

# Natural for Mainframes

## Parameter-Referenz

Version 9.2.1

November 2024

Dieses Dokument gilt für Natural for Mainframes ab Version 9.2.1.

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

Copyright © 1979-2024 Software AG, Darmstadt, Deutschland und/oder Software AG USA, Inc., Reston, VA, USA, und/oder ihre Tochtergesellschaften und/oder ihre Lizenzgeber.

Der Name Software AG und die Namen der Software AG Produkte sind Marken der Software AG und/oder Software AG USA Inc., einer ihrer Tochtergesellschaften oder ihrer Lizenzgeber. Namen anderer Gesellschaften oder Produkte können Marken ihrer jeweiligen Schutzrechtsinhaber sein.

Nähere Informationen zu den Patenten und Marken der Software AG und ihrer Tochtergesellschaften befinden sich unter <http://documentation.softwareag.com/legal/>.

Diese Software kann Teile von Software-Produkten Dritter enthalten. Urheberrechtshinweise, Lizenzbestimmungen sowie zusätzliche Rechte und Einschränkungen dieser Drittprodukte können dem Abschnitt "License Texts, Copyright Notices and Disclaimers of Third Party Products" entnommen werden. Diese Dokumente enthalten den von den betreffenden Lizenzgebern oder den Lizenzen wörtlich vorgegebenen Wortlaut und werden daher in der jeweiligen Ursprungssprache wiedergegeben. Für einzelne, spezifische Lizenzbeschränkungen von Drittprodukten siehe PART E der Legal Notices, abrufbar unter dem Abschnitt "License Terms and Conditions for Use of Software AG Products / Copyrights and Trademark Notices of Software AG Products". Diese Dokumente sind Teil der Produktdokumentation, die unter <http://softwareag.com/licenses> oder im Verzeichnis der lizenzierten Produkte zu finden ist.

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

**Dokument-ID: NATMF-NNATPARMS-921-20241107DE**

# Inhaltsverzeichnis

Vorwort .....	xvii
1 Über diese Dokumentation .....	1
Dokumentationskonventionen .....	2
Online-Informationen und Support .....	2
Datenschutz .....	3
2 Profilparameter - Einführung .....	5
3 Session-Parameter - Einführung .....	7
Benutzung von Session-Parametern .....	8
Setzen von Session-Parametern .....	8
Verarbeitung von Session-Parametern .....	10
4 Profilparameter sortiert nach Kategorien .....	13
Batch-Modus .....	14
Zeichenzuweisungen .....	14
Kompilierungsoptionen .....	15
Datenbankverwaltung .....	15
Datums- und Uhrzeiteinstellungen .....	16
Debugging und Tracing .....	16
Externe Subprogramme .....	17
Maximalwerte (Limits) .....	17
Überwachung .....	18
Natural-Systemdateien .....	18
Natural und andere Produkte der Software AG .....	19
Leistungsoptimierung .....	20
Ausgabe-Reports und Arbeitsdateien .....	20
Remote Procedure Call-Einstellungen (RPC) .....	21
Steuerung der Natural-Session .....	21
Initialisierung und Beendigung einer Natural-Session .....	21
Source-Verwaltung .....	22
Speicherverwaltung .....	22
Terminal-Kommunikation .....	23
Zeichenumsetzungstabellen .....	24
Unicode- und Codepage-Unterstützung .....	24
Nutzungs- und Konfigurationseinstellungen für Profilparameter .....	25
Unterstützung von Statements für den Internet- und XML-Zugriff .....	25
TP-Monitor-Schnittstellen .....	25
5 ABLOG – Programmausführungsfehler protokollieren .....	27
6 AD - Attribut-Definition .....	29
AD-Parameter-Syntax .....	30
Feldanzeige .....	31
Ausrichtung der Feldwerte .....	32
Feldeingabe/ausgabe-Charakteristika .....	32
Auslegung alphanumerischer Felder .....	34
Eingabezwang .....	34

Mindestlänge der Eingabewerte .....	34
Groß-/Kleinschreibung .....	35
Füllzeichen .....	35
7 ADAACBX – Verwendung des erweiterten Adabas Control Block .....	37
8 ADAMODE - Adabas-Call-Schnittstellenmodus .....	39
9 ADANAME - Name der Adabas-Link-Routine .....	43
10 ADAPRM - Adabas Review-Unterstützung .....	45
11 ADASBV - Adabas Security durch Einstellung .....	47
12 AL - Alphanumerische Länge der Ausgabe .....	49
13 ASIZE - Entire System Server-Hilfspuffer .....	51
14 ASPSIZE (für interne Verwendung) .....	53
15 ATTN - Unterstützung für Programm-Unterbrechungstaste .....	55
16 AUTO - Automatische Anmeldung .....	57
17 BP82 - Buffer Pool-Platzhalter wenn Objekt nicht gefunden .....	59
18 BPC64 - Buffer Pool Cache-Speichertyp .....	61
19 BPCSIZE - Cache-Größe für den Natural Buffer Pool .....	63
20 BPI - Buffer Pool-Initialisierung .....	65
BPI-Parameter-Syntax .....	66
NTBPI-Makro-Syntax .....	67
Schlüsselwort-Subparameter .....	67
Beispiele für BPI-Parameter .....	72
Beispiele für NTBPI-Makros .....	73
21 BPLIST - Name der Preload-Liste für den Natural Buffer Pool .....	75
22 BPMETH - Suchalgorithmus zur Zuweisung von Speicherplatz im Buffer Pool .....	77
23 BPNAME - Name des globalen Natural Buffer Pool .....	79
24 BPPROP - Global Buffer Pool-Weiterleitung .....	81
25 BPSFI - Objektsuche zuerst im Buffer Pool .....	83
26 BPSIZE - Größe des lokalen Natural Buffer Pool .....	85
27 BPTEXT - Größe der Textsegmente im Natural Buffer Pool .....	87
28 BSIZE - Größe des EntireX Broker Buffer .....	89
29 BX - Feldumrahmung (Box-Definition) .....	91
30 CANCEL - Abbruch der Session mit Dump .....	93
31 CC - Fehlerverarbeitung im Batch-Modus .....	95
32 CCHAR - Ausgabe-Steuerzeichen zulassen .....	97
33 CCTAB - Drucker-ESC-Folge-Definition .....	99
CCTAB-Parameter-Syntax .....	100
NTCCTAB-Makro-Syntax .....	102
Syntax der Zeichenketten bei OPN, CLS, CODE, CS, CSS bzw. CSE .....	102
Proportionale Schriftarten .....	103
Beispiele für CCTAB-Parameter .....	103
Beispiele für NTCCTAB-Makros .....	103
34 CD - Farbdefinition .....	105
35 CDYNAM - Dynamisches Laden von Nicht-Natural-Programmen .....	107
36 CF - Steuerzeichen für Terminalkommandos .....	109
37 CFICU - Unicode- und Codepage-Unterstützung .....	111

