiterzeichen auf dem Stack abgelegt. Beispiel: <i>yy-mm-dd</i>	
	C	Wie DFSTACK=S. Außerdem wird eine Änderung des Jahrhunderts (d.h. wenn das Jahrhundert, das genommen wird, wenn der Wert vom Stack gelesen wird, nicht dasselbe ist wie das Jahrhundert des ursprünglichen Datumswertes, bevor er auf dem Stack abgelegt wurde) zur Laufzeit abgefangen und ein Laufzeitfehler ausgegeben.	
	I	Datumsvariablen werden mit vollständiger 4-stelliger Jahreskomponente und ohne Delimiterzeichen auf dem Stack abgelegt. Beispiel: <i>yyyymmdd</i>	
Standard-Einstellung	S		
Dynamische Spezifikation	ja		
Spezifikation in Session	ja	Gültige Statements:	SET GLOBALS

		Gültiges Kommando:	GLOBALS
Programmierschnittstelle (API)	USR1005N	Siehe SYSEXT - <i>Natural Application Programming Interfaces</i> in der <i>Utilities-Dokumentation</i> .	

Der DFFSTACK-Parameter gilt nicht für STACK-, RUN- und FETCH-Statements, für die der Session-Parameter **DF** gesetzt ist.

Siehe auch *Datumsinformationen verarbeiten im Leitfaden zur Programmierung*.

68

DFTITLE - Output Format of Date in Standard Report Title

Datumsformat in Standard-Seitenüberschrift

Dieser Natural Profil- und Session-Parameter bestimmt die Form des Datums in einer Standard-Seitenüberschrift einer Reportseite (wie sie mit einem `DISPLAY-`, `WRITE-` oder `PRINT-Statement` ausgegeben wird).

In einer Natural-Session kann der Profilparameter `DFTITLE` durch den Session-Parameter `DFTITLE` überschrieben werden.

Mögliche Werte	S	Das Datum wird mit 2-stelliger Jahreskomponente und Delimiterzeichen ausgegeben. Beispiel: <i>yy-mm-dd</i>	
	L	Das Datum wird mit 4-stelliger Jahreskomponente und Delimiterzeichen ausgegeben. Beispiel: <i>yyyy-mm-dd</i>	
	I	Das Datum wird mit 4-stelliger Jahreskomponente ohne Delimiterzeichen ausgegeben. Beispiel: <i>yyyymmdd</i>	
Standard-Einstellung	S		
Dynamische Spezifikation	ja		
Spezifikation in Session	ja	Gültige Statements:	SET GLOBALS
		Gültiges Kommando:	GLOBALS

Programmierschnittstelle (API)	USR1005N	Siehe <i>SYSEXT - Natural Application Programming Interfaces</i> in der <i>Utilities-Dokumentation</i> .
---------------------------------------	----------	--



Anmerkungen:

1. Der `DFTITLE`-Parameter wird zur Laufzeit ausgewertet und bestimmt, ob das Datum mit einer 2-stelligen oder 4-stelligen Jahreskomponente mit oder ohne Delimiter angezeigt wird. Er hat keine Auswirkungen auf benutzerdefinierte Seitenüberschriften (wie sie mit einem `WRITE TITLE`-Statement angegeben werden).
2. Die Reihenfolge der Tages-, Monats- und Jahreskomponenten sowie die Delimiterzeichen werden durch den Profilparameter `DTFORM` bestimmt.

Siehe auch *Datumsinformationen verarbeiten* und *Datumsformat für Standard-Seitenüberschriften – Der DFTITLE-Parameter im Leitfaden zur Programmierung*.

69 DL - Display Length for Output

Ausgabelänge

Mit diesem Session-Parameter bestimmen Sie die Ausgabelänge eines Felds des Formats A oder U. Die Standardausgabelänge ist die Länge des Feldes.

Mögliche Werte	1 bis n	n = Wert des LS-Parameters (Line Size) minus 1
Standard-Einstellung	keine	
Gültige Statements:	FORMAT	
	DISPLAY INPUT PRINT WRITE	Parameter kann auf Statement- und/oder Element-Ebene angegeben werden.
Gültiges Kommando:	keines	

Beispiel:

```
FORMAT DL=20
```

Weitere Informationen und ein Beispiel für die Benutzung des Sessionparameters DL finden Sie in den folgenden Abschnitten des *Leitfadens zur Programmierung*:

- *Parameter zur Beeinflussung der Ausgabe von Feldern*
- *Ausgabelänge — der AL- und der NL-Parameter*
- *Ausgabelänge — der 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 180
- NTDS Macro Syntax 181
- Table of Buffer Sizes 181
- Examples 182

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.



Anmerkung: 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)`.

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 debug buffer	0, 2-2097151	2	yes Anmerkung: 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
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) Anmerkung: A value specified for this buffer does not

Buffer Name	Description	Buffer Size (KB)	Default	Available as subparameter of DS and alternatively as individual profile parameter
				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.



Anmerkung: 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 <code>DSIZE</code> 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

Dump-Generierung

Mit diesem Natural Profil- und Session-Parameter bestimmen Sie, ob ein Memory-Dump generiert wird, falls die Ausführung eines Natural-Programms aufgrund eines Fehlers abgebrochen wird.

In einer Natural-Session kann der Profilparameter `DU` durch den Session-Parameter `DU` überschrieben werden.

Mögliche Werte	ON	Bei einem Programmabbruch wird ein Memory-Dump generiert (TP-Monitor-Dump-Datei oder <code>SYSUDUMP</code> im z/OS Batch-Modus oder TSO). Dann wird die Natural-Session mit der Fehlermeldung NAT9967 oder NAT9974 beendet.
	OFF	Es wird kein Memory-Dump generiert. Im Batch-Betrieb sind von Natural ergriffene Folgemaßnahmen von der Einstellung des Profilparameters <code>CC</code> abhängig. Im Online-Betrieb antwortet Natural mit der Fehlermeldung NAT0950, NAT0953, NAT0954, NAT0955 oder NAT0956. Weitere Informationen zum ABEND erhalten Sie über das Systemkommando <code>DUMP</code> .
	SNAP	Bei einem Programmabbruch im Verlauf der Natural-Session wird sofort ein Dump erzeugt. Die Natural-Session wird anschließend fortgesetzt (wie bei <code>DU=OFF</code>), nachdem der <code>DUMP</code> generiert wurde.
	FORCE	Bei einem Programmabbruch im Verlauf der Natural-Session wird sofort ein Dump erzeugt und die Session sofort abgebrochen. Dies kann in manchen Umgebungen zu Testzwecken sinnvoll sein. Anmerkung: Falls Natural für LE-Unterstützung eingestellt ist, beendet Natural die Natural-Session sofort ohne <code>DUMP</code> und übergibt die Kontrolle an die LE-Fehlerbehandlung. Deshalb wird dringend empfohlen, die LE-Laufzeitoption

		TERMTHDACT(UAIMM) anzugeben, um alle erforderlichen Diagnoseinformationen zu erhalten.	
	ABEND	Funktioniert wie DU=ON, nur dass die Session bei auftretendem ABEND beendet wird – anstatt der Fehlermeldung NAT9974. DU=ABEND steht beim Natural Session-Parameter DU nicht zur Verfügung.	
Standard-Einstellung	OFF		
Dynamische Spezifikation	ja		
Spezifikation in Session	ja	Gültige Statements:	SET GLOBALS
		Gültiges Kommando:	GLOBALS



Anmerkungen:

1. Das Setzen des Profilparameters DU kann aufgrund der I/O-Verarbeitung der Dump-Datei die System-Verarbeitungszeit erheblich beeinträchtigen.
2. Seien Sie vorsichtig, wenn Sie diesen Parameter benutzen, weil alle für den Benutzer gerade aktiven Programme und Subroutinen im Natural Buffer Pool zurückgehalten werden.
3. DU=ON, DU=SNAP bzw. DU=FORCE kann sich aufgrund einer möglichen Puffer-Fragmentierung merklich negativ auf die System-Verarbeitungszeit auswirken.
- 4.
5. Unter UTM wird dieser Parameter ignoriert, und im Falle eines Programmabbruchs wird immer ein Dump erzeugt. Under UTM, this parameter is ignored; under UTM, a dump is always produced in the case of an abnormal program termination.
6. Der Profilparameter DUE kann benutzt werden, um für spezifische Fehler einen Speicherauszug zu erhalten.

Unter Natural Security: Die Einstellung dieses Parameters kann durch die *Session Parameters*-Option des Library-Profiles überschrieben werden.

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-Parametersyntax 195
- Beispiele 197

Dynamische Attribute

Mit diesem Session-Parameter werden Anzeigefeldern dynamisch Feldattribute zugewiesen

Mögliche Werte		Siehe unten.
Standard-Einstellung	Keine	
Gültige Statements	DISPLAY INPUT PRINT WRITE	Parameter kann auf Statement- und/oder Element-Ebene angegeben werden.
Gültiges Kommando	Keine	

Anfang und Ende einer Attribut-Definition werden mit besonderen Identifizierungszeichen (Escape-Zeichen) markiert.

Ein alphanumerisches Feld, das in einem INPUT-, DISPLAY-, WRITE- oder PRINT-Statement verarbeitet wird und Identifizierungszeichen enthält, wird an den Identifizierungszeichen getrennt und in Teilfelder aufgeteilt. Dann wird dem Teilfeld das entsprechende Attribut zugeordnet, und die Identifizierungszeichen werden durch Leerzeichen ersetzt.

Wenn für einen Teil eines Feldes eine mit dem DY-Parameter gemachte Angabe wirksam ist, dann werden die aktuellen Felddarstellungen und Farben auf die im DY-Eintrag neu festgelegten Angaben geändert. Falls das DY-Segment *keine neuen Einstellungen enthält* für die:

■ Felddarstellung

(d.h. keine Werte B, C, D, I, N, U, V), dann bleibt das für das gesamte Feld aktive Attribut wirksam, und zwar unabhängig davon, ob es ursprünglich von einer statischen Einstellung (z.B. AD=I) oder einer von einer Kontrollvariablen (z.B. CV=#C) abgeleitet wurde.

■ Feldfarbe

(d.h. keine Werte BL, GR, NE, PI, RE, TU, YE), dann wird das Farbattribut auf den Wert gesetzt, der dem Feld statisch zugeordnet wurde (CD= . .), und zwar ohne Rücksicht auf ein Farbattribut, dass möglicherweise über eine Kontrollvariable (CV= . .) gesetzt worden war. Falls das Feld keine statische Zuweisung (CD= . .) hat, wird die Farbinformation bei dem durch die DY-Parameter-Neufestlegung betroffenen Feld vollständig entfernt.

Beispiel:

```

DEFINE DATA LOCAL
1 #FLD (A10) INIT <'123<XX>789'>
1 #CV (C) INIT <(AD=V CD=PI)>
END-DEFINE
INPUT (AD=M)
  #FLD (AD=I CD=YE CV=#CV DY=<P>) /* <- the segment "XX" displays yellow, due to
static setting CD=YE,
                                     /* the control variable #CV (including (CD=PI))
is disregarded.
END                                     /* displays with Reverse Video, due to (AD=V) in
control variable #CV.

```

DY-Parametersyntax

```

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

```

Die möglichen Werte werden im Folgenden erläutert.

escape-character1

Ein Zeichen, das den Anfang der Attribut-Definition markiert. Als Zeichen können Sie ein beliebiges Sonderzeichen (c) oder eine Hexadezimalzahl mit einem vorangestellten Apostroph ('xx) verwenden.

color-attribute

Das zuzuordnende Farbattribut. Erklärung der Farbcodes siehe Session-Parameter CD.

BL	blau
GR	grün
NE	neutral
PI	rosa
RE	rot
TU	türkis
YE	gelb

i/o-characteristics

Wert	Bedeutung
P	Unterfeld soll schreibgeschützt sein.

Es kann ein P angegeben werden, damit das Unterfeld schreibgeschützt ist. Siehe auch Session-Parameter [AD](#) (Attribut-Definition).

field-representation-attribute

Zusätzliche Attribute können zugewiesen werden. Siehe auch Session-Parameter [AD](#) (Attribut-Definition).

Wert	Bedeutung
B	blinkend (*)
C	kursiv (*)
D	Standard normal
I	intensiviert
N	Nicht-Anzeige
U	unterstrichen
V	invers (*)

* Die mit einem Stern (*) markierten Feldanzeige-Attribute sind an entsprechende Hardware-Voraussetzungen gebunden und werden zur Laufzeit ignoriert, falls diese Voraussetzungen nicht gegeben sind.

escape-character2

Ein Identifizierungszeichen, das das Ende der Attribut-Definition markiert. Als Zeichen können Sie ein beliebiges Sonderzeichen (c) oder eine Hexadezimalzahl mit einem vorangestellten Apostroph ('xx) verwenden.

Sie können bis zu acht Attributsequenzen (Anfangsidentifizierungszeichen und Attribute) vor dem Zeichen, das das Ende der Attribut-Definitionen bestimmt, angeben.

Beispiele

Beispiel 1:

```
DY=<U>
```

Die Textkette:

```
THIS <i s> UNDERLINED
```

wird ausgegeben als:

```
THIS i s UNDERLINED
```

Beispiel 2:

```
DY=<BL|RE/GR>
```

Weist zu:

Blau zu <

Rot zu |

Grün zu /

> schaltet wieder zur ursprünglichen Farbe des Feldes zurück.

Beispiel 3:

```
DY=<P>;
```

Die Textkette:

```
Do not overwrite <this>
```

wird ausgegeben als:

Do not overwrite this

(wobei this geschützt ist)

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 NATPARAM.

Possible settings	ON	All profile parameters can be specified dynamically.
	OFF	No profile parameters can be specified dynamically.
	DYNPARAM=(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.
	DYNPARAM=(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 DYNPARAM 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 <i>SYSEXT - Natural Application Programming Interfaces</i> in the <i>Utilities</i> documentation.

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

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

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 <code>NATPARM</code> , the macro <code>NTEDBP</code> must be used instead.
Specification within session	yes	Use the <code>SYSEDT</code> 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 `SYSEDT` Editor Buffer Pool Services utility documentation.

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



Anmerkungen:

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.



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

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

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

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

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

83 EJ - Page Eject

Seitenvorschub

Mit diesem Natural Profil- und Session-Parameter bestimmen Sie, ob am Ende einer logischen Seite, beim Wechsel zwischen Programmeingabe und -ausgabe und nach der Meldung „Normal End“ ein Seitenvorschub erfolgen soll.

In einer Natural-Session kann der Profilparameter EJ durch den Session-Parameter EJ überschrieben werden. Die EJ-Einstellung kann wiederum durch ein EJECT-Statement überschrieben werden.

Mögliche Werte	ON	Ein Seitenvorschub wird ausgeführt.		
	OFF	Ein Seitenvorschub wird nicht ausgeführt. Bei Testläufen, bei denen Seitenumbrüche keine Rolle spielen, empfiehlt es sich, EJ=OFF zu setzen, um Papier zu sparen.		
Standard-Einstellung	ON			
Dynamische Spezifikation	ja			
Spezifikation in Session	ja	Gültige Statements:	SET GLOBALS	Parameter wird zur Laufzeit ausgewertet.
		Gültiges Kommando:	GLOBALS	Parameter kann mit dem GLOBALS-Systemkommando dynamisch angegeben werden.
Programmierschnittstelle (API)	USR1005N	Siehe SYSEXT - <i>Natural Application Programming Interfaces</i> in der <i>Utilities</i> -Dokumentation.		

Dieser Parameter gilt nur für den ersten auszugebenden Report (Report 0). Für weitere Reports ist das EJECT-Statement mit Report-Spezifikation (*rep*) zu verwenden.



Vorsicht: Der Profilparameter EJ hat eine etwas andere Bedeutung, wenn er für eine Natural-Session unter CICS im Batch-Betrieb angegeben wird (zum Beispiel: TTYPE=ASYL oder

TTYPE=BTCH); siehe *Asynchronous Natural Processing under CICS* in der *TP Monitor Interfaces-Dokumentation*.

Unter Natural Security: Die Einstellung dieses Parameters kann durch die Session Parameters-Option des Library-Profiles überschrieben werden.

84

EM - Edit Mask

▪ Syntax	218
▪ Leerzeichen in Editiermasken	219
▪ Standard-Editiermasken	219
▪ Editiermasken für numerische Felder	220
▪ Editiermasken für alphanumerische Felder	223
▪ Editiermasken für binäre Felder – Format B	225
▪ Hexadezimale Editiermasken	225
▪ Editiermasken für Datums- und Zeitfelder (Formate D und T)	227
▪ Editiermasken für logische Felder (Format L)	232

Editiermaske

Mit diesem Session-Parameter können Sie für ein Eingabe- und/oder Ausgabefeld, das in einem DEFINE DATA-, DISPLAY-, INPUT-, MOVE EDITED-, PRINT-, PROCESS PAGE- oder WRITE-Statement verwendet wird, eine Editiermaske definieren.

Der Session-Parameter EM kann auch bei Feldern mit Format U (Unicode) verwendet werden, siehe auch *Unicode and Code Page Support in the Natural Programming Language, Session Parameters*, EMU, ICU, LCU, TCU im Vergleich zu EM, IC, LC, TC.

Mögliche Werte		Siehe unten.
Standard-Einstellung	Keine	
Gültige Statements	FORMAT	Parameter kann dynamisch mit dem FORMAT-Statement angegeben werden.
	DEFINE DATA DISPLAY INPUT PRINT WRITE	Parameter kann auf Statement- und/oder Element-Ebene angegeben werden.
	MOVE EDITED	Parameter kann auf Element-Ebene angegeben werden.
Gültiges Kommando	Keines	

Die folgenden Themen werden nachfolgend erörtert:

Siehe auch *Editiermasken – der EM-Parameter im Leitfaden zur Programmierung*.

Syntax

Bei Eingabefeldern muss ein Wert genau entsprechend der Editiermaske eingegeben werden. Um die Editiermaske für die Eingabe sichtbar zu machen, sollte ein Eingabefeld als modifizierbar (AD=M) definiert werden.

Für ein Datenbankfeld kann im DDM bereits eine Standard-Editiermaske definiert sein. Wenn Sie mit dem EM-Parameter eine Editiermaske für ein Datenbankfeld angeben, so gilt diese anstelle einer möglicherweise im DDM für das Feld definierten Standard-Editiermaske.