CFICU-Parameter-Syntax .....	112
NTCFICU-Makro-Syntax .....	113
Schlüsselwort-Subparameter .....	113
Beispiel für CFICU-Parameter .....	118
Beispiel für NTCFICU-Makro .....	118
38 CFWSIZE (für interne Verwendung) .....	119
39 CICSP - Umgebungsparameter für Natural CICS-Interface .....	121
NTCICSP-Makro-Syntax .....	122
Schlüsselwort-Subparameter .....	123
Beispiel für ein NTCICSP-Makro .....	142
40 CLEAR - Verarbeitung der CLEAR-Taste im NEXT-Modus .....	143
41 CM - Kommandomodus .....	145
42 CMPO - Kompilierungsoptionen .....	147
CMPO-Parameter-Syntax .....	148
NTCMPO-Macro-Syntax .....	149
Schlüsselwort-Subparameter .....	150
Beispiel für CMPO-Parameter .....	150
Beispiel für NTCMPO-Makro .....	150
43 CMPR - Standard-Algorithmus zur Komprimierungsoptimierung .....	151
44 COMP - Parameter für Natural Com-plete/SMARTS Interface .....	153
COMP Parameter Syntax .....	154
NTCOMP-Makro-Syntax .....	154
Schlüsselwort-Subparameter .....	155
Beispiel für NTCOMP-Makro .....	159
Beispiel für COMP-Parameter .....	159
45 CP - Name der Standard-Codepage .....	161
46 CPCVERR - Codepage-Umsetzungsfehler .....	165
47 CPOBJIN - Codepage der Batch-Eingabedatei .....	167
48 CPPRINT - Codepage der Batch-Ausgabedatei .....	169
49 CPSYNIN - Codepage der Batch-Eingabedatei für Kommandos .....	171
50 CSIZE - Größe des Con-nect/Con-form-Pufferbereichs .....	173
51 CSTATIC - Statisch verlinkte Module .....	175
CSTATIC-Parameter-Syntax .....	176
NTCSTAT-Macro-Syntax .....	177
Beispiel für CSTATIC-Parameter .....	177
Beispiele für NTCSTAT-Makros .....	177
52 CV - Kontrollvariable .....	179
53 CVMIN - Status 'Modified' der Kontrollvariable bei Eingabe .....	181
54 DATSIZE - Minimale Größe des lokalen Daten-Puffers .....	183
55 DB - Datenbanktypen und -optionen .....	187
DB-Parameter-Syntax .....	188
NTDB-Makro-Syntax .....	189
Mögliche Datenbanktypen .....	190
Mögliche Datenbank-Optionen .....	191
Beispiele für DB-Parameter .....	191

Beispiel für NTDB-Makro .....	192
56 DB2 - Parameter für SQL-Datenbank-Verwaltungssystem-Schnittstellen .....	193
DB2-Parameter-Syntax .....	194
NTDB2-Makro-Syntax .....	194
Schlüsselwort-Subparameter .....	195
Beispiel für DB2-Parameter .....	208
Beispiel für NTDB2-Makro .....	208
57 DB2SIZE - Natural-Pufferbereich für Db2 .....	209
58 DBCLOSE - Datenbank-Schließung am Ende einer Session .....	211
59 DBGAT - Debug Attach Server für NaturalONE .....	213
DBGAT-Parameter-Syntax .....	214
NTDBGAT-Makro-Syntax .....	215
Beispiel für DBGAT-Parameter .....	215
Beispiel für NTDBGAT-Makro .....	215
60 DBGERR - Automatischer Debugger-Start bei Laufzeitfehler .....	217
61 DBID - Standard-Datenbankkennung für Natural-Systemdateien .....	219
62 DBOPEN - Datenbank-Öffnung ohne ETID .....	221
63 DBROLL - Datenbankaufrufe vor Session-Unterbrechung .....	223
64 DBUPD - Datenbankänderungen .....	225
65 DC - Dezimalstellenzeichen .....	227
66 DD - Tagesdifferenz einstellen .....	229
67 DELETE - Löschung dynamisch geladener Programme .....	231
68 DF - Datumsformat .....	233
69 DFOUT - Datumsformat für Ausgabe .....	235
70 DFSTACK - Datumsformat für Natural-Stack .....	237
71 DFTITLE - Datumsformat in Standard-Seitenüberschrift .....	239
72 DL - Ausgabelänge .....	241
73 DLISIZE - Pufferbereichsgröße für DL/I .....	243
74 DO - Anzeige-Reihenfolge von Ausgabedaten .....	245
75 DS - Größe der Pufferspeicher festlegen .....	247
DS-Parameter-Syntax .....	248
NTDS-Makro-Syntax .....	249
Tabelle der Puffer-Größen .....	249
Beispiel für DS-Parameter .....	252
Beispiele für NTDS-Makros .....	252
76 DSC - Datenstromkomprimierung bei Terminals des Typs 3270 .....	253
77 DSIZE - Größe des DBLOG-Puffers .....	255
78 DTFORM - Datumsformat .....	257
79 DU - Dump-Erstellung .....	259
80 DUE - Dump-Erstellung, fehlerspezifisch .....	261
81 DY - Dynamische Attribute .....	263
DY-Parameter-Syntax .....	265
Beispiele .....	266
82 DYNPARM - Verwendung dynamischer Parameter .....	269
DYNPARM-Parameter-Syntax .....	270

NTDYNP-Makro-Syntax .....	271
Beispiel .....	271
83 ECHO - Ausdrucken von Batch-Eingabedaten ein-/ausschalten .....	273
84 EDBP - Software AG Editor Buffer Pool-Definitionen .....	275
EDBP-Parameter-Syntax .....	276
NTEDBP-Makro-Syntax .....	277
Schlüsselwort-Subparameter .....	277
Beispiel für EDBP-Parameter .....	283
Beispiel für NTEDBP-Makro .....	283
85 EDPSIZE - Größe des Software AG Editor-Hilfs-Buffer-Pool .....	285
86 EJ - Seitenvorschub .....	287
87 EM - Editiermaske .....	289
EM-Parameter-Syntax .....	290
Beispiele .....	291
Leerzeichen in Editiermasken .....	291
Standard-Editiermasken .....	292
Editiermasken für numerische Felder .....	292
Editiermasken für alphanumerische Felder .....	295
Editiermasken für binäre Felder – Format B .....	297
Hexadezimale Editiermasken .....	297
Editiermasken für Datums- und Zeitfelder (Formate D und T) .....	299
Editiermasken für logische Felder (Format L) .....	303
88 EMFM - Editiermasken-Frei-Modus .....	305
89 EMU - Unicode-Editiermaske .....	307
90 ENDBT - BACKOUT TRANSACTION am Session-Ende .....	309
91 ENDMMSG - Session-Ende-Meldung anzeigen .....	311
92 ES - Leerzeilenunterdrückung .....	313
93 ESCAPE - Terminalkommandos %% und % ignorieren .....	315
94 ESIZE - Größe des User Buffer-Erweiterungsbereichs .....	317
95 ET - Ausführung von END TRANSACTION/BACKOUT TRANSACTION-Statements .....	319
96 ETA - Fehlertransaktionsprogramm .....	321
97 ETDB - Datenbank für Transaktionsdaten .....	323
98 ETEOP - END TRANSACTION-Statement bei Programmende .....	325
99 ETID - Adabas-Benutzerkennung .....	327
100 ETIO - END TRANSACTION-Statement bei Terminal-Ein-/Ausgabe .....	329
101 ETRACE - Externe Trace-Funktion aktivieren/deaktivieren .....	331
102 ETSYNC - Syncpoint bei END TRANSACTION/BACKOUT TRANSACTION .....	333
103 EXCSIZE - Puffergröße für Natural Expert C Interface .....	335
104 EXRSIZE - Puffergröße für Natural Expert-Regeltabellen .....	337
105 FAMSTD - Überschreiben der Arbeitsdatei-Zugriffsmethoden-Zuweisungen .....	339
106 FC - Füllzeichen für INPUT-Statement .....	341
107 FC - Füllzeichen für DISPLAY-Statement .....	343
108 FCDP - Füllzeichen für dynamisch geschützte Felder .....	345
109 FDIC - Predict-Systemdatei .....	347