Geben Sie für ein Feld EM=OFF an, so wird für das Feld keine Editiermaske verwendet, auch keine möglicherweise im DDM definierte.

Auf Statement-Ebene eines DISPLAY-, FORMAT-, INPUT- oder WRITE-Statements kann keine Editiermaske definiert, sondern allenfalls EM=OFF gesetzt werden.

Wenn eine Editiermaske definiert ist, so überschreibt diese etwaige Einstellungen der Session-Parameter [AL](#), [NL](#) and [SG](#).

Diese Schreibweise gilt nur für die Zeichen 9, H, X und Z, mit denen bei numerischen (9, Z), hexadezimalen (H) und alphanumerischen (X) Editiermasken die signifikanten Stellen dargestellt werden. Der Unterschied zwischen 9 und Z ist im Abschnitt [Editiermasken für numerische Felder](#) weiter unten beschrieben.

Beispiele:

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

Lange Editiermasken können in Form einer Kurzschreibweise definiert werden. Die folgenden Beispiele zeigen die für numerische, hexadezimale und alphanumerische Editiermasken verwendbare Kurzschreibweise.

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

Leerzeichen in Editiermasken

Leerzeichen innerhalb einer Editiermaske lassen sich durch das Zeichen auf Ihrer Tastatur darstellen, das in Hexadezimalcode H'20' (ASCII) bzw. H'5F' (EBCDIC) entspricht, d.h. das Zeichen ^ (oder ¬).

Standard-Editiermasken

Wenn Sie für ein Feld keine Editiermaske angeben, erhält das Feld entsprechend seinem Format eine Standard-Editiermaske:

Feldformat	Standard-Editiermaske
A	X
B	H
N, P, I	Z9
F	wissenschaftliche Darstellung
D	abhängig vom Standard-Datumsformat (wie mit dem Profilparameter DTFORM gesetzt)

Feldformat	Standard-Editiermaske
T	HH:II:SS
L	Leerzeichen / X

Editiermasken für numerische Felder

Eine für Felder mit Format N, P, I oder F definierte Editiermaske muss mindestens eine 9 oder ein Z enthalten.

Enthält sie mehr „9er“ und „Zs“ als der Feldwert lang ist, wird die Anzahl der Ausgabestellen in der Editiermaske der Anzahl der für den Feldwert definierten Stellen angepasst.

Hat die Editiermaske weniger signifikante Stellen als der Feldwert, wird der Feldwert um die entsprechende Anzahl der Stellen vor bzw. nach dem Komma verkürzt ausgegeben.

Die folgenden Themen werden nachfolgend erörtert:

- Zeichen zur Definition numerischer Editiermasken
- Vorzeichen
- Führende Literale
- Literale Einschubzeichen und nachgestellte Zeichen
- Nachfolgende Vorzeichen
- Beispiele für numerische Editiermasken

Zeichen zur Definition numerischer Editiermasken

Zeichen	Funktion
9	Auszugebende Stelle (eine Stelle des Feldwertes).
.	<p>(period) Ein Punkt, als erstes Zeichen verwendet, stellt ein Komma (Dezimalpunkt) dar und ist signifikant. Nachfolgende Punkte werden als Literale behandelt.</p> <p>Anmerkung: An dieser Stelle stellt der Punkt das zurzeit als Dezimalpunktzeichen festgelegte Zeichen dar. Falls mit dem Session- oder Profilparameter DC ein anderes Zeichen gewählt wird (zum Beispiel ein Komma), dann ist anstelle des Punktes dieses Zeichen zu verwenden.</p>
Z	Nullunterdrückung bei vorangestellten Nullen. Dies gilt standardmäßig für numerische Felder. Zur gleitenden Nullunterdrückung kann das Z mehrmals angegeben werden. Rechts vom Komma (Dezimalpunkt) darf kein Z stehen. Ein Nullwert kann unter Einbeziehung aller Zs in der Editiermaske als lauter Leerzeichen ausgegeben werden (siehe auch Session-Parameter ZP).

Vor den Neunern oder Zs können eins oder mehrere andere Zeichen stehen.

Vorzeichen

Wenn das erste Zeichen vor den Neunern oder Zs ein +, -, S oder N ist, kann ein Vorzeichen angezeigt werden:

Zeichen	Funktion
+	Ein gleitendes Vorzeichen, das vor/nach der Zahl ausgegeben werden soll. Das Zeichen wird je nach Wert der Zahl als + oder - generiert.
-	Ein gleitendes Minus-Vorzeichen, das vor/nach der Zahl ausgegeben werden soll, wenn die Zahl negativ ist.
S	Ein Vorzeichen, das vor dem Feld ausgegeben werden soll. Das Vorzeichen wird je nach Wert der Zahl als + oder - generiert.
N	Ein Minus-Vorzeichen, das vor dem Feld ausgegeben werden soll, wenn der Feldwert negativ ist.

Führende Literale

Eine beliebige Anzahl von führenden Literalen kann vor der ersten anzeigbaren Stelle erscheinen (wie durch Z oder 9 angezeigt). Diese müssen auf ein Vorzeichen folgen. Wenn kein Vorzeichen vorhanden ist, und das erste führende Literal ist +, -, S oder N, muss es in Apostrophen stehen. Wenn ein führendes Literal H, X, Z oder 9 ist, muss es in Apostrophen stehen.

Das zuerst angegebene führende Literal erscheint nur in der Ausgabe, wenn der Wert führende Nullen enthält und die Editermaske mit Z definiert ist (führende Nullwertunterdrückung). Dieses Zeichen wird dann als Füllzeichen benutzt, das anstatt eines Leerzeichens für führende Nullen angezeigt wird. Nachfolgende führende Literale werden so angezeigt, wie sie eingegeben werden.

Literale Einschubzeichen und nachgestellte Zeichen

Es können auch literale Einschubzeichen und nachgestellte Zeichen benutzt werden. Das Symbol (^) kann zur Darstellung eines vorangestellten, eingefügten oder nachgestellten Leerzeichens verwendet werden. Durch Setzen von signifikanten Zeichen (9, H, Z, X) in Apostrophen ist es möglich, vorangestellte, eingefügte und nachgestellte Zeichenketten zu haben.

Nicht signifikante Editiermasken-Zeichen müssen nicht in Apostrophen stehen. Innerhalb derselben Editiermasken-Notation ist es möglich, Gruppen von vorangestellten Zeichenketten, Einschubzeichen und/oder nachfolgenden Zeichen zu haben, von denen einige in Apostrophen stehen und andere nicht.

Nachfolgende Vorzeichen

Ein nachfolgendes Vorzeichen wird durch ein + oder - als letztes Zeichen der Editiermaske angegeben. Ein + bewirkt, dass das Vorzeichen je nach Feldwert entweder als + oder - ausgegeben wird; ein - bewirkt, dass bei einem positiven Feldwert ein Leerzeichen und bei einem negativen Feldwert ein - ausgegeben wird. Ist für eine Editiermaske ein vorangestelltes und ein nachgestelltes Vorzeichen definiert, werden beide ausgegeben.

Beispiele für numerische Editiermasken

Die folgende Tabelle zeigt in der oberen Zeile die Werte numerischer Felder (Format N), wie sie ohne Editiermaske ausgegeben würden, und darunter die unter Verwendung der verschiedenen Editiermasken ausgegebene Form:

Wert	0000.03 (N4.2)	-0054 (N4)	+0087 (N4)	0962 (N4)	1830 (N4)
Editiermaske					
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

Durch Kombination von Editiermasken mit den Parametern IC und TC ist es bei einem DISPLAY-Statement möglich, negative Zahlen in verschiedenen Formen auszugeben.

Editiermasken für alphanumerische Felder

Für mit Format "A" definierte Felder kann eine alphanumerische Editiermaske definiert werden; sie muss mindestens ein "X" enthalten; jedes X steht für ein auszugebendes Zeichen. Ein H als erstes Zeichen kennzeichnet eine **hexadezimale Editiermaske**. Ein Leerzeichen wird durch ein Circumflex (^) dargestellt.

Alle anderen Zeichen — außer Klammern — können als vorangestellte, eingeschobene oder nachgestellte Zeichen verwendet werden, wobei diese Zeichen wahlweise durch Apostrophe eingegrenzt werden können oder nicht. Sollen die Zeichen \, eine abschließende runde Klammer () oder ein Anführungszeichen (") als nichtsignifikante Einschubzeichen verwendet werden, müssen sie in Apostrophen angegeben werden.

Werden dem ersten signifikanten X Zeichen vorangestellt, wird das erste dieser Zeichen nicht ausgegeben, sondern als Füllzeichen benutzt, und es tritt an die Stelle aller führenden Leerzeichen im alphanumerischen Ausgabefeld.

Beispiel:

```
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
```

Ausgabe:

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

Zeichen, die unmittelbar auf das letzte signifikante X folgen, werden ausgegeben.

Ist die Editiermaske kürzer als das Feld, wird die Anzahl der ausgegebenen Stellen auf die Länge der Editiermaske gekürzt.

Ist die Editiermaske länger als das Feld, wird die Anzahl der ausgegebenen Stellen bei der ersten überstehenden Stelle abgeschnitten.

Beispiel:

```

DEFINE DATA LOCAL
  1 #TEXT (A4) INIT <'BLUE'>
END-DEFINE
WRITE #TEXT (EM=X-X-X)      displays as 'B-L-U'      .. only three bytes of field
WRITE #TEXT (EM=X-X-X-X)   displayed
                           displays as 'B-L-U-E-' .. mask definition was
                           truncated to (EM=X-X-X-X-)

```

Beispiel für alphanumerische Editiermasken

Das folgende Programm definiert Editiermasken für ein Feld mit Format/Länge A4, das den Wert BLUE enthält:

```

** 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

```

Ausgabe des Programms 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

```


Editiermasken für binäre Felder – Format B

Editiermasken für binäre Felder können mittels der Notation X oder H definiert werden. Für binäre Felder wird die Notation X unterstützt, als ob H anstelle von X angegeben worden wäre.

Hexadezimale Editiermasken

Wird als erstes Zeichen einer Editiermaske ein H angegeben, so wird der Wert eines alphanumerischen oder numerischen Feldes in hexadezimaler Form ausgegeben. Jedes H steht für zwei Hexadezimalstellen, die jeweils einem numerischen/alphanumerischen Byte im Source-Feld entsprechen.

Alle anderen Zeichen können als Einschubzeichen oder nachgestellte Zeichen verwendet werden. Ist die Editiermaske kürzer als das Feld, wird der Feldwert entsprechend verkürzt ausgegeben. Ist das Feld kürzer als die Editiermaske, wird die Editiermaske der Feldlänge entsprechend verkürzt ausgegeben.

Einschubzeichen oder nachgestellte Zeichen können wahlweise durch Apostrophe eingegrenzt werden.

Alle mit einer hexadezimalen Editiermaske angezeigten Felder werden als alphanumerische Felder behandelt. Ist die Editiermaske kürzer als das Feld, werden daher alle numerischen oder alphanumerischen Stellen von links nach rechts ohne Berücksichtigung von Dezimalstellen ausgegeben.

Wenn eine hexadezimale Editiermaske als eine Eingabeeditiermaske benutzt wird, werden alle Zeichen 0-9, a-f, , A-F, Leerzeichen und die hexadezimale Null als eine hexadezimale Ziffer akzeptiert.



Anmerkung: Leerzeichen und die hexadezimale Null werden als 0 und ein Kleinbuchstabe (a-f) als Großbuchstabe angesehen.

Editiermasken-Beispiele für hexadezimale Felder

Die folgenden Tabellen zeigen hexadezimale Editiermasken mit Ergebnissen aus den ursprünglichen Feldern und über jeder Spalte angezeigten Werten. Alle numerischen Werte (-10, +10, 01), für die Editiermasken gelten, stammen aus im Format N2 definierten Feldern. Der alphanumerische Wert AB stammt aus einem mit Format/Länge A2 definierten Feld.

ASCII:

Wert =>	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:

Wert =>	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

Beispielprogramm mit hexadezimalen Editiermasken:

```

** 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

```

Ausgabe des Programms 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

```

Ausgabe des Programms 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

```

Editiermasken für Datums- und Zeitfelder (Formate D und T)

Zur Definition von Editiermasken für Felder, die mit dem Format D (Datumsfeld) oder T (= Time; Zeitfeld) definiert sind, können die in den folgenden Abschnitten beschriebenen Zeichen verwendet werden:

- Datums- und Zeitfelder (Formate D und T)
- Syntaktische Einschränkungen für Datumszeichen
- Hinweise für Eingabe-Editiermaske
- Hinweise für Wochenanzeige (WW oder ZW) in Ausgabe-Editiermaske
- Nur für Zeitfelder (Format T):

- Beispiele für Datums- und Zeit-Editiermasken

Datums- und Zeitfelder (Formate D und T)

Zeichen	Bedeutung
DD	Tag (Day).
ZD	Tag mit Nullwertunterdrückung.
MM	Monat.
ZM	Monat mit Nullwertunterdrückung.
YYYY	Jahr (Year), vierstellig (siehe Abschnitt Hinweise für Eingabe-Editiermaske).
YY	Jahr, zweistellig (siehe Abschnitt Hinweise für Eingabe-Editiermaske).
Y	Jahr, einstellig. Darf nicht für Eingabefelder verwendet werden.
WW	Woche (siehe die Abschnitte Hinweise für Eingabe-Editiermaske und Hinweise für Wochenanzeige in Ausgabe-Editiermaske).
ZW	Woche mit Nullwertunterdrückung (siehe die Abschnitte Hinweise für Eingabe-Editiermaske und Hinweise für Wochenanzeige in Ausgabe-Editiermaske).
JJJ	Julianischer Tag.
ZZJ	Julianischer Tag mit Nullwertunterdrückung.
NN . . . oder N(<i>n</i>)	Name des Wochentages (sprachabhängig). Die Maximallänge wird durch die Anzahl der Ns bzw. durch <i>n</i> bestimmt. Ist der Name länger als die Maximallänge, wird er abgeschnitten; ist er kürzer, wird seine tatsächliche Länge genommen.
0	<p>Nummer des Wochentags.</p> <p>Ob Montag oder Sonntag als erster Wochentag genommen wird, hängt vom Profilparameter DTFORM ab).</p> <p>Ist DTFORM=U, dann ist Sonntag = 1, Montag = 2 usw.).</p> <p>Ist DTFORM=<i>sonstige</i>, dann ist Montag = 1, Dienstag = 2 usw.).</p>
LL . . . oder L(<i>n</i>)	<p>Name des Monats (sprachabhängig).</p> <p>Die Maximallänge wird durch die Anzahl der Ls bzw. durch <i>n</i> bestimmt.</p> <p>Ist der Name länger als die Maximallänge, wird er abgeschnitten; ist er kürzer, wird seine tatsächliche Länge genommen.</p>
R	<p>Jahr in römischen Ziffern (maximal 13 Stellen).</p> <p>Darf für Eingabefelder nicht benutzt werden.</p> <p>Die obere Grenze für anzeigbare Jahrwerte ist 2887.</p>

Syntaktische Einschränkungen für Datumszeichen

Für *Eingabe-/Ausgabe*-Editiermasken dürfen Sie die folgenden Zeichen *nicht* verwenden:

Text			Zeichen		
Monat	mit	Monatsnamen	MM oder ZM	mit	LL oder L(<i>n</i>)
Tagesname	mit	Wochentag	NN oder N(<i>n</i>)	mit	0

Für *Eingabe*-Editiermasken dürfen Sie die folgenden Zeichen *nicht* verwenden:

Text			Zeichen		
1-stelliges Jahr	und auch nicht	ein Jahr in römischen Ziffern	Y	und auch nicht	R
Day	ohne	Monat oder Monatsnamen	DD oder ZD	ohne	MM oder ZM oder LL oder L(<i>n</i>)
Woche	ohne	Jahr	WW oder ZW	ohne	YYYY oder YY
Monat	ohne	Jahr	MM oder ZM	ohne	YYYY oder YY
Julian. Tag	ohne	Jahr	JJJ oder ZZJ	ohne	YYYY oder YY
Tagesname	ohne	Woche	NN oder N(<i>n</i>)	ohne	WW oder ZW
Wochentag	ohne	Woche	0	ohne	WW oder ZW
Julian. Tag	mit	Monat	JJJ oder ZZJ	mit	MM oder ZM
Julian. Tag	mit	Woche	JJJ oder ZZJ	mit	WW oder ZW
Monat	mit	Woche	MM oder ZM	mit	WW oder ZW

Hinweise für Eingabe-Editiermaske

Die gültigen Jahreswerte (YYYY) sind 1582 - 2699.

Wird der Profilparameter `MAXYEAR` auf "9999" gesetzt, ist der Bereich der gültigen Jahreswerte 1582 - 9999.

Wird in einer Eingabe-Editiermaske nur das Jahr (YY oder YYYY) angegeben, aber nicht Monat und Tag, werden die Werte für Monat und Tag jeweils auf 01 gesetzt. Werden in einer Eingabe-Editiermaske nur Jahr (YY oder YYYY) und Monat (MM) angegeben, aber kein Tag, wird der Wert für Tag auf 01 gesetzt.

Bei einer 2-stelligen Jahresangabe (YY) ist das zum Füllen der Jahresdarstellung benutzte Jahrhundert standardmäßig das aktuelle Jahrhundert. Dies gilt aber nicht, wenn ein Sliding Window oder Fixed Window definiert ist. Weitere Einzelheiten entnehmen Sie dem Profilparameter `YSLW`.

Wird eine Woche (WW oder ZW) aber kein Wochentag (0) oder Tagesname (NN. . .) angegeben, geht man vom ersten Wochentag aus.

Hinweise für Wochenanzeige (WW oder ZW) in Ausgabe-Editiermaske

Wenn DTFORM=U (USA-Format) gesetzt ist, beginnt die Woche am Sonntag, wohingegen bei allen anderen DTFORM-Einstellungen der erste Wochentag der Montag ist. Ob eine Woche die 52./53. Woche des alten oder die 1. Woche des neuen Jahres ist, hängt davon ab, welches Jahr mehr Wochentage enthält. Mit anderen Worten, wenn der Donnerstag (Mittwoch bei DTFORM=U) dieser Woche im alten Jahr liegt, gehört die Woche zum alten Jahr; liegt er im neuen Jahr, gehört die Woche zum neuen Jahr.

Wenn sich eine Darstellung der Woche (WW oder ZW) und des Jahres (YYYY oder YY oder Y) auf derselben Editiermaske befinden, entspricht die Anzeige des Jahres stets der Woche, ungeachtet des Jahres in dem zugrundeliegenden Datumsfeld.

Beispiel:

```
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)
```

Obwohl das zugrundeliegende Datum der 31. Dezember 2003 ist, wenn die Woche WW in der Editiermaske enthalten ist, wird es wie folgt angezeigt:

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

Nur für Zeitfelder (Format T):

Zeichen	Bedeutung
T	Zehntelsekunden (Tenths of a second).
SS	Sekunden.
ZS	Sekunden mit Nullwertunterdrückung.
II	Minuten.
ZI	Minuten mit Nullwertunterdrückung.
HH	Stunden.
ZH	Stunden mit Nullwertunterdrückung.
AP	AM/PM-Element (englische Zeitangabe: AM = vormittags, PM = nachmittags).

Beispiele für Datums- und Zeit-Editiermasken

```

** 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

```

Ausgabe des Programms 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

```

Editiermasken für logische Felder (Format L)

Editiermasken für Felder, die das Format L haben (logische Felder), können wie folgt definiert werden:

```
(EM=[false-string]true-string)
```

wobei *false-string* für die für „falsch“ auszugebende Zeichenkette und *true-string* für die für „wahr“ auszugebende Zeichenkette steht.