110 FL - Gleitkomma-Mantissenlänge .....	349
111 FNAT - Natural-Systemdatei für Systemprogramme .....	351
112 FNR - Standard-Dateinummer der Natural-Systemdateien .....	355
113 FPROF - Natural-Systemdatei für Parameterprofile .....	357
114 FREEGDA - GDA im Utility-Modus freigeben .....	361
115 FREG - Natural-Registry-Systemdatei .....	363
116 FS - Format-Spezifikation für Benutzervariablen .....	367
117 FSEC - Natural Security-Systemdatei .....	369
118 FSIZE (für interne Verwendung) .....	373
119 FSPOOL - Spool-Datei für Natural Advanced Facilities .....	375
120 FUSER - Natural-Systemdatei für Benutzerprogramme .....	377
121 GC - Füllzeichen für Gruppenüberschriften .....	381
122 HC - Überschriften-Zentrierung .....	383
123 HCAM - Hardcopy-Zugriffsmethode .....	385
124 HCDEST - Ziel für Hardcopy-Ausgabe .....	387
125 HD - Festlegung der Standard-Spaltenüberschrift .....	389
126 HE - Helproutine .....	391
HE-Parameter-Syntax .....	392
Ausführung von Helproutinen .....	394
Beispiele .....	395
127 HI - Hilfe-Aufrufzeichen .....	397
128 HW - Überschriftenbreite .....	399
129 IA - Input-Zuweisungszeichen .....	401
130 IC - Einfügungszeichen .....	405
131 ICU - Unicode-Einfügungszeichen .....	407
132 ID - Input-Begrenzungszeichen .....	409
133 IKEY - Verarbeitung von PA- und PF-Tasten .....	411
134 IM - Input-Modus .....	413
135 IMMSG - Nachricht bei Session-Initialisierungsfehler .....	415
136 IMSP - Allgemeine Parameter für das Natural IMS TM Interface .....	417
NTIMSP-Makro-Syntax .....	418
Schlüsselwort-Subparameter .....	418
Beispiel für NTIMSP-Makro .....	420
137 IMSPE - Umgebungsparameter für das Natural IMS TM Interface .....	421
NTIMSPE-Makro-Syntax .....	422
Schlüsselwort-Subparameter .....	423
Beispiel für NTIMSPE-Makro .....	434
138 IMSPT - Transaktionsdefinitionen für Natural IMS TM Interface .....	435
NTIMSPT-Makro-Syntax .....	436
Schlüsselwort-Subparameter .....	436
Beispiel für NTIMSPT-Makro .....	440
139 INTENS - Drucken von hervorgehobenen Feldern .....	441
140 IP - Eingabeaufforderungstext .....	443
141 IS - Unterdrückung identischer Werte .....	445
142 ISIZE - Größe des Puffers für Initialisierung .....	447

143	ITERM - Session-Beendigung bei Initialisierungsfehler .....	449
144	ITRACE - Interne Trace-Funktion aktivieren/deaktivieren .....	451
145	KD - Anzeige der PF-Tastennamen .....	453
146	KEY - PA-, PF- und CLEAR-Tastenbelegungen zuweisen .....	455
147	LC - Umsetzung von Klein- in Großbuchstaben .....	457
148	LC - Vorangestellte Zeichen .....	459
149	LCU - Vorangestellte Unicode-Zeichen .....	461
150	LE - Reaktion auf Limit-Überschreitung bei Verarbeitungsschleifen .....	463
151	LFILE - Logische Systemdatei-Definition .....	465
	LFILE-Parameter-Syntax .....	466
	NTLFILE-Makro-Syntax .....	467
	Beispiel für LFILE-Parameter .....	467
	Beispiele für NTLFILE-Makros .....	468
152	LIBNAM - Name der externen Programm-Load Library .....	469
153	LOG (für interne Verwendung) .....	471
154	LS - Zeilenlänge .....	473
	Profilparameter LS .....	474
	Session-Parameter LS .....	474
	Angabe in Statements .....	475
155	LT - Limit für Verarbeitungsschleifen .....	477
156	MADIO - Maximale Anzahl der DBMS-Aufrufe zwischen Bildschirm-Ein-/Ausgaben .....	479
157	MAINPR - Standard-Ausgabe-Report-Nummer .....	481
158	MAXCL - Maximale Anzahl an Programmaufrufen .....	483
159	MAXROLL - Anzahl der CMROLL-Aufrufe vor vorübergehender Session-Unterbrechung .....	485
160	MAXYEAR - Maximale Jahresangabe bei Datums-/Zeitwerten .....	487
161	MC - Anzahl multipler Feldwerte .....	489
162	MENU - Menü-Modus .....	491
163	ML - Position der Meldungszeile .....	493
164	MONSIZE - Größe des SYSTP Monitor-Puffers .....	495
165	MP - Maximale Seitenzahl eines Reports .....	497
166	MS - Manuelle Cursor-Positionierung .....	499
167	MSGSF - Anzeigeformat von Systemfehlermeldungen .....	501
168	MT - Maximale CPU-Zeit .....	503
169	NAFSIZE - Puffergröße für Natural Advanced Facilities .....	505
170	NAFUPF - Benutzerprofil für Natural Advanced Facilities .....	507
171	NC - Verwendung von Natural-Systemkommandos .....	509
172	NISN (für interne Verwendung) .....	511
173	NL - Numerische Länge der Ausgabe .....	513
174	NUCNAME - Name des gemeinsam genutzen Nukleus .....	515
175	OBJIN - Benutzung von CMOBJIN als Natural-Eingabedatei .....	517
176	OPF - Überschreiben geschützter Felder durch Helprountinen .....	519
177	OPRB - Datenbank-Open/Close-Befehlsverarbeitung .....	521
	OPRB-Parameter-Syntax .....	522