Beispiel für Editiermasken für logisches Feld

```
** 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
```

Ausgabe des Programms 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 <code>BACKOUT TRANSACTION</code> statement at session end.
	OFF	Natural will not issue an implicit <code>BACKOUT TRANSACTION</code> statement at session end.
	ETDB	Natural will issue an implicit <code>BACKOUT TRANSACTION</code> statement at session end only for the database specified with the profile parameter <code>ETDB</code> .
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

Leerzeilenunterdrückung

Mit diesem Session-Parameter können Sie die Ausgabe der von einem DISPLAY- oder WRITE-Statement erzeugten Leerzeilen unterdrücken.

Mögliche Werte	ON	Eine Zeile aus einem DISPLAY- oder WRITE-Statement, die alle Leerwerte enthält, wird nicht ausgegeben. Diese Einstellung ist besonders nützlich, wenn Arrays angezeigt werden (z.B. multiple Felder oder Felder in Periodengruppen), um nicht überflüssig viele Leerwerte auszugeben.
	OFF	Leerzeilenunterdrückung ist ausgeschaltet.
Standard-Einstellung	OFF	
Spezifikation in Session	ja	
Gültige Statements	DISPLAY FORMAT WRITE	
Gültiges Kommando	Keines	

Um die Leerwertunterdrückung auch für numerische Werte zu erhalten, muss für die betreffenden Felder neben `ES=ON` auch der Parameter `ZP=OFF` gesetzt werden, was bewirkt, dass Nullwerte in Leerwerte umgesetzt und dann ebenfalls nicht ausgegeben werden. Siehe auch Session-Parameter [IS](#) und [ZP](#).

Beispiel:

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

Siehe auch *Parameter zur Beeinflussung der Ausgabe von Feldern im Leitfaden zur Programmierung*.

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. Anmerkung: 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	



Anmerkung: 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.



Anmerkung: 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 END TRANSACTION statement at the end of a Natural program.
	OFF	Natural will not issue any implicit END TRANSACTION statement at the end of a Natural program.
Default setting	OFF	
Dynamic specification	yes	
Specification within session	no	

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 "ON" to enforce a close command at session end. Otherwise, the user is not logged off from Adabas and the Adabas user queue element is not deleted. This may cause an overflow situation in the Adabas user queue.
Default setting	*INIT-USER	
Dynamic specification	yes	
Specification within session	no	

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

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

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

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.



Vorsicht: 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 [NTWORK](#).

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

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 INPUT 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

Füllzeichen für DISPLAY-Statement

Mit diesem Session-Parameter bestimmen Sie das Füllzeichen, mit dem bei Spaltenüberschriften, die über ein DISPLAY-Statement erzeugt werden, der Platz rechts und links der Überschrift aufgefüllt wird.

Im Gegensatz zum Parameter `GC`, der für Überschriften über einen Reihe von Spalten hinweg gilt, gilt der Parameter `FC` für einzelne Spalten.

Mögliche Werte	ein beliebiges Zeichen	Füllzeichen für einzelne Überschriften. FC gilt nur, wenn die Spaltenbreite von der Feldlänge und nicht von der Länge der Überschrift bestimmt wird. Siehe auch Session-Parameter HW); sonst wird der FC-Parameter ignoriert.	
Standard-Einstellung	leer		
Spezifikation in Session	ja	Gültige Statements:	DISPLAY FORMAT
		Gültiges Kommando:	Keines

Beispiel:

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


105

FCDP - Filler Character for Dynamically Protected Input

Fields

Füllzeichen für dynamisch geschützte Felder

Mit diesem Natural Profil- und Session-Parameter können Sie die Anzeige von Füllzeichen für Eingabefelder unterdrücken, die dynamisch schreibgeschützt wurden (d.h. denen das Attribut `AD=P` mittels einer Kontrollvariablen zugewiesen wurde).

Je nach dem Wert dieses Parameters werden dynamisch geschützte Eingabefelder entweder mit Leerzeichen oder mit den definierten Füllzeichen angezeigt.

In einer Natural-Session kann der Profilparameter `FCDP` durch den Session-Parameter `FCDP` überschrieben werden.

Mögliche Werte	ON	Dynamisch geschützte Eingabefelder werden mit Füllzeichen gefüllt angezeigt. Dies kann bei Benutzern den Eindruck erwecken, sie könnten in diese Felder etwas eingeben.	
	OFF	Dynamisch geschützte Eingabefelder werden mit Leerzeichen gefüllt angezeigt.	
Standard-Einstellung	ON		
Dynamische Spezifikation	ja		
Spezifikation in Session	ja	Gültige Statements:	SET GLOBALS
		Gültiges Kommando:	GLOBALS
Programmierschnittstelle (API)	USR1005N	Siehe <i>SYSEXT - Natural Application Programming Interfaces</i> in der <i>Utilities</i> -Dokumentation.	

Beispiel:

```
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
```

Die Ausführung des vorstehenden Programms führt zu folgender Anzeige:

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.



Anmerkung: 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

Fließpunkt-Mantissenlänge

Mit diesem Session-Parameter bestimmen Sie die Mantissen-Länge einer Gleitkomma-Variablen während der Ein- oder Ausgabe.

Mögliche Werte	1 - 16	Mantissen-Länge. Die Gesamtlänge für Vorzeichen, Exponent und Dezimalkomma ist FL + 6.	
Standard-Einstellung	keine		
Spezifikation in Session	ja	Gültige Statements:	DISPLAY FORMAT INPUT PRINT WRITE
		Gültiges Kommando:	Keines

Beispiel:

```
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

		<i>ROSY</i> , 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)
```

 **Anmerkungen:**

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.

110

FREEGDA - Release GDA in Utility Mode

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

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

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

Variables

Format-Spezifikation für Benutzervariablen

Dieser Natural Profil- und Session-Parameter gilt nur für den Reporting Mode; er hat keine Auswirkungen im Structured Mode.

Mit diesem Session-Parameter bestimmen Sie, ob für die Definition von Benutzervariablen im Reporting Mode Standardformat und -länge gelten sollen (siehe auch *Format und Länge von Benutzervariablen im Leitfaden zur Programmierung*).

In einer Natural-Session kann der Profilparameter FS durch den Session-Parameter FS überschrieben werden.

Mögliche Werte	ON	Einer neuen Benutzervariablen wird im Reporting Mode von Natural kein/e Standardformat/-länge zugeordnet; Sie müssen Format und Länge explizit definieren.	
	OFF	Eine Benutzervariable in einem Natural-Programm, für die Format und Länge nicht explizit definiert sind, erhält Standardformat/-länge N7.	
Standard-Einstellung	OFF		
Dynamische Spezifikation	ja		
Spezifikation in Session	ja	Applicable Statements:	SET GLOBALS
		Applicable Command:	GLOBALS
Programmierschnittstelle (API)	USR1005N	Siehe SYSEXT - <i>Natural Application Programming Interfaces</i> in der <i>Utilities</i> -Dokumentation.	

Unter Natural Security: Die Einstellung dieses Parameters kann durch die Session-Parameter-Option des Library-Profiles überschrieben werden.

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.
	RO	Read-only option. RO 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)



Anmerkung: 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

FSPPOOL - 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 FSPPOOL 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: FSPPOOL=(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.
	RO		Read-only option. RO indicates that the Natural user-program system file is „read-only“ and is only specified if modifications on the file are to be disabled.
Default setting	none		
Dynamic specification	yes		If you specify the FUSER parameter dynamically in conjunction with any of the parameters DBID , FNR , SYSPSW , SYSCIP or ROSY , you must specify the FUSER parameter <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)
```



Anmerkung: 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

Füllzeichen für Gruppenüberschriften

Mit diesem Session-Parameter definieren Sie das Füllzeichen, mit dem bei Überschriften, die über ein `DISPLAY`-Statement erzeugt werden, der Platz rechts und links der Überschrift aufgefüllt wird.

Im Gegensatz zum Session-Parameter `FC` gilt dieses Zeichen für Überschriften, die über mehrere Spalten, also über eine Gruppe von Feldern, gehen.

Mögliche Werte	beliebiges Zeichen	Füllzeichen für Gruppenüberschriften.	
Standard-Einstellung	leer		
Spezifikation in Session	ja	Gültige Statements:	<code>DISPLAY</code> <code>FORMAT</code>
		Gültiges Kommando:	Keines

Beispiel:

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


116

HC - Header Centering

Überschriften-Zentrierung

Mit diesem Session-Parameter bestimmen Sie die Ausrichtung von Spaltenüberschriften.

Mögliche Werte	C	Spaltenüberschriften werden zentriert ausgegeben.	
	L	Spaltenüberschriften werden linksbündig ausgegeben.	
	R	Spaltenüberschriften werden rechtsbündig ausgegeben.	
Standard-Einstellung	C		
Spezifikation in Session	ja	Gültige Statements:	DISPLAY FORMAT
		Gültiges Kommando:	Keines

Beispiel:

```
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

Festlegung der Standard-Spaltenüberschrift

Mit diesem Session-Parameter können Sie festlegen, welche Standard-Spaltenüberschrift verwendet werden soll, wenn

- das Feld mit einem `DISPLAY`-Statement ausgegeben wird;
- ein Gleichheitszeichen (=) unmittelbar vor das Feld in einem `WRITE`- oder `INPUT`-Statement platziert wird.

Der Parameter kann angegeben werden:

- auf Feld-/Elementebene in einem `DEFINE DATA`-Statement; siehe Abschnitte *View-Definition* und *EM-, HD-, PD-Parameter für Feld/Variable*;
- im Feld `Miscellaneous` des Data Area Editor (siehe *Columns in the Editing Area*);
- in der Utility `SYSDDM`(siehe *Specifying Extended Field Attributes*).

Mögliche Werte	<code>'text'</code>	Maximal 120 alphanumerische or Unicode Zeichen.
Standard-Einstellung	keine	
Gültiges Statement:	<code>DEFINE DATA</code>	Parameter kann auf Feld-/Elementebene angegeben werden.
Gültiges Kommando:	keines	

120 HE - Helproutine

▪ HE-Parameter-Syntax	306
▪ Ausführung von Helproutinen	308
▪ Beispiele	308

Mit diesem Session-Parameter können Sie den Namen einer Helproutine angeben, die einem Feld zugewiesen werden soll.

Mögliche Werte		Siehe <i>HE-Parameter-Syntax</i> weiter unten.	
Standard-Einstellung	keine		
Spezifikation in Session	ja	Gültige Statements:	INPUT
		Gültiges Kommando:	keines

Helproutinen können mit dem Natural-Programm-Editor erstellt werden, Help Maps mit dem Natural-Map-Editor.

Die zugewiesene Helproutine kann dann bei der Verarbeitung des betreffenden INPUT-Statements oder der betreffenden Map aufgerufen werden, indem der Benutzer ein Fragezeichen (?) (das ist die Voreinstellung) in das Feld eingibt oder den Cursor in das Feld plaziert und die mit dem SET KEY-Statement definierte Hilfe-Funktionstaste drückt.

Wenn ein Wert von einer Helproutine an ein Eingabefeld übergeben werden soll, muss das Feld als änderbar (AD=M) definiert sein.

Die folgenden Themen werden nachfolgend erörtert:

HE-Parameter-Syntax

Der Parameter hat die folgende Syntax:

$$HE=operand1 \left[, \left\{ \begin{array}{l} operand2 \\ = \\ nX \end{array} \right\} \right] \dots 20$$

Operanden-Definitionstabelle:

Operand	Mögliche Struktur	Mögliche Formate	Referenzierung erlaubt	Dynam. Definition
<i>operand1</i>	C S	A	nein	nein
<i>operand2</i>	C S A	A U N P I F B D T L C O	nein	nein

Syntax-Beschreibung:

<i>operand1</i>	<p><i>operand1</i> ist der Name der Helproutine oder Map, die aufgerufen werden soll. Der Name kann eine 1 bis 8 Zeichen umfassende alphanumerische Konstante oder Benutzervariable sein. Wenn eine Variable verwendet wird, muss sie vorher definiert worden sein. Die Groß-/Kleinschreibung des Namens wird nicht verändert. Der Name kann ein Kaufmännisches Und (&) enthalten; zur Ausführungszeit wird dieses Zeichen durch den aus einem Zeichen bestehenden Code ersetzt, der dem aktuellen Wert der Systemvariablen *LANGUAGE entspricht. Diese Funktion ermöglicht die Verwendung von mehrsprachigen Helproutinen oder Maps.</p> <p>Weitere Informationen zum <i>operand1</i> in einer Map siehe die HE Helproutine-Option im Abschnitt <i>Extended Field Editing in Map Editor</i> in der <i>Editors</i>-Dokumentation.</p>
<i>operand2</i>	<p>Nach dem Helproutinen-Namen können Sie 1 bis 20 Parameter (<i>operand2</i>) angeben, die an die Helproutine übergeben werden. Diese können Konstanten oder Benutzervariablen sein, die die Parameterwerte enthalten.</p> <p>Weitere Informationen zum <i>operand2</i> in einer Map siehe die HE Helproutine-Option im Abschnitt <i>Extended Field Editing in Map Editor</i> in der <i>Editors</i>-Dokumentation.</p>
=	<p>Geben Sie ein Gleichheitszeichen (=) als Parameter an, wird der Name des Feldes (unter dem es in der Map definiert ist) an die Helproutine übergeben. Ist die Helproutine nicht einem Feld, sondern einer Map zugeordnet, wird mit = der Map-Name übergeben.</p> <ul style="list-style-type: none"> ■ Wird das Gleichheitszeichen bei HE= auf Statementebene angegeben, wird der Name des zurzeit ausgeführten Objekts (so wie er in der Systemvariablen *PROGRAM enthalten ist) an die Helproutine oder Help-Map übergeben. In Beispiel 3 ist der übergebene Objektname PROGRAM1. ■ Wird das Gleichheitszeichen bei HE= auf Elementebene (Feldebene) angegeben, wird der Name des Feldes an die Helproutine oder Help-Map übergeben. In Beispiel 3 ist der übergebene Feldname #PARM1. <p>Wird mit = ein Feld- bzw. Map-Name übergeben, so muss der entsprechende Parameter in der Helproutine mit Format/Länge A65 definiert werden.</p>
<i>nX</i>	<p>Die Notation <i>nX</i> können Sie verwenden, um wegzulassende Parameter anzugeben, das heisst, Parameter, für die keine Werte übergeben werden sollen. Die entsprechenden empfangsseitigen Parameter im DEFINE DATA PARAMETER-Statement der aufgerufenen Helproutine müssen als OPTIONAL definiert werden.</p>

**Anmerkungen:**

1. Die Operanden müssen entweder mit dem Input-Delimiterzeichen (wie mit dem Session-Parameter ID definiert) oder mit einem Komma voneinander getrennt werden. Ein Komma darf hierzu allerdings nicht verwendet werden, falls das Komma als Dezimalkomma (mit dem Session-Parameter DC) definiert ist.
2. Wenn Parameter angegeben werden, so muss die Helproutine mit einem DEFINE DATA PARAMETER-Statement beginnen, in dem Felder definiert werden, die in Format und Länge den übergebenen Parametern entsprechen.

3. Der Wert des Feldes, dem die Helproutine zugeordnet ist, kann in der Helproutine referenziert werden. Hierzu muss im `DEFINE DATA PARAMETER`-Statement der Helproutine ein Feld definiert werden, das in Format und Länge dem ursprünglichen Feld entspricht. Werden in dem `DEFINE DATA PARAMETER`-Statement noch andere Felder definiert, so muss dieses Feld immer als letztes definiert werden.
4. Ist das Feld, für das die Helproutine angegeben wird, ein Element eines Arrays, so können die Ausprägungen dieses Feldes von der Helproutine referenziert werden; hierzu müssen Sie Index-Parameter mit Format I (Integer), N (numerisch ungepackt) oder P (gepackt numerisch) am Schluss des `DEFINE DATA PARAMETER`-Statements definieren. Entsprechend der Array-Dimensionen können Sie bis zu drei Index-Parameter angeben.

Ausführung von Helproutinen

Wenn eine Helproutine — durch Eingabe eines Fragezeichens (?) in das Feld oder durch Drücken der (mit einem `SET KEY`-Statement definierten) Hilfetaste, oder über ein `REINPUT USING HELP`-Statement — aufgerufen wird, werden alle in andere Felder eingegebenen Werte erst verarbeitet, nachdem die Ausführung der Helproutine beendet ist.



Anmerkung: Pro `INPUT`-Statement ist jeweils nur eine Hilfe-Anforderung möglich. Wenn für mehrere Felder gleichzeitig Hilfe angefordert wird (z.B. durch Eingabe von Fragezeichen in mehrere Felder), wird nur die erste Hilfe-Anforderung ausgeführt.