Dynamisch angegebener OPRB-Parameter bei Natural Security .....	523
OPRB für VSAM .....	524
OPRB für Adabas .....	524
NTOPRB-Makro-Syntax .....	526
Beispiele für NTOPRB-Makros .....	526
178 OPT - Steuerung des Natural Optimizer Compiler .....	527
OPT-Parameter-Syntax .....	528
NTOPT-Makro-Syntax .....	528
179 OSP - Parameter für z/OS Batch .....	529
OSP-Parameter-Syntax .....	530
NTOSP-Makro-Syntax .....	530
Schlüsselwort-Subparameter .....	531
Beispiel für OSP-Parameter .....	534
Beispiel für NTOSP-Makro .....	534
180 OUTDEST - Ausgabebziel für asynchrone Verarbeitung .....	535
181 OVSIZE - Speicher-Thread-Überlaufgröße .....	537
182 PARM - Alternatives Parametermodul .....	539
183 PC - Personal-Computer-Zugriffsmethode .....	541
184 PC - Anzahl der Periodengruppen-Ausprägungen .....	543
185 PCNTRL - Druckersteuerzeichen .....	545
186 PD - Seiten-Limit für NATPAGE .....	547
187 PDPSIZE - Profiler-Daten-Pool-Größe .....	549
188 PECK - PCHECK/ECHECK-Fehlerverarbeitung .....	551
189 PGP - Eigenschaften für externe Programme .....	553
PGP-Parameter-Syntax .....	554
NTPGP-Makro-Syntax .....	556
Beispiele für PGP-Parameter .....	557
Beispiele für NTPGP-Makros .....	557
190 PLOG - Protokollieren von dynamischen Parametern .....	559
191 PM - Druck-/Anzeige-Modus .....	561
Profilparameter PM .....	562
Session-Parameter PM .....	563
192 POS22 - Version 2.2 Algorithmus für Systemfunktion POS .....	565
193 PRINT - Druckdatei-Zuweisungen .....	567
PRINT-Parameter-Syntax .....	568
NTPRINT-Makro-Syntax .....	569
Schlüsselwort-Subparameter für alle Umgebungen .....	570
Schlüsselwort-Subparameter für AM=STD in allen Umgebungen .....	576
Schlüsselwort-Subparameter für AM=STD in z/OS-Umgebungen .....	579
Schlüsselwort-Subparameter für AM=CICS .....	581
Schlüsselwort-Subparameter für AM=COMP (Com-plete) .....	583
Schlüsselwort-Subparameter für AM=SMARTS (Com-plete) .....	583
Schlüsselwort-Subparameter für AM=IMS .....	584
Schlüsselwort-Subparameter für das DEFINE PRINTER-Statement .....	584
194 PROFILE - Dynamisches Parameterprofil anwenden .....	587

PROFILE-Parameter-Syntax .....	589
195 PROGRAM - Kontrolle durch Nicht-Natural-Programm nach Session-Beendigung .....	593
196 PS - Länge einer Reportseite .....	597
197 PSEUDO - Pseudo-konversationeller Modus unter CICS .....	599
198 RCA - Adressen von statischen Nicht-Natural-Programmen auflösen .....	601
199 RCALIAS - Externe Namen für Nicht-Natural-Programme .....	603
RCALIAS-Parameter-Syntax .....	604
NTALIAS-Makro-Syntax .....	605
Beispiel für RCALIAS-Parameter .....	605
Beispiele für NTALIAS-Makro .....	605
200 RCFIND - Maßnahme bei Adabas Response Code 113 beim FIND-Statement .....	607
201 RCGET - Maßnahme bei Adabas Response Code 113 beim GET-Statement .....	609
202 RDACT - (für interne Verwendung) .....	611
203 RDC - Konfiguration des Natural Data Collector .....	613
RDC-Parameter-Syntax .....	614
NTRDC-Makro-Syntax .....	615
Schlüsselwort-Subparameter .....	615
Beispiel für RDC-Parameter .....	618
Beispiel für NTRDC-Makro .....	618
204 RDCEXIT - User Exits für den Natural Data Collector .....	619
205 RDCSIZE - Puffergröße für den Natural Data Collector .....	621
206 RDNODE (für interne Verwendung) .....	623
207 RDPORT (für interne Verwendung) .....	625
208 RECAT - Dynamisches Rekatalogisieren .....	627
209 REINP - Interner REINPUT bei ungültigen Daten .....	629
210 RELO - Speicher-Thread-Verschiebung .....	631
211 RFILE - Datei für Aufzeichnung per Recording-Funktion .....	633
212 RI - Freigabe von Adabas ISNs .....	635
213 RJESIZE - Anfangsgröße des NATRJE-Puffers .....	637
214 RM - Rückübertragung von geänderten Feldern .....	639
215 RNCONST - Zeilennummern in Konstanten umnummerieren .....	641
216 ROSY - Schreibgeschützter Zugriff auf Systemdateien .....	643
217 RPC - Remote Procedure Call-Einstellungen .....	645
RPC-Parameter-Syntax .....	646
NTRPC-Macro-Syntax .....	646
Schlüsselwort-Subparameter für Client und Server .....	647
Schlüsselwort-Subparameter nur für Client .....	651
Schlüsselwort-Subparameter nur für Server .....	656
Beispiel für RPC-Parameter .....	665
Beispiel für NTRPC-Makro .....	665
218 RUNSIZE - Größe des Laufzeit-Puffers .....	667
219 SA - Terminal-Warnton .....	669
220 SB - Auswahlfeld .....	671
Anmerkungen zur Syntax .....	672