Beispiele

Beispiel 1:

```
/* MAIN PROGRAM
DEFINE DATA
1 #A(A20/1:3)
END-DEFINE
...
SET KEY PF1=HELP
...
INPUT #A (2) (HE='HELPA',=)
...
END
```

Beispiel 2:

```

/* HELP-ROUTINE 'HELPA'
DEFINE DATA PARAMETER
1 #VARNAME (A65)
1 #PARM1 (A20)
1 #VARINDEX (I2)
END-DEFINE
...

```

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

122 HW - Heading Width

Überschriftenbreite

Mit diesem Session-Parameter bestimmen Sie die Breite einer mit einem `DISPLAY`-Statement erzeugten Ausgabespalte.

Mögliche Werte	ON	Die Breite einer <code>DISPLAY</code> -Spalte wird entweder durch die Länge des Überschriftentextes oder die Länge des Feldes bestimmt, je nachdem was länger ist. Dies gilt auch, wenn kein Überschriftentext ausgegeben wird, entweder weil das <code>DISPLAY</code> -Statement das Schlüsselwort <code>NOHDR</code> enthält, oder das <code>DISPLAY</code> -Statement ein sekundäres <code>DISPLAY</code> ist (siehe auch <code>DISPLAY</code> -Statement).	
	OFF	Die Breite einer <code>DISPLAY</code> -Spalte wird durch die Länge des Feldes bestimmt. <code>HW=OFF</code> gilt nur für <code>DISPLAY</code> -Statements, die keine Überschriften erzeugen (d.h. entweder ein erstes <code>DISPLAY</code> -Statement mit der <code>NOHDR</code> -Option oder ein sekundäres <code>DISPLAY</code> -Statement).	
Standard-Einstellung	ON		
Spezifikation in Session	ja	Gültige Statements:	<code>DISPLAY</code> <code>FORMAT</code>
		Gültiges Kommando:	Keines

Beispiel:

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


123

IA - Input Assign Character

Input-Zuweisungszeichen

Das mit diesem Session-Parameter definierte Zeichen gilt als Zuweisungszeichen für Eingabeparameter bei der Verarbeitung von `INPUT`-Statements im Keyword/Delimiter-Modus oder bei der Verarbeitung von Daten aus dem Natural-Stack.

In einer Natural-Session kann der Profilparameter `IA` durch den Session-Parameter `IA` überschrieben werden.

Mögliche Werte	jedes Sonderzeichen	Zuweisungszeichen für die Verarbeitung des Eingabeparameters in <code>INPUT</code> -Statements. Das mit dem <code>IA</code> -Parameter angegebene Zeichen <ul style="list-style-type: none"> ■ muss ein anderes sein als das mit dem Profil/Session-Parameter <code>CF</code> angegebene (Steuerzeichen für Terminalkommandos auf Großrechnern), <code>DC</code> (Dezimalzeichen) oder <code>ID</code> (<code>INPUT</code>-Delimiterzeichen); ■ sollte ein anderes sein als das mit dem <code>HI</code>-Profilparameter (Hilfezeichen) definierte. 	
Standard-Einstellung	=	Gleichheitszeichen.	
Dynamische Spezifikation	ja		
Spezifikation in Session	ja	Gültige Statements:	SET GLOBALS
		Gültiges Kommando:	GLOBALS
Programmierschnittstelle (API)	USR0350N, USR1005N *	Siehe <i>SYSEXT - Natural Application Programming Interfaces</i> in der <i>Utilities</i> -Dokumentation. * Empfohlen.	

Beispiel:

In dem folgenden Beispiel wird davon ausgegangen, dass anfänglich das standardmäßige INPUT-Zuweisungszeichen (=) gilt.

```
** 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. Geben Sie das folgende Kommando ein:

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

Das Programm erzeugt die folgende Ausgabe:

```
Page      1                                05-01-19  11:05:51
Field #A: Y
Field #B: X
```

2. Geben Sie das folgende Kommando ein:

```
GLOBALS IA=:
```

Das INPUT-Zuweisungszeichen wird auf Doppelpunkt (:) gesetzt.

3. Geben Sie dann das folgende Kommando ein:

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

Das Programm erzeugt die folgende Ausgabe:

Page 1

06-11-13 12:12:24

Field #A: Y

Field #B: X

Unter Natural Security: Die Einstellung dieses Parameters kann durch die *Session-Parameter-Option* des Library-Profiles überschrieben werden.

124 IC - Insertion Character

Einfüguingszeichen

Die mit diesem Session-Parameter angegebene Zeichenkette wird bei einem Feld, das über ein `DISPLAY`-Statement ausgegeben wird, unmittelbar vor dem Feldwert ausgegeben. Die Ausgabelänge des Feldes vergrößert sich dadurch entsprechend.

Bei numerischen Werten werden die Einfüguingszeichen vor der ersten signifikanten Stelle ausgegeben.

Die Parameter `IC` und `LC` schließen einander aus.

Der Parameter `IC` kann auch bei Feldern mit Format `U` verwendet werden.

Mögliche Werte	beliebige Zeichen	Einzufügende Zeichenkette. Sie können eine Zeichenkette von eins bis zehn Zeichen eingeben. Sie können die Zeichenkette wahlweise in Apostrophen (') angeben; in diesem Fall darf die Zeichenkette jedes beliebige Zeichen enthalten. Eine Zeichenkette, die Anführungszeichen (") oder eine schließende Klammer enthält, muss in Apostrophen stehen. Ein Leerzeichen in einer nicht durch Apostrophe eingegrenzten Zeichenkette wird durch ein Zirkumflex (^) dargestellt.	
Standard-Einstellung	keine		
Spezifikation in Session	ja	Gültige Statements:	<code>DISPLAY</code> <code>FORMAT</code>
		Gültiges Kommando:	Keines

Beispiel:

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

Siehe auch *Parameter zur Beeinflussung der Ausgabe von Feldern im Leitfaden zur Programmierung*.

125

ID - Input Delimiter Character

Input-Delimiterzeichen

Das mit dem Natural Profil- und Session-Parameter definierte Zeichen gilt als Delimiterzeichen zum Abgrenzen von Werten bei `INPUT`-Statements im Keyword/Delimiter-Modus.

In einer Natural-Session kann der Profilparameter `ID` durch den Session-Parameter `ID` überschrieben werden.

Mögliche Werte	jedes Sonderzeichen	INPUT-Delimiterzeichen. Das mit diesem Parameter angegebene Zeichen <ul style="list-style-type: none">■ darf nicht dasselbe sein wie das mit dem Profil/Session-Parameter <code>DC</code> (Dezimalzeichen) oder <code>IA</code> (INPUT-Zuweisungszeichen) angegebene Zeichen;■ sollte nicht dasselbe sein wie das mit dem <code>CF</code>-Parameter (Steuerzeichen für Großrechner-Terminalkommandos) oder <code>HI</code>-Parameter (Hilfezeichen) angegebene Zeichen. Der Punkt (.) sollte nicht als INPUT-Delimiter benutzt werden, weil dies zu Situationen führen könnte, in denen ein Programmende-Punkt fälschlicherweise als INPUT-Delimiterzeichen interpretiert wird. Ein Stern (*) sollte auch nicht verwendet werden.	
	Leerzeichen	Es wird kein INPUT-Delimiterzeichen definiert. Diese Option ist nicht möglich auf UNIX- und Windows-Plattformen.	
Standard-Einstellung	, (Komma)	Falls das Delimiterzeichen das Komma sein soll, so muss dies als dynamischer Parameter in Apostrophen <code>ID= ' , '</code> angegeben werden, da bei dynamischen Parametern das Komma als Delimiterzeichen der Eingabeparameter gilt.	
Dynamische Spezifikation	ja		
Spezifikation in Session	ja	Gültige Statements:	SET GLOBALS
		Gültiges Kommando:	GLOBALS

Programmierschnittstelle (API)	USR1005N	Siehe <i>SYSEXT - Natural Application Programming Interfaces</i> in der <i>Utilities</i> -Dokumentation.
---------------------------------------	----------	--

Unter Natural Security: Die Einstellung dieses Parameters kann durch die Session Parameters-Option des Library-Profiles überschrieben werden.

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

127

IM - Input Mode

Input-Modus

Dieser Natural Profil- und Session-Parameter bestimmt den Standardmodus für Video-Terminals.

In einer Natural-Session kann die Einstellung des Profilparameters IM durch den Session-Parameter IM überschrieben werden.

Mögliche Werte	F	Forms-Modus.	
	D	Delimiter-Modus.	
Standard-Einstellung	F		
Dynamische Spezifikation	ja		
Spezifikation in Session	ja	Gültige Statements:	SET GLOBALS
		Gültiges Kommando:	GLOBALS
Programmierschnittstelle (API)	USR1005N	Siehe <i>SYSEXT - Natural Application Programming Interfaces</i> in der <i>Utilities</i> -Dokumentation.	



Anmerkungen:

1. Die IM-Parametereinstellung kann auch mit den Natural Terminalkommandos %D und %F geändert werden.
2. Weitere Informationen zum Delimiter-Modus und Forms-Modus, entnehmen Sie dem INPUT-Statement.

Unter Natural Security: Die Einstellung dieses Parameters kann durch die Session Parameters-Option des Library-Profiles überschrieben werden.

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.



Vorsicht: 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

Eingabeaufforderungstext

Mit diesem Session-Parameter wird bei INPUT-Statements der Text, der zur Eingabe auffordert, gesteuert.

Mögliche Werte	ON	Eingabe-/Ausgabefeldern eines INPUT-Statements, denen kein Text-element vorangestellt ist, wird der betreffende Feldname vorangestellt.	
	OFF	Es wird kein Eingabeaufforderungstext in Form von Feldnamen generiert; nur wenn einem Feld explizit ein Textelement vorangestellt ist, wird dieser Text als Eingabeaufforderung ausgegeben.	
Standard-Einstellung	ON		
Spezifikation in Session	ja	Gültige Statements:	FORMAT INPUT
		Gültiges Kommando:	Keines

Beispiel:

```
FORMAT IP=OFF
```


131 IS - Identical Suppress

Unterdrückung identischer Werte

Mit diesem Session-Parameter können Sie die mehrfache Ausgabe identischer Feldwerte in aufeinander folgenden Zeilen bei einem `WRITE-` oder `DISPLAY-`Statement unterdrücken.

Mit dem Statement `SUSPEND IDENTICAL SUPPRESS` können Sie die Wirkung von `IS=ON` für einen einzelnen Datensatz unterdrücken.

Der `IS`-Parameter kann in Verbindung mit den Parametern `ES` und `ZP` zur Unterdrückung der Ausgabe von Leerzeilen eingesetzt werden.

Mögliche Werte	ON	Wenn <code>IS=ON</code> gesetzt ist, wird der Wert eines Feldes nicht angezeigt, falls er mit dem vorherigen Wert des Feldes identisch ist. Erzeugt ein <code>DISPLAY-</code> oder <code>WRITE-</code> Statement unter Verwendung der <code>VERT-</code> Option oder der Schrägstrich-Notation (<code>/</code>) mehrzeilige Ausgaben, so gilt <code>IS=ON</code> nur für die jeweils erste Zeile.	
	OFF	Es wird keine automatische Unterdrückung angewendet.	
Standard-Einstellung	OFF		
Spezifikation in Session	ja	Gültige Statements:	<code>DISPLAY</code> <code>FORMAT</code> <code>WRITE</code>
		Gültiges Kommando:	Keines

Beispiel:

```
FORMAT IS=ON
```

Siehe auch *Parameter zur Beeinflussung der Ausgabe von Feldern im Leitfaden zur Programmierung*.

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	



Anmerkung: 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	If session initialization errors occur, the following happens: In online mode: the initialization errors are displayed and you can choose to either continue or terminate the session. In batch mode: the session is continued with the initialization errors going unnoticed - possibly leading to errors or undesired results later in the session. The setting <code>ITERM=OFF</code> is not possible when an <code>INPL</code> command is placed on the Natural command stack at the beginning of the Natural session, that is, with <code>STACK=INPL</code> .
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.

 **Wichtig:** 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

PF-Tasten-Anzeige

Dieser Session-Parameter dient dazu, die PF-Tasten zugewiesenen Namen (siehe SET KEY-Statement) anzeigen zu lassen.

Wenn `KD=ON` gesetzt ist, werden diese Informationen automatisch bei jeder mit `INPUT`, `WRITE`, `DISPLAY` und `PRINT` erzeugten Ausgabe am unteren Bildschirmrand angezeigt.

Da diese Anzeige zwei Zeilen in Anspruch nimmt, muss die logische Seitenlänge (siehe Session-Parameter `PS`) entsprechend um zwei Zeilen reduziert werden.

Mögliche Werte	ON	Die den PF-Tasten zugewiesenen Namen werden angezeigt.	
	OFF	Die den PF-Tasten zugewiesenen Namen werden nicht angezeigt.	
Standard-Einstellung	OFF		
Spezifikation in Session	ja	Gültige Statements:	FORMAT
		Gültiges Kommando:	Keines

Beispiel:

```
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.



Anmerkung: 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.



Anmerkungen:

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

Vorangestellte Zeichen

Die mit diesem Session-Parameter angegebene Zeichenkette wird bei einem Feld, das über ein `DISPLAY`-Statement ausgegeben wird, unmittelbar vor dem Feld ausgegeben. Die Breite der Ausgabespalte vergrößert sich dadurch entsprechend.

Die Session-Parameter `LC` und `IC` schließen einander aus.

Der Parameter `LC` kann auch mit Felder des Formats `U` benutzt werden. Informationen zu Unicode-Format entnehmen Sie dem Dokument *Unicode and Code Page Support in the Natural Programming Language, Session Parameters, EMU, ICU, LCU, TCU versus EM, IC, LC, TC*.

Mögliche Werte	beliebige Zeichen	<p>Sie können eine Zeichenkette von 1 bis 10 Zeichen definieren.</p> <p>Sie können die Zeichenkette wahlweise in Apostrophen (') angeben; in diesem Fall darf die Zeichenkette jedes beliebige Zeichen enthalten. Eine Zeichenkette, die Anführungszeichen (") oder eine schließende Klammer enthält, muss in Apostrophen stehen. Ein Leerzeichen in einer nicht durch Apostrophe eingegrenzten Zeichenkette wird durch ein Zirkumflex (^) dargestellt.</p>	
Standard-Einstellung	Keine		
Spezifikation in Session	ja	Gültige Statements:	DISPLAY FORMAT
		Gültiges Kommando:	Keines

Beispiel:

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

Siehe auch *Parameter zur Beeinflussung der Ausgabe von Feldern im Leitfaden zur Programmierung*.

139

LE - Reaction when Limit for Processing Loop Exceeded

Reaktion auf Limit-Überschreitung bei Verarbeitungsschleifen

Mit diesem Natural Profil- und Session-Parameter bestimmen Sie, was geschehen soll, wenn bei der Ausführung einer Verarbeitungsschleife das angegebene Limit (d.h. die maximale Anzahl der Schleifendurchläufe) bei einem Statement `READ`, `FIND` oder `HISTOGRAM` erreicht wird.

Das Limit kann entweder ein (mit dem `LIMIT`-Statement angegebenes) globales Limit oder ein schleifenspezifisch festgesetztes Limit sein.

In einer Natural-Session kann der Profilparameter `LE` durch den Session-Parameter `LE` überschrieben werden.

Mögliche Werte	ON	Die Schleife wird beendet, wenn das Limit erreicht ist. Das Programm wird dann mit dem auf die beendete Schleife folgenden Statement fortgesetzt. Nach Beendigung der Ausführung des Natural-Objekts, wird der Fehler NAT0957 (Database loop limit reached with 'LE=ON'.) ausgegeben. LE=ON gilt nur für Programme, die von einer in der Systemdatei <code>FUSER</code> abgelegten Library geladen werden, d.h. der Library <code>SYSTEM</code> , oder einer Library mit einem Namen, der nicht mit dem Präfix <code>SYS</code> anfängt.	
	OFF	Die Schleife wird beendet, wenn das Limit erreicht ist. Das Programm wird dann mit dem auf die beendete Schleife folgenden Statement fortgesetzt. Nach Beendigung der Ausführung des Natural-Objekts, wird keine Fehlermeldung ausgegeben.	
Standard-Einstellung	OFF		
Dynamische Spezifikation	ja		
Spezifikation in Session	ja	Gültige Statements:	SET GLOBALS
		Gültiges Kommando:	GLOBALS

Programmierschnittstelle (API)	USR1005N	Siehe SYSEXT - <i>Natural Application Programming Interfaces</i> in der <i>Utilities-Dokumentation</i> .
---------------------------------------	----------	--

Beispiel:

```
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: nach 10 Datensätzen wird die Schleife ohne eine Meldung beendet.

LE=ON: nach 10 Datensätzen wird die Schleife mit einer Fehlermeldung NAT0957 (Database loop limit reached with 'LE=ON') beendet.

140

LFIL - Logical System File Definition

▪ LFIL Parameter Syntax	353
▪ NTLFIL Macro Syntax	353
▪ Old NTLFIL Macro Syntax	353
▪ Example of LFIL Parameter	353
▪ Example of NTLFIL 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 NTFILF 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. * Recommended.	
	USR2004N *		

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



Anmerkung: 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.

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.



Vorsicht: 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

▪ Profilparameter LS	362
▪ Session-Parameter LS	362
▪ Spezifikation in Statements	363

Zeilenlänge

Mit diesem Natural Profil- und Session-Parameter bestimmen Sie, wieviele Stellen eine von einem DISPLAY-, INPUT- oder WRITE-Statement erzeugte Zeile höchstens lang sein darf.

Die folgenden Themen werden behandelt:

Profilparameter LS

Wenn LS als Profilparameter benutzt wird, kommt LS im Batch-Betrieb zum Einsatz und definiert die physische Zeilenlänge. Im Online-Betrieb ist die Zeilenlänge immer auf die physische Bildschirmbreite gesetzt.

Mögliche Werte	35 - 250	Höchstwert der pro Zeile zulässigen Zeichen.
	0	Physische Zeilenlänge benutzen (meistens 132).
Standard-Einstellung	0	
Dynamische Spezifikation	ja	

Session-Parameter LS

Mögliche Werte	2 - 250	Höchstwert der pro Zeile zulässigen Zeichen.
Standard-Einstellung	0	Physische Zeilenlänge.
Gültiges Kommando	GLOBALS	
Gültige Statements	FORMAT SET GLOBALS	
Programmierschnittstelle (API)	USR1005N	Siehe <i>SYSEXT - Natural Application Programming Interfaces</i> in der <i>Utilities</i> -Dokumentation.



Anmerkung: Beim Logon in eine Library wird LS auf die physische Zeilenlänge zurückgesetzt.

Unter Natural Security: Die Einstellung dieses Parameters kann durch die Session Parameters-Option des Library-Profiles überschrieben werden.

Spezifikation in Statements

Wenn er in einem Statement angegeben wird, wird der LS-Parameter bei der Kompilierung ausgewertet.

Gültige Statements	DISPLAY INPUT WRITE	Der Parameter kann auf Statement-Ebene angegeben werden.
---------------------------	---------------------------	--

145

LT - Limit for Processing Loops

Limit für Verarbeitungsschleifen

Mit diesem Natural Profil- und Session-Parameter können Sie das allgemeine Limit für Verarbeitungsschleifen in Natural-Programmen bestimmen, d.h. wieviele Datensätze eine Verarbeitungsschleife in einer Natural-Anwendung maximal verarbeiten darf.

Dieses Limit gilt für alle Statements, mit denen Datensätze aus der Datenbank gelesen werden, d.h.:

- Statements, die eine Verarbeitungsschleife auslösen, wie zum Beispiel `READ`, `FIND`, `HISTOGRAM` oder `SELECT` und
- Statements, mit denen nur ein einzelner Datensatz gelesen wird, wie zum Beispiel `FIND UNIQUE`, `FIND NUMBER`, `FIND FIRST`, `GET (SAME)` und `SELECT SINGLE`.

Alle gelesenen Datensätze werden gezählt und das Ergebnis dieser Zählung wird mit dem im LT-Parameter gesetzten Wert verglichen. Hierbei werden alle gelesenen Datensätze mitgezählt, auch solche, die aufgrund einer `WHERE`-Klausel eines `FIND`-, `READ`- oder `HISTOGRAM`-Statements zurückgewiesen und nicht weiterverarbeitet werden. Das mit dem LT-Parameter gesetzte Limit hat keine Auswirkung auf die Statements `STORE`, `UPDATE`, `DELETE`, `END TRANSACTION` und `BACKOUT TRANSACTION`.

Wird ein Datensatz von der Datenbank gelesen, wird die Anzahl der gelesenen Datensätze erhöht, bevor sie mit dem Wert des LT-Parameters verglichen wird. Übersteigt der erhöhte Zählwert den aktuellen LT-Wert, wird der Natural-Fehler NAT1003 (Global limit for database calls reached) ausgegeben. Der Wert der gezählten gelesenen Datensätze wird immer dann auf Null gesetzt, wenn ein Natural-Programm auf Level 1 gestartet wird. Der Wert wird allerdings nicht zurückgesetzt, wenn das Programm auf Level 1 ein anderes Natural-Objekt aufruft (weitere Informationen hierzu siehe *Mehrere Stufen (Levels) aufgerufener Objekte* im Leitfaden zur Programmierung). Deshalb wird mit dem LT-Parameter die Anzahl der Datensätze begrenzt, die von einem Level 1-Programm und von den Objekten gelesenen wurden, die von diesem Programm auf einem anderen Level als Level 1 aufgerufen wurden.

Wird der Wert des LT-Parameters dynamisch in einem Programm mittels einem SET GLOBALS LT=*n*-Statement geändert, wird der neue Limit-Wert beim nächsten Statement wirksam, das einen Datensatz von der Datenbank liest.

In einer Natural-Session kann der Profilparameter LT durch den Session-Parameter LT überschrieben werden.

Mögliche Werte	0 - 2147483647 In einer Session: 0 bis <i>n</i> (<i>n</i> = Wert des Profilparameters LT beim Start der Session)	Maximale Anzahl der Datensätze, die in einer gegebenen Verarbeitungsschleife gelesen werden können. Hierbei werden alle gelesenen Datensätze mitgezählt, auch solche, die aufgrund einer WHERE-Klausel zurückgewiesen und nicht weiterverarbeitet werden. LT=0 legt fest, dass für Verarbeitungsschleifen kein Limit gelten soll.	
Standard-Einstellung	99999999		
Dynamische Spezifikation	ja		
Spezifikation in Session	ja	Gültige Statements:	SET GLOBALS, siehe Anmerkung.
		Gültiges Kommando:	GLOBALS, siehe Anmerkung.
Programmierschnittstelle (API)	USR1005N	Siehe SYSEXT - <i>Natural Application Programming Interfaces</i> in der <i>Utilities</i> -Dokumentation.	



Anmerkung: Wird der LT-Parameter mit einem SET GLOBALS-Statement oder einem SET GLOBALS-Systemkommando benutzt, kann der einstellbare Limit-Wert nicht größer als der LT-Wert sein, der im Natural-Parametermodul NATPARM gesetzt wurde.

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. * Recommended.
	USR1068N *	

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. * Recommended.
	USR1068N *	

150

MAXROLL - Number of CMROLL Calls before Session Suspension

This Natural profile parameter only applies under Complete 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 Complete) 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 Complete 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.



Anmerkung: 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

 **Vorsicht:** 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

Anzahl multipler Feldwerte



Anmerkung: Dieser Parameter darf nur im Reporting Mode verwendet werden.

Mit diesem Session-Parameter geben Sie an, wieviele Werte eines multiplen Feldes standardmäßig ausgegeben werden sollen, wenn das Feld ohne Index in einem `DISPLAY-` oder `WRITE-Statement` angegeben ist.

Mögliche Werte	0 - 191	Anzahl der Werte. Wenn MC=0 angegeben wird, dann gibt es keinen standardmäßigen Indexbereich für die Ausgabe eines MU-Feldes. Wird ein MU-Feld ausgegeben, ist es deshalb erforderlich, einen expliziten Index oder Indexbereich anzugeben, sonst tritt ein Syntaxfehler (NAT0281) auf.	
Standard-Einstellung	1		
Spezifikation in Session	ja	Gültige Statements:	DISPLAY FORMAT INPUT PRINT WRITE
		Gültiges Kommando:	Keines.

Beispiel:

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).

154 ML - Position of Message Line

Meldungszeilen-Position

Dieser Profilparameter gibt die Zeile an, die benutzt werden soll für die Anzeige von Anwendungen, die die Meldungszeilen-Position nicht explizit mittels des `SET CONTROL 'M'`-Statements setzen. Informationen zum Operanden 'M' siehe auch Natural Terminalkommando %M.

Mögliche Werte	B	Natural-Meldungen werden am unteren Rand des Bildschirms angezeigt.	
	T	Natural-Meldungen werden am oberen Rand des Bildschirms angezeigt.	
Standard-Einstellung	T		
Dynamische Spezifikation	ja		
Spezifikation in Session	ja	Gültige Statements:	SET CONTROL 'M'
		Gültiges Kommando:	
Programmierschnittstelle (API)	USR1005N	Siehe <i>SYSEXT - Natural Application Programming Interfaces</i> in der <i>Utilities</i> -Dokumentation.	

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

Maximale Seitenzahl eines Reports

Mit diesem Natural Profil- und Session-Parameter bestimmen Sie, wieviele Seiten ein Report höchstens erzeugen darf.

In einer Natural-Session kann die Einstellung des Profilparameters MP vom `FORMAT`-Statement verkleinert aber nicht vergrößert werden. Der mit dem Session-Parameter MP angegebene Wert gilt nur für den angegebenen Report.

Mögliche Werte	1 - 99999	Die angegebene Zahl bezieht sich auf die Anzahl der physischen Seiten und ist unabhängig von der Seitennummer der Startseite. Wird die maximale Seitenzahl überschritten, so wird das Programm mit einer Fehlermeldung abgebrochen.	
	0	Es ist keine maximale Seitenzahl definiert.	
Standard-Einstellung	0		
Dynamische Spezifikation	ja		
Spezifikation in Session	nein	Gültige Statements:	DISPLAY FORMAT PRINT WRITE
		Gültiges Kommando:	Keines.

157 MS - Manual Skip

Manuelle Cursor-Positionierung

Mit diesem Session-Parameter steuern Sie die Positionierung des Cursors bei der Verarbeitung eines INPUT-Statements.

Mögliche Werte	ON	Siehe Beispiel weiter unten. Anmerkung: Die Einstellung MS=ON wird unter BS2000/OSD nicht unterstützt.	
	OFF	Ist MS=OFF gesetzt, so wird der Cursor ins nächste Eingabefeld plaziert, sobald der Wert des aktuellen Feldes vollständig eingegeben ist.	
Standard-Einstellung	OFF		
Spezifikation in Session	ja	Gültige Statements:	FORMAT INPUT
		Gültiges Kommando:	Keines.

Beispiel:

```
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

Maximale CPU-Zeit

Dieser Natural Profil- und Session-Parameter gilt nur für im Batch-Betrieb unter Natural Development Server (SPoD) oder unter Natural for TSO ausgeführte Programme.

Mit diesem Session-Parameter bestimmen Sie, wieviel CPU-Zeit ein Natural-Programm in Anspruch nehmen darf.

Die CPU-Zeitmessung beginnt, wenn ein Natural-Programm von der NEXT-Zeile oder mittels eines FETCH-Statements gestartet wird, d.h. auf Programmebene 1. Im Nicht-Batch-Betrieb (Natural Development Server, Natural for TSO) wird die CPU-Zeitmessung bei jeder Terminal I/O neu gestartet.

In einer Natural-Session kann der Profilparameter MT durch den Session-Parameter MT überschrieben werden.

Mögliche Werte	1 - 9999999	Maximale CPU-Zeit in Sekunden. Wenn Natural Security installiert ist, kann der Profilparameter MT in Natural Security überschrieben werden. Mit Natural Security ist der Höchstwert für den Profilparameter MT gleich 32767. Um einen höheren Wert zu benutzen als mit dem MT Profil- oder Session-Parameter angegeben, spezifizieren Sie MT=0 in Natural Security.	
	0	MT=0 legt fest, dass kein Natural CPU-Zeitlimit gelten soll.	
Standard-Einstellung	60		
Dynamische Spezifikation	ja		
Spezifikation in Session	ja	Gültige Statements:	SET GLOBALS
		Gültiges Kommando:	GLOBALS
Programmierschnittstelle (API)	USR1005N	Siehe SYSEXT - <i>Natural Application Programming Interfaces</i> in der <i>Utilities</i> -Dokumentation.	



Anmerkungen:

1. Das Limit für im interaktiven Betrieb laufende Programme wird von dem benutzten TP-Monitor gesteuert.
2. Der benutzbare Höchstwert wird von der jeweiligen Betriebssystemumgebung festgelegt. Überschreitet der mit dem MT-Parameter gesetzte Wert das vom Betriebssystem erlaubte Maximum, wird der Wert entsprechend der Betriebssystem-Vorgaben verringert.
3. Bei Systemumgebungen, die keine CPU-Zeitmessung unterstützen, wird das Limit als die verstrichene Zeit interpretiert. Bei Systemen ohne Zeitmessung wird das CPU-Zeitlimit ignoriert.



Wichtig: In Server-Umgebungen, in denen der Server selbst ohne betriebssystemgesteuertes CPU-Zeitlimit läuft, empfiehlt es sich sehr, den Profilparameter MT auf einen Nicht-Nullwert zu setzen, um die Bildung endloser Schleifen (bspw. aufgrund von Anwendungsfehlern) zu vermeiden. Diese Empfehlung gilt für Natural RPC- und Natural Development-Server.

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

Verwendung von Systemkommandos

Mit diesem Natural Profil- und Session-Parameter bestimmen Sie, ob Natural-Systemkommandos während der Natural- Session verwendet werden können.

In einer Natural-Session kann der Profilparameter NC durch den Session-Parameter NC überschrieben werden.


Mögliche Werte	ON	Systemkommandos können nicht verwendet werden – außer FIN, LAST, LOGOFF, LOGON, MAINMENU, RENUMBER, RETURN, SETUP und TECH. Ist Natural Security installiert, sind ungeachtet der Einstellung des NC-Profilparameters alle von Ihnen mit Natural Security gesetzten Systemkommando-Beschränkungen (Command Restrictions) gültig. In einer Natural Development Server-Umgebung auf Großrechnern wird der Wert OFF für den Natural Development Server angenommen, auch wenn NC=ON angegeben wurde. Wurde NC=ON client-seitig angegeben, werden clientseitig abgesetzte Systemkommandos im Anschluss daran zurückgewiesen (siehe oben).	
	OFF	Alle Systemkommandos können verwendet werden.	
Standard-Einstellung	OFF		
Dynamische Spezifikation	ja		
Spezifikation in Session	ja	Gültige Statements:	SET GLOBALS
		Gültiges Kommando:	
Programmierschnittstelle (API)	USR1005N	Siehe SYSEXT - <i>Natural Application Programming Interfaces</i> in der <i>Utilities</i> -Dokumentation.	



Anmerkung: Natural-Terminalkommandos und Benutzerkommandos (Objektmodul-Namen) sind vom NC-Parameter nicht betroffen.

163 NISN (Internal Use)

This parameter is reserved for internal use by Natural.

 **Vorsicht:** Do not change its setting.

164 NL - Numeric Length for Output

Numerische Länge der Ausgabe

Mit diesem Session-Parameter bestimmen Sie die Standard-Eingabe-/Ausgabelänge eines numerischen Feldes, das in einem DISPLAY-, INPUT-, PRINT- oder WRITE-Statement verwendet wird.

Der Parameter NL darf nicht für Gruppen angegeben werden.

Eine für ein Feld definierte Editiermaske setzt den NL-Parameter für dieses Feld außer Kraft.

Mögliche Werte	<i>nn.m</i>	<p>Die Länge wird in der Form <i>nn.m</i> angegeben, wobei <i>nn</i> für die Stellen vor dem Komma (Dezimalpunkt) und <i>m</i> für die Stellen nach dem Komma steht.</p> <p>Die Angabe von Stellen nach dem Komma ist nicht erforderlich. <i>m</i> darf nicht größer als 7 sein. Insgesamt darf die Länge von <i>nn</i> und <i>m</i> zusammen 29 Stellen nicht überschreiten.</p> <p>Anmerkung:</p> <ol style="list-style-type: none"> 1. Ist die NL-Länge kleiner als die Feldlänge, werden die ausgegebenen Werte entsprechend abgeschnitten, ohne dass dies zu einer Fehlermeldung führt. 2. Ist die NL-Länge größer als die Feldlänge, werden die freien Stellen mit Leerzeichen aufgefüllt, und es führt zu keinem Fehler, wenn ein Eingabefeld abgeschnitten wird. 	
Standard-Einstellung	Keine		
Spezifikation in Session	ja	Gültige Statements:	DISPLAY FORMAT INPUT PRINT WRITE
		Gültiges Kommando:	Keines

Beispiel:

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

Siehe auch *Parameter zur Beeinflussung der Ausgabe von Feldern im Leitfaden zur Programmierung*.

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.



Anmerkung: 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	

Überschreiben geschützter Felder durch Helproutinen

Mit diesem Natural Profil- und Session-Parameter bestimmen Sie, ob der Inhalt eines schreibgeschützten Feldes (Attribut `AD=P`) durch eine Helproutine, die dem Feld zugewiesen ist, überschrieben werden kann.

In einer Natural-Session kann der Profilparameter `OPF` durch den Session-Parameter `OPF` überschrieben werden.

Mögliche Werte	ON	Eine einem Feld zugewiesene Helproutine kann den Inhalt des Feldes Werte überschreiben, selbst wenn das Feld schreibgeschützt ist.	
	OFF	Die Inhalte schreibgeschützter Felder können nicht durch Helproutinen überschrieben werden.	
Standard-Einstellung	ON		
Dynamische Spezifikation	ja		
Spezifikation in Session	ja	Gültige Statements:	SET GLOBALS
		Gültiges Kommando:	GLOBALS
Programmierschnittstelle (API)	USR1005N	Siehe <i>SYSEXT - Natural Application Programming Interfaces</i> in der <i>Utilities</i> -Dokumentation.	



Anmerkungen:

1. Der Profilparameter `OPF` gilt nur für das Feld, für das eine Helproutine aufgerufen wird; er gilt nicht für Parameter, die explizit an die Helproutine übergeben werden. Das bedeutet, dass der `OPF`-Parameter wirkungslos bleibt, falls Sie das Feld, für das Hilfe aufgerufen wird, auch noch explizit als an die Helproutine zu übergebenden Parameter angegeben haben.
2. Des weiteren können Sie im Reporting Mode die `OPF`-Einstellung mittels des Statements `SET GLOBALS` ändern.

169

OPRB - Database Open/Close Processing

▪ 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`.

- [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. Anmerkung: 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. Anmerkung: 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 <code>NATPARM</code> , the macro <code>NTOPT</code> must be used instead.
Specification within session	yes	

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, Complete 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 Complete/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](#).

175 PC - Periodic Group Count

Anzahl der Periodengruppen-Ausprägungen

Dieser Session-Parameter kann nur im Reporting Mode verwendet werden. Er dient zur Festlegung der Anzahl der Periodengruppen-Ausprägungen, die standardmäßig ausgegeben werden, wenn eine Periodengruppe (oder ein in einer Periodengruppe enthaltenes Feld) in einem DISPLAY- oder WRITE-Statement ohne Index angegeben wird.