Anmerkungen zur Laufzeitumgebung .....	673
Funktionen .....	674
Beschränkungen .....	676
221 SCTAB - Scanner-Zeichentabelle .....	677
SCTAB-Parameter-Syntax .....	678
NTSCTAB-Makro-Syntax .....	679
Beispiel für SCTAB-Parameter .....	680
Beispiel für NTSCTAB-Makros .....	680
222 SELUNIT - Aktivierung ausgewählter Natural-Merkmale .....	681
Zur Verfügung stehende wählbare Natural-Merkmale ("Selectable Units") .....	682
SELUNIT-Parameter-Syntax .....	683
Beispiel für SELUNIT-Parameter .....	683
223 SENDER - Bestimmungsort für Bildschirmausgabe bei asynchroner Verarbeitung .....	685
224 SF - Spaltenabstand .....	687
225 SG - Vorzeichen-Stelle .....	689
226 SHAPED - Steuerung der Zeichenumformung .....	691
227 SKEY - Speicherschutzschlüssel für Programmausführung .....	693
228 SL - Sourcecode-Zeilenlänge .....	695
229 SLOCK - Source-Sperrung .....	697
230 SM - Programmierung im Structured Mode .....	699
231 SORT - Steuerung des Sortierprogramms .....	701
SORT-Parameter-Syntax .....	702
NTSORT-Makro-Syntax .....	702
Schlüsselwort-Subparameter .....	703
Beispiel für SORT-Parameter .....	706
Beispiele für NTSORT-Macros .....	707
232 SOSI - Shift-Out/Shift-In-Codes für Doppel-Byte-Zeichensätze .....	709
SOSI-Parameter-Syntax .....	710
Positionsgebundene Subparameter .....	710
Konvertierung von logischen Shift-Out/Shift-In Zeichen .....	711
Automatische Anpassung von Umsetzungstabellen .....	711
Kompatibilität des SOSI-Profilparameters mit den veralteten SO- und SI-Profilparametern .....	712
SOSI-Parameter-Beispiel .....	712
233 SRETAIN - Source-Format beibehalten .....	713
234 SSIZE - Größe des Software AG Editor-Arbeitsbereichs .....	715
235 STACK - Daten/Kommandos auf den Natural Stack stellen .....	717
236 STACKD - Stack-Kommando-Abgrenzungszeichen .....	719
237 STEPLIB - Zusätzliche Steplib Library .....	721
238 SUBSID - Subsystemkennung unter z/OS .....	723
239 SYNERR - Übergabe von Syntaxfehlern .....	725
240 SYS - Satz dynamischer Profilparameter definieren und aktivieren .....	727
SYS-Parameter-Syntax .....	728
NTSYS-Makro-Syntax .....	729

Beispiele für NTSYS-Makros .....	730
241 SYSCIP - Adabas-Standard-Chiffrierschlüssel für Natural-Systemdateien .....	731
242 SYSPSW - Adabas-Standard-Passwort für Natural-Systemdateien .....	733
243 TAB - Standard-Ausgabezeichen-Umsetzung .....	735
TAB-Parameter-Syntax .....	736
NTTAB-Makro-Syntax .....	737
Beispiel für TAB-Parameter .....	737
Beispiel für NTTAB-Makro .....	737
244 TAB1 - Alternative Ausgabezeichen-Umsetzung .....	739
TAB1-Parameter-Syntax .....	740
NTTAB1-Makro-Syntax .....	741
Beispiel für TAB1-Parameter .....	741
Beispiel für NTTAB1-Makro .....	741
245 TAB2 - Alternative Eingabezeichen-Umsetzung .....	743
TAB2-Parameter-Syntax .....	744
NTTAB2-Makro-Syntax .....	745
Beispiel für TAB2-Parameter .....	745
Beispiel für NTTAB2-Makro .....	745
246 TABA1 - EBCDIC/ASCII-Zeichenumsetzung .....	747
TABA1-Parameter-Syntax .....	748
NTTABA1-Makro-Syntax .....	748
Beispiel für TABA1-Parameter .....	749
Beispiel für NTTABA1-Makro .....	749
247 TABA2 - ASCII/EBCDIC-Zeichenumsetzung .....	751
TABA2-Parameter-Syntax .....	752
NTTABA2-Makro-Syntax .....	752
Beispiel für NTTABA2-Makro .....	753
Beispiel für TABA2-Parameter .....	753
248 TABL - SYS-Library-Ausgabe-Zeichenumsetzung .....	755
TABL-Parameter-Syntax .....	756
NTTABL-Makro-Syntax .....	757
Beispiel für TABL-Parameter .....	757
Beispiel für NTTABL-Makro .....	757
249 TC - Nachgezogene Zeichen .....	759
250 TCU - Nachgezogene Zeichen (Unicode) .....	761
251 TD - Zeitdifferenz .....	763
252 TF - Umsetzung der Datenbankkennung/Dateinummer .....	765
TF-Parameter-Syntax .....	767
NTTF-Makro-Syntax .....	767
Beispiel für TF-Parameter .....	768
Beispiel für NTTF-Makro .....	768
253 THPINIT - Name des vorinitialisierten Speicher-Thread-Modells .....	769
254 THSEPCH - Tausender-Trennzeichen .....	771
255 THSIZE - Größe des Speicher-Thread .....	773
256 TMODEL - IBM 3270-Terminal-Modell .....	775

257 TPF (für interne Verwendung) .....	777
258 TQ - Translate Quotation Marks .....	779
259 TRACE - Komponenten für den Trace angeben .....	781
TRACE-Parameter-Syntax .....	782
NTTRACE-Makro-Syntax .....	783
Beispiel für TRACE-Parameter .....	783
Beispiel für NTTRACE-Makro .....	783
260 TS - Umsetzung von System-Library-Programmausgaben .....	785
261 TSIZE - Größe des Puffers für Adabas Text Retrieval .....	787
262 TSOP - Parameter für Natural TSO Interface .....	789
TSOP-Parameter-Syntax .....	790
NTTSOP-Makro-Syntax .....	790
Schlüsselwort-Subparameter .....	791
Beispiel für TSOP-Parameter .....	795
Beispiel für NTTSOP-Makro .....	795
263 TTYPE - Terminaltyp .....	797
264 UC - Unterstreichungszeichen .....	799
265 UCONMAX - Maximale Anzahl gleichzeitiger Sessions für einen Benutzer .....	801
266 UDB - Benutzer-Datenbankkennung .....	803
267 ULANG - Benutzersprache .....	805
268 UNIO (für interne Verwendung) .....	807
269 USER - Verwendung von Profilparameterketten und Modulen einschränken .....	809
USER-Parameter-Syntax .....	810
NTUSER-Makro-Syntax .....	811
Beispiel für USER-Parameter .....	812
Beispiel für NTUSER-Makro .....	812
270 USERBUF (für interne Verwendung) .....	813
271 UTAB1 - Umsetzung von Klein- in Großschreibung .....	815
UTAB1-Parameter-Syntax .....	816
NTUTAB1-Makro-Syntax .....	817
Beispiel für UTAB1-Parameter .....	817
Beispiel für NTUTAB1-Makro .....	817
272 UTAB2 - Umsetzung von Groß- in Kleinschreibung .....	819
UTAB2-Parameter-Syntax .....	820
NTUTAB2-Makro-Syntax .....	821
Beispiel für UTAB2-Parameter .....	821
Beispiel für NTUTAB2-Makro .....	821
273 VSAM - Parameter für Natural for VSAM .....	823
VSAM-Parameter-Syntax .....	824
NTVSAM-Makro-Syntax .....	824
NTVEXIT-Makro-Syntax .....	825
NTVLSR-Makro-Syntax .....	825
NTVTVSD-Makro-Syntax .....	825
Schlüsselwort-Subparameter .....	826
Beispiel für VSAM-Parameter .....	841