Mögliche Werte	0 - 191	Anzahl der Werte. Wenn PC=0 angegeben wird, dann gibt es keinen standardmäßigen Indexbereich für die Ausgabe eines PE-Feldes. Wird ein PE-Feld ausgegeben, ist es deshalb erforderlich, einen expliziten Index oder Indexbereich anzugeben. Sonst tritt ein Syntaxfehler (NAT0281) auf.
Standard-Einstellung	1	
Gültige Statements	FORMAT	
	INPUT DISPLAY WRITE PRINT	Parameter kann auf Statement- und/oder Elementebene (Feldebene) angegeben werden.
Gültiges Kommando:	Keines	

Beispiel:

```
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	The default setting (according to ASA standard settings) is: ' 0 - ' which means that a blank causes a line advance of 1 line, "0" of 2 lines and "-" of 3 lines. Vorsicht: In any IBM environment, do not change the default setting of this parameter.
Dynamic specification	yes	
Specification within session	no	

177

PD - Limit of Pages for NATPAGE

Seiten-Limit für NATPAGE

Mit diesem Natural Profil- und Session-Parameter bestimmen Sie die maximale Anzahl der Seiten (Schirme), die mit der Utility NATPAGE auf einmal in der Natural-Systemdatei (FUSER) aufgezeichnet werden können.

In einer Natural-Session kann der Profilparameter PD durch den Session-Parameter PD überschrieben werden.

Mögliche Werte	0 - 255	Maximale Anzahl der Seiten (Bildschirme).	
Standard-Einstellung	50		
Dynamische Spezifikation	ja		
Spezifikation in Session	ja	Gültige Statements:	SET GLOBALS
		Gültiges Kommando:	GLOBALS



Anmerkungen:

1. Wenn diese Anzahl der aufgezeichneten Bildschirme überschritten wird, wird durch jeden weiteren Schirm ein bereits aufgezeichneter überschrieben, wobei die ältesten, d.h. zuerst aufgezeichneten Schirme nach und nach überschrieben werden („Wrap-Around“-Verfahren).
2. Weitere Informationen zur Utility NATPAGE finden Sie unter den Terminalkommandos %E, %I, %O, %P und %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 <code>CMPLOG</code> . (If <code>CMPLOG</code> is not available, the list is written to the standard output dataset <code>CMPRINT</code> .)
		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 <code>CMPLOG</code> . (If <code>CMPLOG</code> 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 <code>PLOG</code> parameter applies to all subsequent dynamic profile parameters until the next <code>PLOG</code> 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 <code>PLOG=OFF</code> specification and a subsequent <code>PLOG=ON</code> 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	



Anmerkung: 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

■ Profilparameter PM	440
■ Session-Parameter PM	441

Druck-Modus

Folgende Themen werden behandelt:

Profilparameter PM

Mit diesem Profilparameter bestimmen Sie, wie Felder gedruckt oder angezeigt werden sollen.

Mögliche Werte	C, P, I, R, oder Kombinationen	PM=C	Es soll ein alternativer Zeichensatz verwendet werden. Dieser kann mit den Profilparametern TAB1 und TAB2 festgelegt werden.
	CI, CR, PI, PR	PM=P	Es soll der primäre (standardmäßige) Zeichensatz verwendet werden.
		PM=I	Invertierte (von rechts nach links verlaufende) Anzeigerichtung (z.B. zur Anwendung in Ländern des Nahen Ostens).
		PM=R	Rücksetzen von der Einstellung PM=I auf die normale Anzeigerichtung (von links nach rechts).
Standard-Einstellung	PR		
Dynamische Spezifikation	ja		
Spezifikation in Session	ja		Systemkommando GLOBALS oder Terminalkommando %V.
Programmierschnittstelle (API)	USR1005N		Siehe <i>SYSEXT - Natural Application Programming Interfaces</i> in der <i>Utilities</i> -Dokumentation.

PM=I betrifft alle vom System gesteuerten Ausgabeschirmbestandteile, das heisst, Systemvariablen und PF-Tastenzeilen. Darüber hinaus sind von dieser Einstellung alle nicht-alphanumerischen Felder, z.B. numerische Felder und Datumsfelder, betroffen. Außerdem wird bei Natural Web I/O Interface-Terminals die Anzeigerichtung in den Feldern von links nach rechts in rechts nach links geändert. Die Routine für die Feldinvertierung wird in der Natural-Source-Library als Assembler-Modul NATPM ausgeliefert und kann bei Bedarf geändert werden.

Ausführliche Informationen über die Verwendung der Einstellung PM=I siehe *Bidirectional Language Support* in der *Unicode and Code Page Support*-Dokumentation.

Session-Parameter PM

Mit diesem Session-Parameter bestimmen Sie, wie Felder angezeigt werden sollen.

Mögliche Werte	PM=C	(Kann nur auf Großrechnern gesetzt werden.) Es wird ein alternativer Zeichensatz verwendet (siehe Modul NATPM in der Natural-Source-Library).
	PM=D	(Kann nur auf Großrechnern gesetzt werden.) Definiert reine DBCS-Felder, die keine Shift-Out/Shift-In-Zeichen enthalten (siehe <i>Double-Byte Character Sets</i> in der <i>Operations-Dokumentation</i>).
	PM=I	Feldwerte werden in invertierter Richtung, d.h. von rechts nach links, angezeigt (zum Beispiel für Länder des Nahen Ostens).
	PM=N	Von der Anzeige kann keine Hardcopy gemacht werden.
Standard-Einstellung	Keine	Es wird der Standard-Zeichensatz verwendet.
Gültige Statements	DEFINE DATA DISPLAY FORMAT INPUT MOVE LEFT/RIGHT JUSTIFIED PRINT WRITE	



Anmerkung: Es kann mehr als ein Wert angegeben werden.

Beispiel:

```
LIMIT 1
  READ EMPLOYEES
  DISPLAY NOTITLE NAME
  DISPLAY NOTITLE NAME (PM=I)
  DISPLAY NOTITLE NAME
  END
```

Ergebnis:

NAME	

MORENO	
	ONEROM
MORENO	

181

POS22 - Version 2.2 Algorithm for POS System Function

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 Anmerkung: 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.



Vorsicht: 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	446
▪ 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	456
▪ Keyword Subparameters for AM=STD in BS2000/OSD Environments	457
▪ Keyword Subparameters for AM=CICS	458
▪ Keyword Subparameters for AM=COMP (Com-plete)	459
▪ Keyword Subparameters for AM=SMARTS (Com-plete)	459
▪ Keyword Subparameters for AM=IMS	460
▪ Keyword Subparameters for DEFINE PRINTER Statement	460

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`).

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.

Value	Meaning
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.

 **Anmerkung:** PRINT=OFF is equivalent to: PRINT=((1-31)), AM=OFF). It does not affect any of the other keyword subparameter specifications. PRINT=((0), AM=xxx) or NTPRINT ((0), AM=xxx) determines the hardcopy print access method and is equivalent to the profile parameter HCAM=xxx.

DEST - External Dataset Name

DEST=*name* specifies the print destination (1 - 8 characters).

This corresponds to the OUTPUT value of the DEFINE PRINTER statement (and can be overwritten by a DEFINE PRINTER OUTPUT specification).

The meaning of this keyword subparameter depends on the access method.

Access Method	Meaning of DEST
AM=STD	DEST is the logical dataset name (DDNAME, LINK name, DTF name). If the destination is to be for multiple files, two asterisks (**) have to be specified for the file number. These will be replaced by the corresponding logical file number for each print file. A DEST value including two asterisks must be enclosed in apostrophes when it is used as a dynamic parameter. The default value is DEST='CMPRT**' for IBM and DEST='P**' for SIEMENS environments. Under z/VSE, only 7-character names are supported.
AM=CICS	There is no default value for print files under CICS. Here, the DEST subparameter is mandatory, that is, CICS print files defined without a valid DEST specification are ignored. The Natural CICS interface also supports a variable (see TERMVAR parameter in the NCIPARM generation macro; &TID is the default) as part of the DEST value which, when being specified, is replaced by the actual CICS terminal ID. See also <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.



Anmerkung: PRINT=((0),DEST=xxx) or NTPRINT (0),DEST=xxx determines the hardcopy print destination and is equivalent to the Natural profile parameter HCDEST=xxx.

OPEN - Time of File Opening

OPEN=xxx determines when the file is to be opened:

Value	The file is opened
INIT	for output at session initialization.
OBF	according to the default OPEN value for the different environments (batch, CICS, Com-plete, TSO).
OBJ	when the execution of the first object which accesses the file starts. This is the general default, except for AM=COMP and AM=IMS.
OBJ1	when the execution of the first object on Level 1 that accesses the file starts. Otherwise, it is opened when it is first accessed.
ACC	when it is first accessed by a statement. This is the default for AM=COMP and AM=IMS.
INITOBF	for output at session initialization. Any subsequent re-opening of the file sets the default OPEN value for the different environments (batch, CICS, Com-plete, TSO).
INITOBJ	for output at session initialization. Any subsequent re-opening of the file will be performed when the execution of the first object which accesses the file starts.
INITOBJ1	when the execution of the first object on Level 1 that accesses the file starts. Otherwise, it is opened when it is first accessed.
INITACC	for output at session initialization. Any subsequent re-opening of the file will be performed when it is first accessed by a statement.

CLOSE - Time of File Closure

CLOSE=xxx determines when the file is to be closed:

Value	The file is closed
OBJ	either when processing of the object in which it was first accessed is finished or when command mode, NEXT mode or MAINMENU is reached.
CMD	when command mode, NEXT mode or MAINMENU is reached. This is the default for AM=NAF, AM=COMP and AM=IMS.
FIN	at session end (this is the default for AM=STD). With CLOSE=FIN, a DEFINE PRINTER statement causes an error if the printer was opened already. A CLOSE PRINTER statement for the printer is ignored.
USER	only if the file is open and one of the following conditions is true: <ul style="list-style-type: none"> ■ a CLOSE PRINTER statement is issued, ■ a DEFINE PRINTER statement is issued, ■ the session terminates.

ROUTE - Logical Print File Routing

ROUTE=xxx determines whether logical print file routing is done according to the OUTPUT clause of the DEFINE PRINTER statement.

ON	Print file routing is done. The target print file can be any available print file except PC. This is the default value.
OFF	No print file routing is done.
am	Print file routing is done to printers of the specified access method am only. Possible value is any valid print file access method (see description of subparameter AM above). PC is not allowed for am.

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.



Anmerkung: 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	Natural takes the current TP user ID (as contained in the system variable *INIT-USER) as profile name, which means that the profile defined under the name corresponding to that ID is used. If no such profile is found, a profile named AUTO is used instead (if available). You can define such an AUTO profile as default profile for users without individual profiles.
--------------	---

PROFILE=TERMINAL	<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

Länge einer Reportseite

Mit diesem Natural Profil- und Session-Parameter bestimmen Sie, wieviele Zeilen eine von einem DISPLAY- oder WRITE-Statement erzeugte Reportseite höchstens haben darf.

Wenn er als Profilparameter benutzt wird, kommt der PS-Parameter im Batch-Betrieb zum Einsatz und definiert die physische Seitenlänge. Im Online-Betrieb wird die physische Seitenlänge stets auf die physische Bildschirmhöhe gesetzt.

Mögliche Werte	1 - 250	Maximale Anzahl der Zeilen pro Seite.	
	0	<p>Die physische Seitenlänge soll benutzt werden.</p> <p>Wenn PS=0 für den ersten auszugebenden Report (Report 0) angegeben wird, wird die physische Seitenlänge des verwendeten Geräts minus 1 verwendet.</p> <p>Wenn PS=0 für die Reports 1 – 31 angegeben wird, führt dies dazu, dass die automatische Verarbeitung einer neuen Seite (Newpage) unterdrückt wird, d.h. es wird keine automatische Seitenvorschub-Verarbeitung ausgeführt.</p>	
Standard-Einstellung	0		
Dynamische Spezifikation	ja		
Spezifikation in Session	ja	Gültige Statements:	DISPLAY FORMAT INPUT SET GLOBALS WRITE
		Gültiges Kommando:	GLOBALS
Programmierschnittstelle (API)	USR1005N	Siehe SYSEXT - <i>Natural Application Programming Interfaces</i> in der <i>Utilities</i> -Dokumentation.	

Siehe auch *Seitengröße – der PS-Parameter im Leitfaden zur Programmierung*.

Unter Natural Security: Die Einstellung dieses Parameters kann durch die Session Parameters-Option des Library- Profils überschrieben werden.

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	

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.



Vorsicht: Do not change its setting.

192

RDCEXIT - Define Natural Data Collector User Exits

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 RDCEXIT, any exit names can be specified. In the Natural parameter module NATPARM, the exit names are automatically added to the CSTATIC profile parameter's setting list.
Default setting	none	
Dynamic specification	yes	If RDCEXIT is specified dynamically, the exits must be defined in the profile parameters CSTATIC or RCA (RCA can also be specified dynamically). Optionally, the size of the exit work area can be specified after the exit name.
	Possible settings	400 - 32760
	Default setting	400
	Example	RDCEXIT=(MYEXIT,2000,RDCEX1)
Specification within session	no	

For details, refer to *Debugging and Monitoring* in the *SYSRDC Utility* documentation.

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 <code>RDCSIZE=2</code> . To also activate the trace recording, you have to set <code>RDCSIZE</code> 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.



Vorsicht: Do not change its setting.

195

RDPORT (Internal Use)

This parameter is reserved for internal use by Natural.



Vorsicht: Do not change its setting.

196

RDS - Define Remote Directory Server

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

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, 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.

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>Wichtig: 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>Anmerkung: 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>Anmerkung: 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 in the Utilities</i> documentation.

199

REINP - Issue Internal REINPUT Statement for Invalid

Data

Interner REINPUT bei ungültigen Daten

Dieser Natural Profil- und Session-Parameter kann benutzt werden, um einen internen REINPUT für ungültige Daten zu verhindern.

Standardmäßig führt Natural automatisch ein internes REINPUT-Statement aus, wenn auf ein INPUT-Statement hin ungültige Daten eingegeben werden. Mit diesem Session-Parameter können Sie diesen Automatismus ausschalten. Dadurch haben Sie die Möglichkeit, solche Eingabefehler in Ihrer Anwendung selbst zu verarbeiten.

In einer Natural-Session kann der Profilparameter REINP durch den Session-Parameter REINP überschrieben werden.

Mögliche Werte	ON	Bei Eingabe ungültiger Daten wird ein internes REINPUT-Statement ausgeführt.	
	OFF	Bei Eingabe ungültiger Daten wird kein internes REINPUT-Statement ausgeführt.	
Standard-Einstellung	ON		
Dynamische Spezifikation	ja		
Spezifikation in Session	ja	Gültige Statements:	SET GLOBALS
		Gültiges Kommando:	GLOBALS
Programmierschnittstelle (API)	USR1005N	Siehe SYSEXT - <i>Natural Application Programming Interfaces</i> in der <i>Utilities</i> -Dokumentation.	

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 WHERE clause or an ACCEPT/REJECT statement). This reduces the number of ISNs which are contained in the hold queue. This may, however, cause additional performance overhead as an Adabas call is required for each ISN released.
	OFF	The ISN of each record which has been placed in hold status is <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	



Anmerkungen:

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`.

206

RPC - Remote-Procedure-Call Settings

▪ 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	

RPC Parameter Syntax

The parameter syntax of RPC is as follows:

```
RPC=(keyword_subparameter1=value,keyword_subparameter2=value,...)
```

keyword_subparameter - see *Keyword Subparameters* below.

NTRPC Macro Syntax

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

```
NTRPC keyword_subparameter1=value,keyword_subparameter2=value,...
```

keyword_subparameter - see below.

Keyword Subparameters

There are three groups of keyword subparameters available that apply to

- **both Client and Server**
(`ACIVERS` | `MAXBUFF` | `RPCSIZE` | `SERVER` | `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)* documentation.

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 <code>CALLNAT</code> to be executed by the Natural RPC server will be translated into upper case before the <code>CALLNAT</code> is invoked. With this option, non-Natural RPC clients are supported that use mixed case characters in the subprogram names. Anmerkung: On UNIX, OpenVMS and Windows platforms, an implicit upper case translation is already done by Natural itself. <code>RPCUCT=ON</code> 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 <code>CALLNAT</code> 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	14 - 64	Buffer size in KB.
Default setting	20	
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

Terminal-Warnton

Dieser Natural Profil- und Session-Parameter gibt an, ob die Terminal-Alarmfunktion benutzt werden soll.

In einer Natural-Session kann der Profilparameter SA durch den Session-Parameter SA überschrieben werden.

Mögliche Werte	ON	Der Warnton wird jedesmal ausgelöst, wenn der Benutzer von Natural zu einer Eingabe aufgefordert wird.. Anmerkung: Für den Einsatz dieser Funktion ist es erforderlich, dass die Hardware-Funktion für den Terminal-Alarm installiert wurde.	
	OFF	Der Warnton wird zur Eingabeaufforderung nicht ausgelöst. Allerdings kann der Warnton nach wie vor über die SOUND ALARM-Klausel eines REINPUT-Statements ausgelöst werden.	
Standard-Einstellung	OFF		
Dynamische Spezifikation	ja		
Spezifikation in Session	ja	Gültige Statements:	SET GLOBALS
		Gültiges Kommando:	GLOBALS
Programmierschnittstelle (API)	USR1005N	Siehe SYSEXT - <i>Natural Application Programming Interfaces</i> in der <i>Utilities</i> -Dokumentation.	

212

SB - Selection Box

▪ Anmerkungen zur Syntax	526
▪ Anmerkungen zur Laufzeitumgebung	527
▪ Funktionen	528
▪ Beschränkungen	530

Auswahlfeld

Auswahlfelder stehen in einem INPUT-Statement ausschließlich auf Großrechnern zur Verfügung. Für andere Plattformen können Auswahlfelder nur im Map-Editor definiert werden.

Auswahlfelder können an Eingabefelder angehängt werden. Sie bilden eine komfortable Alternative zu an Feldern angehängte Helprountinen, da Sie ein Auswahlfeld unmittelbar in Ihrem Programm kodieren können. Sie brauchen kein zusätzliches Programm wie bei Helprountinen.

Sie können eine Auswahlfeld-Klausel für jede INPUT-Variable vom Typ Alpha definieren, ungeachtet der Tatsache, ob dieses Feld ein Eingabe- oder Ausgabefeld oder beides ist.

Hierzu sieht die Syntax wie folgt aus:

```
SB=operand1 [,operand1]...
```

wobei *operand1* einen Wert-Operanden darstellt, der zum Auffüllen des Auswahlfeldes mit Elementen benutzt wird.

Operand	Mögliche Struktur	Mögliche Formate	Referenzierung erlaubt	Dynam. Definition
<i>operand1</i>	C S A	A	yes	no

Bei SB geben Sie die anzuzeigenden Werte innerhalb des Auswahlfeldes ein.

Um ein Auswahlfeld einem Feld zuzuweisen

geben Sie das Attribut SB für ein INPUT-Alphafeld in Ihrem Natural-Programm mittels der folgenden Beispiel-Syntax ein:

```
INPUT #FLD (SB='value1', #ITEM1, #ITEM2(1:3), #ITEM3(*))
```

Folgende Themen werden behandelt:

Anmerkungen zur Syntax

Es ist möglich, einem Feld sowohl ein Auswahlfeld als auch eine Helprountine zuzuweisen.

Auswahlfelder können für jedes Variablenfeld in einem INPUT-Statement definiert werden. Dazu gibt es folgende Ausnahmen:

Systemvariablen	Zum Beispiel: *PROGRAM, *COM
Benannte Konstanten (nur auf Großrechnern)	Definiert mit einer CONST-Klausel in DEFINE DATA-Statement.

Außer dem SB-Attribut können auch andere Attribute definiert werden, z.B. AD oder CD.

Das Auswahlfeld muss nicht änderbar sein, wie dies bei AD=A oder AD=M der Fall ist. Mit anderen Worten ist es möglich, auch für ein schreibgeschütztes Ausgabefeld wie AD=0 ein Auswahlfeld zur Verfügung zu stellen (und Werte auszuwählen). Wenn Sie AD=0 benutzen, ist der Benutzer dazu gezwungen, aus einer Reihe von vordefinierten Werten zu wählen, die ihrerseits in einem Auswahlfeld erscheinen.

Anmerkungen zur Laufzeitumgebung

Auswahlfeld-Position

Wenn ein ein Auswahlfeld enthaltendes Programm ausgeführt wird, wird das Auswahlfeld nach demselben Positionierungsalgorithmus auf dem Bildschirm positioniert, der für Hilfefenster verwendet wird, d.h. die Länge und Position des Auswahlfeldes wird automatisch festgelegt, *in der Nähe* des Feldes.

Auswahlfeld-Attribute

Die Farbe und die intensivierten Attribute, die dem Feld zugewiesen sind, werden auch für die in dem entsprechenden Auswahlfeld angezeigten Werte benutzt.

Editiermasken in Auswahlfeldern

Wenn für das Feld eine Editiermaske definiert worden ist, wird die Editiermaske für alle Auswahlfeld-Werte verwendet.

Um eine Editiermaske für ein Feld zu definieren, benutzen Sie das INPUT-Statement, damit Sie eine Editiermaske für ein Feld definieren können. Dies ist in dem folgenden Code-Beispiel veranschaulicht.

```
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
```

Zeilenlänge für Auswahlfelder

Die Zeilenlänge des Auswahlfeldes stimmt mit der Feldlänge überein, der das Auswahlfeld entspricht.

Wenn ein für das Auswahlfeld vorgesehener Wert die Zeilenlänge des Auswahlfeldes überschreitet, wird der Wert abgeschnitten.

Reihenfolge der Auswahlfeld-Werte

Auswahlfeld-Werte werden in der Reihenfolge angezeigt, in der sie im SB-Attribut erscheinen.

Funktionen

Kennzeichnung eines Auswahlfeldes bei der Anzeige

Für ein Feld mit angehängtem Auswahlfeld wird neben dem Feld ein „V“-Indikator angezeigt.

Auswahlfelder aufrufen

Um ein Auswahlfeld zu öffnen, gibt es zwei Möglichkeiten:

- Geben Sie ein Fragezeichen (?) in das V-Feld ein, und drücken Sie FREIG.
- Oder positionieren Sie den Cursor auf dem V-Feld, drücken Sie die entsprechend zugewiesene Hilfetaste (z.B. PF1). Weitere Einzelheiten siehe den nächsten Abschnitt.

Um eine Hilfetaste (z.B. PF1) für einen komfortableren Aufruf des Auswahlfeldes zu definieren:

Fügen Sie in Ihrem Programm die folgende Codezeile hinzu:

```
SET KEY PF1=HELP
```

Auswahlfeld durchblättern

Es gibt zwei Möglichkeiten, in einem Auswahlfeld zu blättern:

- Indem Sie den Cursor auf der „Mehr“-Zeile positionieren und FREIG drücken.
- Oder mit den Terminalkommandos %W- und %W+, die mit PF-Tasten belegt sind (z.B. PF7/PF8).

Wert in einem Auswahlfeld auswählen

Ein Wert wird vom Auswahlfeld ausgewählt und durch Positionieren des Cursors auf dem Wert und Drücken von FREIG in das Feld kopiert.

Doppelte Zeilen in einem Auswahlfeld

Zeilen mit demselben Inhalt, die direkt aufeinander folgen, werden unterdrückt.

Beispielsweise erzeugt der folgende Code

```
INPUT #FLD (SB='123', '456', 'XYZ', 'XYZ', 'XYZ', 'ABC', 'DEF')
```

die folgende Ausgabe im Auswahlfeld:

```
123  
456  
XYZ  
ABC  
DEF
```

In dem vorigen Beispiel wird XYZ nur einmal angezeigt. Die anderen Ausprägungen werden als redundant angesehen, da sie direkt aufeinander folgen.

Allerdings erzeugt diese Codezeile jetzt

```
INPUT #FLD (SB='123', 'XYZ', '456', 'XYZ', 'ABC', 'XYZ', 'DEF')
```

die folgende Ausgabe im Auswahlfeld:

```
123  
XYZ  
456  
XYZ  
ABC  
XYZ  
DEF
```

In diesem Fall werden alle drei Ausprägungen von XYZ angezeigt, da sie nicht direkt aufeinander folgen.

Leerzeilen in Auswahlfeldern

Eine Leerzeile wird nur angezeigt, wenn sie zum ersten Mal erscheint; alle nachfolgenden Leerzeilen werden unterdrückt.

Beschränkungen

Die Anzahl der Operanden in der SB-Klausel ist auf 20 beschränkt.

Die maximale Anzahl der Werte in einem Auswahlfeld ist 248. Wenn diese Grenze erreicht ist, werden weitere Werte nicht angezeigt. Es wird keine Fehlermeldung ausgegeben, wenn die Grenze überschritten wurde.

213

SCTAB - Scanner Characters

- SCTAB Parameter Syntax 532
- NTSCTAB Macro Syntax 533
- Example of NTSCTAB Macro 533
- Example of SCTAB Parameter 533

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	

SCTAB Parameter Syntax

The `SCTAB` parameter is specified as follows:

```
SCTAB=(character1,attribute-type1,attribute-type2,...,character2,attribute-type1,attribute-type2,...)
```

character

You specify a character, and after it its attribute type(s).

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

attribute-type(s)

Attribute types can be:

UPPER	upper-case alphabetical
LOWER	lower-case alphabetical
NUM	numeric
HEX	hexadecimal
ALFANUM	alphanumeric
SPECIAL	special
NDELIM	non-delimiter

It is possible to specify more than one character in the list of values. You must enclose the entire string of character/attribute pairs in parentheses.

NTSCTAB Macro Syntax

The NTSCTAB macro is specified as follows:

```
NTSCTAB character1,attribute-type1,attribute-type2,...  
NTSCTAB character2,attribute-type1,attribute-type2,... ..
```

For each character to be overwritten, you have to specify a separate NTSCTAB macro.

Example of NTSCTAB Macro

```
NTSCTAB 5E,LOWER,NDELIM  
NTSCTAB '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>TTYPE=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.
<i>open</i> 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</i> UTM-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

Spaltenabstand

Mit diesem Natural Profil- und Session-Parameter bestimmen Sie, wieviele Leerstellen zwischen zwei Ausgabespalten eines mit einem DISPLAY-Statement erzeugten Natural- Reports standardmäßig eingefügt werden sollen.

In einer Natural-Session kann der Profilparameter SF durch den Session-Parameter SF überschrieben werden.

Mögliche Werte	1 - 30	Anzahl der Leerstellen. Anmerkung: Der SF-Parameter kann nicht auf Null (0) gesetzt werden, d.h. es muss wenigstens ein Leerzeichen zwischen Report-Spalten stehen.	
Standard-Einstellung	1		
Dynamische Spezifikation	ja		
Spezifikation in Session	ja	Gültige Statements:	SET GLOBALS
		Gültiges Kommando:	GLOBALS
Programmierschnittstelle (API)	USR1005N	Siehe SYSEXT - <i>Natural Application Programming Interfaces</i> in der <i>Utilities</i> -Dokumentation.	

Unter Natural Security: Die Einstellung dieses Parameters kann durch die *Session Parameters*-Option des Library-Profiles überschrieben werden.

Siehe auch *Column Spacing - SF Parameter and nX Notation* im Leitfaden zur Programmierung.

217 SG - Sign Position

Vorzeichen-Stelle

Mit diesem Session-Parameter bestimmen Sie, ob einem numerischen Feld eine zusätzliche Stelle zur Anzeige des Vorzeichens vorangestellt werden soll.

Wenn der EM-Parameter (Editiermodus) angegeben wird, überschreibt er den SG-Parameter.

Mögliche Werte	ON	Eine Vorzeichen-Stelle wird belegt.	
	OFF	Keine Vorzeichen-Stelle wird belegt. Wenn SG=OFF gesetzt ist, werden negative Werte ohne das Minuszeichen (-) ausgegeben. SG=OFF verhindert nicht die Eingabe von negativen Werten in Eingabefeldern.	
Standard-Einstellung	ON		
Spezifikation in Session	ja	Gültige Statements:	DISPLAY FORMAT INPUT PRINT WRITE
		Gültiges Kommando:	Keines

Beispiel:

```
FORMAT SG=OFF
```

Siehe auch *Parameter zur Beeinflussung der Ausgabe von Feldern im Leitfaden zur Programmierung*.

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.



Vorsicht: 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

Sourcecode-Zeilenlänge

Dieser Natural Profil- und Session-Parameter gibt die Anzahl der Zeichen an, die in jeder Sourcecode-Zeile von Natural interpretiert werden sollen. Dies gilt auch für den Zeilenmodus-Editor, der mit dem Systemkommando EDT aktiviert wird.

In einer Natural-Session kann der Profilparameter SL durch den Session-Parameter SL überschrieben werden.

Mögliche Werte	20 - 250	Im Batch-Betrieb:	Die Anzahl der in jeder Zeile in den Dateien CMSYNIN und CMOBJIN zu verarbeitenden Zeichen. Einzelheiten zu diesen Dateien siehe betriebssystemspezifische Teile des Abschnitts <i>Natural in Batch Mode</i> in der <i>Operations</i> -Dokumentation.
		Im Online-Betrieb:	Die Anzahl der zu interpretierenden Zeichen, wenn Sie den Natural-Programmeditor im EDT-Modus benutzen (der mit dem Systemkommando EDT aktiviert wurde).
Standard-Einstellung	72		
Dynamische Spezifikation	ja		
Spezifikation in Session	ja	Gültige Statements:	Keines.
		Gültiges Kommando:	GLOBALS
Programmierschnittstelle (API)	USR1005N	Siehe SYSEXT - <i>Natural Application Programming Interfaces</i> in der <i>Utilities</i> -Dokumentation.	

Unter Natural Security: Die Einstellung dieses Parameters kann durch die Session Parameters-Option des Library-Profiles überschrieben werden.

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	

Programmierung im Structured Mode

Mit diesem Natural Profil- und Session-Parameter bestimmen Sie, ob im Structured Mode programmiert werden muss oder nicht.

In einer Natural-Session kann der Profilparameter `SM=OFF` durch den Session-Parameter `SM=ON` überschrieben werden.

Mögliche Werte	ON	Die Programmierung darf nur in Structured Mode-Syntax erfolgen. Anmerkung: Falls mit dem Profilparameter <code>SM</code> festgelegt wird, dass im Structured Mode programmiert werden muss <code>SM=ON</code> , wird ein Versuch, diese Einstellung während der Session per Systemkommando <code>GLOBALS</code> und Session-Parameter <code>SM</code> zu ändern mit einer Fehlermeldung zurückgewiesen (<code>Reporting mode not permitted</code>).	
	OFF	Die Programmierung kann entweder in Structured Mode oder in Reporting Mode erfolgen.	
Standard-Einstellung	OFF		
Dynamische Spezifikation	ja		
Spezifikation in Session	ja	Gültige Statements:	Keine
		Gültiges Kommando:	GLOBALS
Programmierschnittstelle (API)	USR1005N	Siehe <i>SYSEXT - Natural Application Programming Interfaces</i> in der <i>Utilities</i> -Dokumentation.	

Falls Natural Security installiert ist:

- legt die Einstellung der Modus-Option im Security-Profil der Library fest, ob der Profilparameter `SM` benutzt werden kann (siehe auch *Programming mode* in der *Natural Security*-Dokumentation;
- kann es sein, dass dieser Parameter im Security-Profil einer Library deaktiviert ist; ist dies der Fall, so gilt für die Library unabänderlich Structured Mode.

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.



Vorsicht: 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 <code>NTSORT</code> must be used instead.
Specification within session	no	

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 Complete 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/ '
```

EXTE0J - Action in the Event of an Error

EXTE0J 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 562
- Conversion of Logical Shift-Out/Shift-In Characters 563
- Compatibility of SOSI Profile Parameter and Obsolete SO and SI Profile Parameters 563
- Automatic Adaptation of Translation Tables 564
- SOSI Parameter Examples 564

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 <i>Positional Subparameters</i> below.
Default setting	none	
Dynamic specification	yes	
Specification within session	no	

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>

	The default value is the logical shift-out character.
<i>logical-shift-in</i>	The logical shift-in character must be a single character. Specify the hexadecimal representation of the logical shift-in character. Usually, the value 0F is used for IBM hardware and the value 29 is used for Fujitsu hardware.
<i>physical-shift-in</i>	The value of the physical shift-in character must be chosen depending on the screen hardware that is used. The length of the physical shift-in character may be one or two bytes. Specify the hexadecimal representation of the physical shift-in character. The default value is the logical shift-in character.
<i>S0/SI-display-length</i>	The number of bytes occupied on the screen by the physical shift-out/shift-in characters. Possible values are 0 and 1. The default value is 1. For IBM hardware, the value 1 must be used. For Fujitsu hardware, the value 0 must be used.

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.

Compatibility of SOSI Profile Parameter and Obsolete S0 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).

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 translated for input and output.

For detailed information on the translation tables, see *Translation Tables* in the *Operations* documentation.

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 additional information on Natural RPC, 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.



Anmerkung: 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	592
▪ NTSYS Macro Syntax	593
▪ Example of NTSYS Macro	593

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>	<i>set-name</i> (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	



Anmerkung: 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](#), [FSPool](#)) which have been password-protected.

Possible settings	8 characters	If a Natural system file is password-protected, a password which permits updates to the file must be specified. The password specified with the SYSPSW parameter applies to all Natural system files for which no individual passwords are specified. If the OPRB parameter is specified, the SYSPSW password is used for the initial Adabas open call and must permit access and/or update to all the files specified in OPRB as required.
	blanks	If the Natural system files are not password-protected, set SYSPSW to blanks.
Default setting	blanks	
Dynamic specification	yes	If you specify SYSPSW dynamically in conjunction with any of the individual system file parameters FNAT , FUSER , FDIC , FSEC and FSPool , you must specify SYSPSW <i>before</i> any individual system file parameter.
Specification within session	no	



Anmerkung: Passwords for individual system files can be specified with the profile parameters [FNAT](#), [FUSER](#), [FDIC](#), [FSEC](#) and [FSPool](#).

242

TAB - Standard Output Character Translation

▪ TAB Parameter Syntax	600
▪ NTTAB Syntax	600
▪ Example of NTTAB Macro	601
▪ Example of TAB Parameter	601

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 `NATPARAM`.

Possible settings		See TAB Parameter Syntax 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>NATPARAM</code> , the macro <code>NTTAB</code> must be used instead.
Specification within session	no	

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	604
▪ NTTAB1 Macro Syntax	604
▪ Example of NTTAB1 Macro	605
▪ Example of TAB1 Parameter	605

This Natural profile parameter allows you to overwrite the definitions in the translation table NNTAB1 as contained in the configuration module NATCONFIG. The NNTAB1 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 NNTAB1 macro in the Natural parameter module NATPARAM.

Possible settings		See <i>TAB1 Parameter Syntax</i> below.
Default setting		As specified within the macro NNTAB1 in NATCONFIG.
Dynamic specification	yes	This parameter can only be specified dynamically. In the Natural parameter module NATPARAM, the macro NNTAB1 must be used instead.
Specification within session	no	

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.

NNTAB1 Macro Syntax

The NNTAB1 macro is specified as follows:

```
NNTAB1 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 608
- NTTAB2 Macro Syntax 608
- Example of NTTAB2 Macro 609
- Example of TAB2 Parameter 609

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 NATPARAM.

Possible settings		See <i>TAB2 Parameter Syntax</i> 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 NATPARAM, the macro NTTAB2 must be used instead.
Specification within session	no	

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	612
▪ NTTABA1 Macro Syntax	612
▪ Example of NTTABA1 Macro	613
▪ Example of TABA1 Parameter	613

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 <i>TABA1 Parameter Syntax</i> 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	

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')
```


246

TABA2 - ASCII-to-EBCDIC Translation

- TABA2 Parameter Syntax 616
- NTTABA2 Macro Syntax 616
- Example of NTTABA2 Macro 617
- Example of TABA2 Parameter 617

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	

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')
```


247

TABL - SYS Library Output Translation

- TABL Parameter Syntax 620
- NTTABL Macro Syntax 620
- Example of NTTABL Macro 621
- Example of TABL Parameter 621

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 <i>TABL Parameter Syntax</i> 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	

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

Nachgestellte Zeichen

Die mit diesem Session-Parameter angegebene Zeichenkette wird bei einem Feld, das über ein DISPLAY-Statement ausgegeben wird, unmittelbar hinter dem Feld angezeigt. Die Breite der Ausgabespalte vergrößert sich dadurch entsprechend.

Der Parameter TC kann auch für Felder des Formats U benutzt werden. Weitere Informationen zum Unicode-Format entnehmen Sie dem Abschnitt *Unicode and Code Page Support in the Natural Programming Language, Session Parameters, EMU, ICU, LCU, TCU versus EM, IC, LC, TC*.

Mögliche Werte	beliebige(s) Zeichen	Sie können eine Zeichenkette von 1 bis 10 Zeichen definieren. Sie können die Zeichenkette wahlweise in Apostrophen angeben; in diesem Fall darf die Zeichenkette jedes beliebige Zeichen enthalten. Eine Zeichenkette, die Anführungszeichen oder ein schließendes Klammerzeichen enthält, muss in Apostrophen stehen.	
Standard-Einstellung	Keine		
Spezifikation in Session	ja	Gültige Statements:	DISPLAY FORMAT
		Gültiges Kommando:	Keines

Beispiele:

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

Siehe auch *Parameter zur Beeinflussung der Ausgabe von Feldern im Leitfaden zur Programmierung*.