Beispiel für NTVSAM-Makro .....	841
274 VSIZE - Größe des Pufferbereichs für Natural for VSAM .....	843
275 WEBIO - Web I/O Interface Rendering .....	845
WEBIO-Parameter-Syntax .....	846
NTWEBIO-Makro-Syntax .....	846
Schlüsselwort-Subparameter .....	847
Beispiel für WEBIO-Parameter .....	848
Beispiel für NTWEBIO-Macro .....	848
276 WH - Warten auf Datensatz im Hold-Status .....	849
277 WORK - Arbeitsdatei-Zuweisungen .....	851
WORK-Parameter-Syntax .....	852
NTWORK-Makro-Syntax .....	853
Schlüsselwort-Subparameter für alle Umgebungen .....	854
Schlüsselwort-Subparameter für AM=STD in allen Umgebungen .....	859
Schlüsselwort-Subparameter für AM=STD in z/OS-Umgebungen .....	861
Schlüsselwort-Subparameter für AM=CICS .....	863
Schlüsselwort-Subparameter für AM=COMP .....	864
Schlüsselwort-Subparameter für AM=SMARTS .....	865
278 WPSIZE - Größe der Natural-Arbeitsspeicher .....	869
279 WSISIZE - Puffer für das Natural Workstation Interface .....	871
280 XML - Statements PARSE XML und REQUEST DOCUMENT aktivieren/deaktivieren .....	873
XML-Parameter-Syntax .....	874
NTXML-Makro-Syntax .....	875
Schlüsselwort-Subparameter .....	875
Beispiel für XML Parameter .....	881
Beispiel für NTXML Makro .....	881
281 XREF - Erzeugung von XRef-Daten für Natural .....	883
Möglichkeiten, den XREF-Parameter zu setzen .....	884
XRef-Datenerzeugung .....	885
Erweiterte XRef-Datenerzeugung (nur für interne Verwendung) .....	885
282 YD - Jahresdifferenz einstellen .....	887
283 YSLW - Gleitendes oder festes Jahr-Fenster .....	889
Beispiele für den YSLW-Parameter .....	891
284 ZD - Division durch Null .....	893
285 ZIIP - zIIP-Verarbeitung (nur z/OS) .....	895
ZIIP-Parameter-Syntax .....	896
NTZIIP-Makro-Syntax .....	897
Schlüsselwort-Subparameter .....	897
Beispiel für ZIIP-Parameter .....	903
Beispiel für NTZIIP-Macro .....	903
286 ZP - Anzeige von Nullwerten .....	905
287 ZSIZE - Größe des Entire DB-Pufferbereichs .....	907
Stichwortverzeichnis .....	909



---

# Vorwort

---

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



## Anmerkungen:

1. Grundsätzliche Hinweise zur Benutzung von Parametern finden Sie in der *Operations-Dokumentation*, siehe *Profilparameter anwenden*.
2. Mit dem Dienstprogramm (Utility) SYSPARM können Sie einen Satz Natural-Profilparameter als individuelles Parameterprofil anlegen und pflegen. Ein solches Parameterprofil können Sie für jede Natural-Sitzung verwenden.

<b>Profilparameter – Einführung</b>	Enthält allgemeine Informationen zur Benutzung von Profilparametern sowie eine Liste mit Referenzen auf verwandte Themen.
<b>Session-Parameter – Einführung</b>	Enthält allgemeine Informationen zur Benutzung und Verarbeitung von Session-Parametern.
<b>Profilparameter sortiert nach Kategorien</b>	Übersicht über die Natural-Profilparameter, gruppiert nach Verwendungszweck.
Parameter in alphabetischer Reihenfolge	Beschreibungen sämtlicher Profilparameter und Session-Parameter in alphabetischer Reihenfolge.



# 1 Über diese Dokumentation

---

- Dokumentationskonventionen ..... 2
- Online-Informationen und Support ..... 2
- Datenschutz ..... 3

## Dokumentationskonventionen

---

Konvention	Beschreibung
<b>Fettschrift</b>	>Kennzeichnet Elemente auf einem Bildschirm.
Nichtproportionale Schrift	Kennzeichnet Namen und Orte von Diensten im Format <i>Ordner.Unterordner.Dienst</i> , Programmierschnittstellen (APIs), Namen von Klassen, Methoden und Properties in Java.
<i>Kursivschrift</i>	Kennzeichnet:  Variablen, für die Sie situations- oder umgebungsspezifische Werte angeben müssen. Neue Begriffe, wenn sie erstmals im Text auftreten. Verweise auf andere Dokumentationsquellen.
Nichtproportionale Schrift	Kennzeichnet:  Text, den Sie eingeben müssen. Meldungen, die vom System angezeigt werden. Programmcode.
{ }	Zeigt eine Reihe von Auswahlmöglichkeiten an, von denen Sie eine auswählen müssen. Geben Sie nur die innerhalb der geschweiften Klammern vorhandenen Informationen ein. Geben Sie nicht die Klammersymbole { } ein.
	Trennt zwei sich gegenseitig ausschließende Auswahlmöglichkeiten in einer Syntaxzeile voneinander ab. Geben Sie eine der Auswahlmöglichkeiten ein. Geben Sie nicht das Symbol   ein.
[ ]	Zeigt eine oder mehrere Optionen an. Geben Sie nur die innerhalb der eckigen Klammern vorhandenen Informationen ein. Geben Sie nicht die Klammersymbole [ ] ein.
...	Zeigt an, dass Sie mehrere Auswahlmöglichkeiten desselben Typs eingeben können. Geben Sie nur die Informationen ein. Geben Sie nicht die drei Auslassungspunkte (...) ein.

## Online-Informationen und Support

---

### Produktdokumentation

Sie finden die Produktdokumentation auf unserer Dokumentationswebsite unter <https://documentation.softwareag.com>.

Zusätzlich können Sie auch über <https://www.softwareag.cloud> auf die Dokumentation für die Cloud-Produkte zugreifen. Navigieren Sie zum gewünschten Produkt und gehen Sie dann, je nach Produkt, zu „Developer Center“, „User Center“ oder „Documentation“.

## Produktschulungen

Sie finden hilfreiches Produktschulungsmaterial auf unserem Lernportal unter <https://knowledge.softwareag.com>.

## Tech Community

Auf der Website unserer Tech Community unter <https://techcommunity.softwareag.com> können Sie mit Experten der Software AG zusammenarbeiten. Von hier aus können Sie zum Beispiel:

- Unsere umfangreiche Wissensdatenbank durchsuchen.
- In unseren Diskussionsforen Fragen stellen und Antworten finden.
- Die neuesten Nachrichten und Ankündigungen der Software AG lesen.
- Unsere Communities erkunden.
- Unsere öffentlichen Repositories auf GitHub and Docker unter <https://github.com/softwareag> und <https://hub.docker.com/publishers/softwareag> besuchen und weitere Ressourcen der Software AG entdecken.