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 <i>SYSEXT - 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 629
- NTTF Macro Syntax 629
- Example of TF Parameter 629
- Example of NTTF Macro 629



Vorsicht: 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. * Recommended.
	USR2005N *	

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.

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

Tausender-Trennzeichen

Mit diesem Profil- und Session-Parameter definieren Sie das zur Laufzeit als Tausender-Trennzeichen (Thousands Separator) zu verwendende Zeichen. Das Tausender-Trennzeichen dient dann zum Ersetzen der dynamischen Tausender-Trennzeichen (Dynamic Thousands Separators) in Editiermasken.



Anmerkung: In der Natural-Source wird das dynamische Tausender-Trennzeichen immer durch ein Komma (,) oder einen Punkt (.) dargestellt.

Mögliche Werte	beliebiges Zeichen	Zur Laufzeit wird das dynamische Tausender-Trennzeichen durch dieses Zeichen ersetzt. Soll das Tausender-Trennzeichen durch ein Komma ersetzt werden, muss das Komma in Anführungszeichen gesetzt werden, d.h. THSEPCH=' , ', wenn die dynamische Parameterfunktion benutzt wird, um einzelne Parameter von einander abzutrennen. Wenn das Tausender-Trennzeichen ein Anführungszeichen (') sein soll, dann muss es als zwei Anführungszeichen angegeben werden, die wiederum in Anführungszeichen gesetzt sein müssen, d.h. THSEPCH=' '' '.	
Standard-Einstellung	, (Komma)	Standardmäßig wird als Tausender-Trennzeichen ein Komma verwendet.	
Dynamische Spezifikation	ja		
Spezifikation in Session	ja	Gültige Statements:	keine
		Gültiges Kommando:	GLOBALS

Siehe auch:

- Option THSEP des Systemkommandos COMPOPT in der *Systemkommandos*-Dokumentation.

- Subparameter `THSEP` des Profilparameters `CMPO` bzw. des Macros `NTCMPO`.
- *Trennzeichen-Angaben an lokale Standards anpassen im Leitfaden zur Programmierung.*

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	
-------------------------------------	----	--


**Anmerkungen:**

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.

 **Vorsicht:** Do not change its setting.

255

TQ - Translate Quotation Marks

This parameter has been replaced by subparameter TQMARK of profile parameter **CMPO**.

256

TRACE - Define Trace Level for Natural RPC Servers

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

257

TRACE - Define Components to be Traced

▪ TRACE Parameter Syntax	644
▪ NTTRACE Macro Syntax	645
▪ Example of TRACE Parameter	645
▪ Example of NTTRACE Macro	645

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`.



Vorsicht: 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.

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
```


258

TRANSP - Server Transport Protocol

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

Konvertierung von Systemdatei-Programmausgaben

 **Wichtig:** Der TS-Parameter gilt nur für primäre Ausgaben (CMPRINT, siehe *Natural in Batch Mode* in der *Operations*-Dokumentation).


Dieser Natural Profil- und Session-Parameter dient dazu, Ausgaben von Natural-Systemdateien (d.h. Dateien, deren Namen mit SYS anfangen) mittels einer Umsetzungstabelle umzusetzen. Dies ist gegebenenfalls bei nicht-standardmäßiger Verwendung von Kleinbuchstaben (z.B. bei Ländern des Nahen Ostens) erforderlich.

Fehlermeldungen oder Warnungen werden nur umgesetzt, wenn die englische Version des Texts angezeigt wird und die Natural-Session nicht mit englischem Sprachcode abläuft (ULANG=1). Wenn der Text angezeigt wird in der lokalen Sprache (zum Beispiel Hebräisch), wird er nicht in Großbuchstaben umgesetzt. Die Umsetzung von Meldungen und Warnungen ist nicht abhängig von der Library, von wo das Programm ausgeführt wird.

In einer Natural-Session kann der Profilparameter TS durch den Session-Parameter TS überschrieben werden.

Mögliche Werte	ON	Die Ausgabe wird umgesetzt. Bei TS=ON werden der Profilparameter LC=OFF und der Session-Parameter AD=T ignoriert, die beide Eingaben in Großbuchstaben konvertieren, da sie eine unerwünschte Zeichen-Konvertierung für spezifische Zeichensätze verursachen würden	
	OFF	Die Ausgabe wird nicht umgesetzt.	
Standard-Einstellung	OFF		
Dynamische Spezifikation	ja		
Spezifikation in Session	ja	Gültige Statements:	SET GLOBALS

		Gültiges Kommando:	GLOBALS
Programmierschnittstelle (API)	USR1005N	Siehe SYSEXT - <i>Natural Application Programming Interfaces</i> in der <i>Utilities</i> -Dokumentation.	

 **Anmerkung:** Die Umsetzungstabelle kann mit dem [NTTABL](#)-Makro oder dem betreffenden dynamischen Profilparameter [TABL](#) geändert werden.

Unterstützung von TS=ON bei Natural unter IMS/TM-Meldungen

Alle Natural unter IMS/TM-Meldungen werden in Großbuchstaben umgesetzt, wenn in der Natural-Session TS=ON angegeben wird.

Unterstützung von TS=ON beim RPC Server Trace

Alle Meldungen des Natural RPC Server Trace werden in Großbuchstaben umgesetzt, wenn in der Natural RPC Server Session TS=ON angegeben wird. Die Trace-Daten vom/zum Client sind von der Einstellung TS=ON nicht betroffen und bleiben unverändert.

Weitere Parameter für die Umsetzung in Großbuchstaben

Zusätzlich zur Auswertung von TS=ON bieten einige Natural-Komponenten einen UCTRAN-Parameter, der die Umsetzung von Meldungen in Großbuchstaben auch dann bewirkt, wenn die Einstellung des Parameters TS nicht (oder noch nicht) zur Verfügung steht. Diese Komponenten sind:

- Authorized Services Manager
- Roll Server
- Global Buffer Pool Manager unter z/OS and z/VSE
- Natural Com-plete/SMARTS Interface
- Natural Remote Procedure Call

Siehe *Startup Parameters* in *z/OS Batch Mode* und *Startup Parameters* unter CICS in der *Natural Remote Procedure Call*-Dokumentation.

Beim Natural Development Server gibt es den Konfigurationsparameter `UPPERCASE_SYSTEMMESSAGES`, der eine ähnliche Funktion bietet, siehe *Configuring the Natural Development Server* in der *Natural Development Server*-Dokumentation.

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 - 128	Buffer size in KB. If the requested space is not available, the Adabas Text Retrieval facility cannot be used.
	0	Adabas Text Retrieval facility is not used.
Default setting	0	
Dynamic specification	yes	
Specification within session	no	

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 TTYPE 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 TTYPE parameter has the same function as the terminal command %T=.



Anmerkung: If you use the TTYPE parameter, it is no longer necessary to execute a program containing a SET CONTROL 'T=...' statement at the start of the session in order to set the terminal type.

263 UC - Underlining Character

Unterstreichungszeichen

Das mit diesem Session-Parameter angegebene Zeichen wird als Unterstreichungszeichen verwendet für:

- Spaltenüberschriften, die von `DISPLAY`-Statements generiert werden;
- Seitenüberschriften/-unterschriften, die über `WRITE TITLE`-/`WRITE TRAILER`-Statements mit `UNDERLINED`-Option erzeugt werden.

Mögliche Werte	beliebiges Zeichen	Siehe auch die folgende Anmerkung.	
	OFF		
Standard-Einstellung	- (Bindestrich)		
Spezifikation in Session	ja	Gültige Statements:	<code>DISPLAY</code> <code>FORMAT</code> <code>WRITE TITLE</code> <code>WRITE TRAILER</code>
		Gültiges Kommando:	Keines



Anmerkung: Falls Sie keine Unterstreichung von Spaltenüberschriften wünschen, haben Sie folgende Möglichkeiten:

- `UC=` — Statt einer Unterstreichung wird eine Leerzeile ausgegeben.
- `UC=OFF` — Die Feldwerte werden unmittelbar unter der Überschrift, ohne Leerzeile dazwischen, ausgegeben.

Sie können `UC=OFF` nur auf `Statement`-Ebene eines `DISPLAY`-Statements angeben; in diesem Fall können Sie für einzelne Felder in dem Statement keine anderen `UC`-Angaben machen.

Beispiele:

```
FORMAT UC=*  
DISPLAY (UC= ) NAME AGE (UC=+)
```

Siehe auch *Unterstreichungszeichen für Überschriften – UC-Parameter* im Leitfaden zur Programmierung.

264 UDB - User Database ID

This Natural profile parameter specifies the DBID to be used for a database access.

Possible settings	0 - 65535, except 255	Valid database ID. Database ID 255 is reserved for logical system files for Software AG products, see profile parameter LFILE .
Default setting	database ID applicable for FUSER	
Dynamic specification	yes	
Specification within session	no	
Application Programming Interface	USR1005N	See <i>SYSEXT - Natural Application Programming Interfaces</i> in the <i>Utilities</i> documentation. * Recommended.
	USR1040N *	



Anmerkungen:

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.



Anmerkung: 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*.

266

UPSI - z/VSE User Program Switches

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	667
▪ NTUSER Macro Syntax	667
▪ Example of NTUSER Macro	667
▪ Example of USER Parameter	667

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.

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.



Vorsicht: Do not change its setting.

269

UTAB1 - Lower-to-Upper-Case Translation

- UTAB1 Parameter Syntax 672
- NTUTAB1 Macro Syntax 672
- Example of NTUTAB1 Macro 673
- Example of UTAB1 Parameter 673

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	

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')
```


270

UTAB2 - Upper-to-Lower-Case Translation

- UTAB2 Parameter Syntax 676
- NTUTAB2 Macro Syntax 676
- Example of NTUTAB2 Macro 677
- Example of UTAB2 Parameter 677

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	

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	



Anmerkung: 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.

272

WEBIO - Web I/O Interface Screen Rendering

- WEBIO Parameter Syntax 682
- NTWEBIO Macro Syntax 682
- Keyword Subparameters 683

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 <i>Keyword Subparameters</i> , below.	Possible subparameter keywords: <code>ML KEYS WIN</code>
Default setting	<code>ML=OFF , KEYS=OFF , WIN=OFF</code>	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 <code>WEBIO</code> can only be specified dynamically. In <code>NATPARM</code> , use the macro <code>NTWEBIO</code> .
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

Warten auf Datensatz im Hold

Dieser Natural Profil- und Session-Parameter gilt nur für Adabas-Datenbanken.

Er bestimmt, was geschehen soll, wenn ein angeforderter Datensatz von einem anderen Benutzer ins Hold gestellt wurde.

In einer Natural-Session kann der Profilparameter WH durch den Session-Parameter WH überschrieben werden.

Mögliche Werte	ON	Der Benutzer wird solange in den Wartestatus versetzt, bis entweder der angeforderte Datensatz wieder verfügbar ist oder eine Datenbanksystem- Zeitüberschreitung oder Überschreitung eines anderen Limits beim Versuch, den Datensatz ins Hold zu stellen, zur Ausgabe einer entsprechenden Fehlermeldung führt.	
	OFF	Eine Fehlermeldung wird ausgegeben, falls einer der gewünschten Datensätze nicht in den Hold-Status gestellt werden kann.	
Standard-Einstellung	OFF		
Dynamische Spezifikation	ja		
Spezifikation in Session	ja	Gültige Statements:	SET GLOBALS
		Gültiges Kommando:	GLOBALS
Programmierschnittstelle (API)	USR1005N	Siehe SYSEXT - <i>Natural Application Programming Interfaces</i> in der <i>Utilities</i> -Dokumentation.	



Anmerkung: Wenn ein Natural-Statement ausgeführt wird, das bewirkt, dass Adabas-Datensätze gelesen werden und eine Aktualisierungs/Lösch-Operation folgen könnte, fordert Natural Adabas an, diese Datensätze in den Hold-Status zu stellen. Weitere Informationen zur Hold-Verarbeitung siehe die Adabas *Command Reference*-Dokumentation.

Unter Natural Security: Die Einstellung dieses Parameters kann durch die Session Parameters-Option des *Library Profile* überschrieben werden.

274

WORK - Work-File Assignments

▪ WORK Parameter Syntax	688
▪ NETWORK Macro Syntax	689
▪ Keyword Subparameters for All Environments	690
▪ Keyword Subparameters for AM=STD in All Environments	694
▪ Keyword Subparameters for AM=STD in z/OS Environments	695
▪ Keyword Subparameters for AM=STD in z/VSE Environments	697
▪ Keyword Subparameters for AM=STD in BS2000/OSD Environments	698
▪ Keyword Subparameters for AM=CICS	699
▪ Keyword Subparameters for AM=COMP	700
▪ Keyword Subparameters for AM=SMARTS	700

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.

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	Complete 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.



Anmerkung: 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	<p>For usage of DEST under CMS, refer to <i>Natural under VM/CMS</i> in the <i>Operations</i> documentation).</p>
AM=COMP	<p>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.</p>

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 <i>x</i> within single quotes)
	x'xx'	(one hex character <i>xx</i>)
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 <i>x</i> within single quotes)
	x'xx'	(one hex character <i>xx</i>)
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.



Anmerkung: 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 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	

277

XML - Activate PARSE XML and REQUEST DOCUMENT

Statements

▪ XML Parameter Syntax	708
▪ NTXML Macro Syntax	709
▪ Keyword Subparameters	709
▪ Example of NTXML Macro	711
▪ Example of XML Parameter	711

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.



Anmerkung: 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 PARSE XML and REQUEST DOCUMENT 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 <i>Keyword Subparameters</i> 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 NATPARM, use the macro <code>NTXML</code> .
Specification within session	no	

XML Parameter Syntax

The XML parameter is specified as follows:

$$\text{XML} = \left(\begin{Bmatrix} \text{ON} \\ \text{OFF} \end{Bmatrix} , \text{keyword_subparameter1} = \text{value}, \text{keyword_subparameter2} = \text{value}, \dots \right)$$

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,                             *
            RDSPOINT=port-number,                 *
            RDNOP=domain-name                     *

```



Anmerkung: The keyword subparameters RDPPORT and RDPS are currently for z/OS only.

Keyword Subparameters

[RDOC](#) | [PARSE](#) | [RDCP](#) | [RDP](#) | [RDPPORT](#) | [RDPS](#) | [RDSPOINT](#) | [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.



Anmerkung: 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.



Anmerkung: 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,RDPPOINT=8080,RDPS=SSLPROXY.MYCOMPANY.COM,RDSPOINT=443,RDNOP=*.MYCOMPANY.COM,RDOC=ON,PARSE=ON
```



Anmerkung: The keyword subparameters RDPPOINT and RDPS are currently for z/OS only.

Example of XML Parameter

```
XML=(ON,RDP='HTTPPROXY.MYCOMPANY.COM',RDPPOINT=8080,RDPS='SSLPROXY.MYCOMPANY.COM',RDSPOINT=443,RDNOP='*.MYCOMPANY.COM',RDOC=ON,PARSE=ON)
```



Anmerkung: The keyword subparameters RDPPOINT and RDPS are currently for z/OS only.

278

XREF - Creation of XRef Data for Natural

- Extended XRef Data Generation (For Internal Use Only) 715

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

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)



Wichtig: 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.



Vorsicht: 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	



Anmerkungen:

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	By setting the parameter to a value between "1-99", you determine when the 100-year range begins in the past. The YSLW setting is subtracted from the current year to determine the first year of the window range. Example: If the current year is 2002 and you specify YSLW=40, the sliding window will cover the years 1962 to 2061. A 2-digit year setting <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> ".
	Fixed Window	1582 - 2600	By setting the parameter to a value between "1582-2600", you determine the first year of a 100-year range. The upper boundary of the 100-year range is evaluated by adding "99" to the value specified. Example: 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> ".

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

Teilung durch Null

Mit diesem Natural Profil- und Session-Parameter bestimmen Sie, was im Falle einer Division durch Null (0) geschehen soll.

In einer Natural-Session kann der Profilparameter ZD durch den Session-Parameter ZD überschrieben werden.

Mögliche Werte	ON	Natural gibt eine Fehlermeldung aus, falls versucht wird, eine Zahl durch Null (0) zu teilen.	
	OFF	Natural gibt bei einer Division durch Null als Ergebnis eine Null aus.	
Standard-Einstellung	ON		
Dynamische Spezifikation	ja		
Spezifikation in Session	ja	Gültige Statements:	SET GLOBALS
		Gültiges Kommando:	GLOBALS
Programmierschnittstelle (API)	USR1005N	Siehe SYSEXT - <i>Natural Application Programming Interfaces</i> in der <i>Utilities</i> -Dokumentation.	

Unter Natural Security: Die Einstellung dieses Parameters kann durch die *Session Parameters*-Option des Library-Profiles überschrieben werden.

283

ZP - Zero Printing

Anzeige von Nullwerten

Dieser Natural Profil- und Session-Parameter gibt an, wie ein Feld mit lauter Nullen ausgegeben werden soll, d.h. er wird benutzt, um die Anzeige eines numerischen Feldes (Format N, P, I oder F) oder eines Zeitfeldes (Format T), dessen Wert aus lauter Nullen besteht, zu unterdrücken.

In einer Natural-Session kann der Profilparameter ZP durch den Session-Parameter ZP überschrieben werden.

Mögliche Werte	ON	Bei einem Feldwert, der aus lauter Nullen besteht, wird bei numerischen Feldern eine Null rechtsbündig bzw. bei Zeitfeldern der ganze Feldwert angezeigt.	
	OFF	Ein Feldwert, der aus lauter Nullen besteht, wird nicht angezeigt.	
Standard-Einstellung	ON		
Dynamische Spezifikation	ja		
Spezifikation in Session	ja	Gültige Statements:	DISPLAY FORMAT INPUT PRINT REINPUT SET GLOBALS WRITE
		Gültiges Kommando:	GLOBALS
Programmierschnittstelle (API)	USR1005N	Siehe SYSEXT - <i>Natural Application Programming Interfaces</i> in der <i>Utilities</i> -Dokumentation.	

Siehe auch *Parameter zur Beeinflussung der Ausgabe von Feldern im Leitfaden zur Programmierung*.

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 ZSIZE=0 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	

Stichwortverzeichnis

P

parameter
 overview, 1
profile parameter
 overview, 1

S

session parameter
 overview, 1