## Produktsupport

Support für die Produkte der Software AG steht lizenzierten Kunden über unser Empower-Portal unter <https://empower.softwareag.com> zur Verfügung. Für viele Dienstleistungen auf diesem Portal benötigen Sie ein Konto. Wenn Sie noch keines haben, dann können Sie es unter <https://empower.softwareag.com/register> beantragen. Sobald Sie ein Konto haben, können Sie zum Beispiel:

- Produkte, Aktualisierungen und Programmkorrekturen herunterladen.
- Das Knowledge Center nach technischen Informationen und Tipps durchsuchen.
- Frühwarnungen und kritische Alarmer abonnieren.
- Supportfälle öffnen und aktualisieren.
- Anfragen für neue Produktmerkmale einreichen.

## Datenschutz

---

Die Produkte der Software AG stellen Funktionen zur Verarbeitung von personenbezogenen Daten gemäß der Datenschutz-Grundverordnung (DSGVO) der Europäischen Union zur Verfügung. Gegebenenfalls sind in der betreffenden Systemverwaltungsdokumentation entsprechende Schritte dokumentiert.

---

## 2 Profilparameter - Einführung

---

Einen Natural-Profilparameter können Sie folgendermaßen angeben: Entweder dynamisch beim Start der Natural-Session oder statisch im Parameter-Makro `NTPRM` und in zusätzlichen Parameter-Makros, die im Natural-Parameter-Modul enthalten sind.

Natural-Profilparameter, zu denen in der zugehörigen Beschreibung kein Makro-Name oder keine Makro-Syntax angegeben ist, werden im Makro `NTPRM` definiert.

### **NTPRM-Makro-Syntax**

Das Makro `NTPRM` wird wie folgt angegeben:

```
NTPRM profile-parameter=value, ...
```

Dabei steht *profile-parameter* für eine Natural-Profilparametereinstellung, die für den angegebenen Profilparameter gültig ist.

Weitere Informationen zur Verwendung der Natural-Profilparameter finden Sie in folgenden Abschnitten in der *Natural Operations*-Dokumentation:

- Profile Parameter Usage
- Natural Parameter Hierarchy
- Assignment of Parameter Values

---

# 3 Session-Parameter - Einführung

---

- Benutzung von Session-Parametern ..... 8
- Setzen von Session-Parametern ..... 8
- Verarbeitung von Session-Parametern ..... 10

## 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-Natural-Parametermodul, 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 Profilparameter sortiert nach Kategorien

---

▪ Batch-Modus .....	14
▪ Zeichenzuweisungen .....	14
▪ Kompilierungsoptionen .....	15
▪ Datenbankverwaltung .....	15
▪ Datums- und Uhrzeiteinstellungen .....	16
▪ Debugging und Tracing .....	16
▪ Externe Subprogramme .....	17
▪ Maximalwerte (Limits) .....	17
▪ Überwachung .....	18
▪ Natural-Systemdateien .....	18
▪ Natural und andere Produkte der Software AG .....	19
▪ Leistungsoptimierung .....	20
▪ Ausgabe-Reports und Arbeitsdateien .....	20
▪ Remote Procedure Call-Einstellungen (RPC) .....	21
▪ Steuerung der Natural-Session .....	21
▪ Initialisierung und Beendigung einer Natural-Session .....	21
▪ Source-Verwaltung .....	22
▪ Speicherverwaltung .....	22
▪ Terminal-Kommunikation .....	23
▪ Zeichenumsetzungstabellen .....	24
▪ Unicode- und Codepage-Unterstützung .....	24
▪ Nutzungs- und Konfigurationseinstellungen für Profilparameter .....	25
▪ Unterstützung von Statements für den Internet- und XML-Zugriff .....	25
▪ TP-Monitor-Schnittstellen .....	25

## Batch-Modus

---

Folgende Profilparameter gelten bei Natural im Batch-Modus:

Parameter	Kurzbeschreibung
CC	Fehlerverarbeitung im Batch-Modus
CPOBJIN	Codepage der Batch-Eingabedatei
CPPRINT	Codepage der Batch-Ausgabedatei
CPSYNIN	Codepage der Batch-Eingabedatei für Kommandos
ECHO	Ausdrucken von Batch-Eingabedaten ein-/ausschalten
OBJIN	Benutzung von CMOBJIN als Natural-Eingabedatei
OSP	Parameter für z/OS Batch

## Zeichenzuweisungen

---

Folgende Profilparameter können Sie benutzen, um Standard-Zeichenzuweisungen zu ändern:

Parameter	Kurzbeschreibung
CVMIN	Status „Modified“ der Kontrollvariable bei Eingabe
CF	Steuerzeichen für Terminalkommandos
DC	Dezimalstellenzeichen
FC	Füllzeichen für INPUT-Statement
FCDP	Füllzeichen für dynamisch geschützte Felder
HI	Hilfe-Aufrufzeichen
IA	Input-Zuweisungszeichen
ID	Input-Begrenzungszeichen
SOSI	Shift-Out/Shift-In-Codes für Doppel-Byte-Zeichensätze
THSEPCH	Tausender-Trennzeichen

## Kompilierungsoptionen

Folgende Profilparameter können Sie benutzen, um die Kompilierungsoptionen zu steuern:

Parameter	Kurzbeschreibung
CMPO	Kompilierungsoptionen
FS	Format-Spezifikation für Benutzervariablen
SM	Programmierung im Structured Mode
XREF	Erzeugung von XRef-Daten für Natural

## Datenbankverwaltung

Folgende Profilparameter können Sie zum Konfigurieren von Datenbanken und zum Kontrollieren und Steuern des Zugriffs auf Datenbanken benutzen:

Parameter	Kurzbeschreibung
ADACBX	Verwendung des erweiterten Adabas Control Block
ADANAME	Name der Adabas-Link-Routine
ADAMODE	Adabas-Call-Schnittstellenmodus
ADAPRM	Adabas Review-Unterstützung
ADASBV	Adabas Security durch Einstellung
DB	Datenbanktypen und -optionen
DB2	Parameter für SQL-Datenbank-Verwaltungssystem-Schnittstellen
DBCLOSE	Datenbank-Schließung am Ende einer Session
DBOPEN	Datenbank-Öffnung ohne ETID
DBROLL	Datenbankaufrufe vor Session-Unterbrechung
DBUPD	Datenbank-Aktualisierung
ENDBT	BACKOUT TRANSACTION am Session-Ende
ET	Ausführung von END TRANSACTION/BACKOUT TRANSACTION-Statements
ETDB	Datenbank für Transaktionsdaten
ETEOP	END TRANSACTION-Statement bei Programmende
ETID	Adabas-Benutzerkennung
ETIO	END TRANSACTION-Statement bei Terminal-Ein-/Ausgabe
ETSYNC	Syncpoint bei END TRANSACTION/BACKOUT TRANSACTION
LFIELD	Logische Systemdatei-Definition

Parameter	Kurzbeschreibung
OPRB	Datenbank-Open/Close-Befehlsverarbeitung (bei Adabas- und VSAM-Datenbanken)
RCFIND	Maßnahme bei Adabas Response Code 113 beim FIND-Statement
RCGET	Maßnahme bei Adabas Response Code 113 beim GET-Statement
RI	Freigabe von Adabas ISNs
TF	Umsetzung der Datenbankkennung/Dateinummer
UDB	Benutzer-Datenbankkennung
WH	Warten auf Datensatz im Hold-Status

## Datums- und Uhrzeiteinstellungen

---

Folgende Profilparameter beeinflussen die Behandlung von Datums- und Uhrzeitwerten durch Natural sowie die Natural-intern verwendete Datums-/Uhrzeiteinstellung:

Parameter	Kurzbeschreibung
DD	Tagesdifferenz einstellen
DFOUT	Datumsformat für Ausgabe
DFSTACK	Datumsformat für den Natural-Stack
DFTITLE	Datumsformat in Standard-Seitenüberschrift
DTFORM	Datumsformat
STACKD	Stack-Kommando-Abgrenzungszeichen
TD	Zeitdifferenz
YD	Jahresdifferenz einstellen
YSLW	Gleitendes oder festes Jahr-Fenster

## Debugging und Tracing

---

Folgende Profilparameter können Sie für die Fehlersuche und zur schrittweisen Protokollierung einer Programmausführung benutzen:

Parameter	Kurzbeschreibung
<a href="#">CANCEL</a>	Abbruch der Session mit Dump
<a href="#">DBGAT</a>	Debug Attach Server für NaturalONE
<a href="#">DBGERR</a>	Automatischer Debugger-Start bei Laufzeitfehler
<a href="#">DU</a>	Dump-Erstellung
<a href="#">DUE</a>	Dump-Erstellung, fehlerspezifisch
<a href="#">ETRACE</a>	Externe Trace-Funktion aktivieren/deaktivieren
<a href="#">ITRACE</a>	Interne Trace-Funktion aktivieren/deaktivieren
<a href="#">PECK</a>	PCHECK/ECHECK-Fehlerverarbeitung, siehe auch Systemkommando COMPOPT
<a href="#">RDC</a>	Konfiguration des Natural Data Collector
<a href="#">RELO</a>	Speicher-Thread-Verschiebung
<a href="#">TRACE</a>	Komponenten für den Trace angeben

## Externe Subprogramme

Folgende Profilparameter beeinflussen das dynamische Laden und Löschen von Nicht-Natural-Programmen:

Parameter	Kurzbeschreibung
<a href="#">CDYNAM</a>	Dynamisches Laden von Nicht-Natural-Programmen
<a href="#">CSTATIC</a>	Statisch verlinkte Module
<a href="#">DELETE</a>	Löschung dynamisch geladener Nicht-Natural-Programme
<a href="#">LIBNAM</a>	Name der externen Programm-Load Library (nur bei z/OS Batch und TSO)
<a href="#">PGP</a>	Eigenschaften für externe Programme
<a href="#">RCA</a>	Adressen von statischen Nicht-Natural-Programmen auflösen
<a href="#">RCALIAS</a>	Externe Namen für Nicht-Natural-Programme

## Maximalwerte (Limits)

Folgende Profilparameter können Sie benutzen, um zu verhindern, dass ein einzelnes Programm eine zu große Menge an internen Ressourcen in Anspruch nimmt:

Parameter	Kurzbeschreibung
LE	Reaktion auf Limit-Überschreitung bei Verarbeitungsschleifen
LT	Limit für Verarbeitungsschleifen
MADIO	Maximale Anzahl der DBMS-Aufrufe zwischen Bildschirm-Ein-/Ausgaben
MAXCL	Maximale Anzahl an Programmaufrufen
MAXYEAR	Maximale Jahresangabe bei Datums-/Zeitwerten
MT	Maximale CPU-Zeit
OVSIZE	Speicher-Thread-Überlaufgröße
PD	Seiten-Limit für NATPAGE

## Überwachung

---

Folgende Profilparameter können Sie benutzen, um Ihre Anwendungen und Ressourcen zu überwachen:

Parameter	Kurzbeschreibung
RDC	Konfiguration des Natural Data Collector
RDCEXIT	User Exits für den Natural Data Collector
RDCSIZE	Puffergröße für den Natural Data Collector

## Natural-Systemdateien

---

Natural-Systemdateien werden zum Speichern von Daten und Programmen benutzt. Siehe *Natural-Systemdateien* in der *Natural System-Architektur-Dokumentation*.

Folgende Profilparameter gelten für alle Systemdateien:

Parameter	Kurzbeschreibung
DBID	DBID - Standard-Datenbankkennung für Natural-Systemdateien
FNR	Standard-Dateinummer der Natural-Systemdateien
SYSPSW	Standard-Passwort für Natural-Systemdateien
SYSCIP	Standard-Chiffrierschlüssel für Natural-Systemdateien
RFILE	Datei für Aufzeichnung per Recording-Funktion
ROSY	Schreibgeschützter Zugriff auf Systemdateien (nur bei FNAT, FUSER und FSEC)

Mit den folgenden Profilparametern können Sie die Standardwerte für einzelne Systemdateien überschreiben:

Parameter	Kurzbeschreibung
FNAT	Natural-Systemdatei für Systemprogramme
FUSER	Natural-Systemdatei für Benutzerprogramme
FDIC	Predict-Systemdatei
FSEC	Natural Security-Systemdatei
FSP00L	Spool-Datei für Natural Advanced Facilities
FPROF	Natural-Systemdatei für Parameterprofile
FREG	Natural-Registry-Systemdatei

## Natural und andere Produkte der Software AG

Die folgenden Natural-Profilparameter werden im Zusammenhang mit anderen Produkten der Software AG verwendet:

Produkt	Parameter	Kurzbeschreibung
Adabas Text Retrieval	TSIZE	Größe des Puffers für Adabas Text Retrieval
Con-nect	CSIZE	Größe des Con-nect/Con-form-Pufferbereichs
EntireX Broker	BSIZE	Größe des EntireX Broker Buffer
Entire DB Engine	ZSIZE	Größe des Entire DB-Pufferbereichs
Entire System Server	ASIZE	Entire System Server-Hilfspuffer
Natural Advanced Facilities	NAFSIZE	Puffergröße für Natural Advanced Facilities
	NAFUPF	Benutzerprofil für Natural Advanced Facilities
Natural Connection	PC	Personal-Computer-Zugriffsmethode für Natural Connection
Natural for Db2	DB2	Parameter für SQL-Datenbank-Verwaltungssystem-Schnittstellen
	DB2SIZE	Natural-Pufferbereich für Db2
Natural for DL/I	DLISIZE	Pufferbereichsgröße für DL/I
Natural for VSAM	VSAM	Parameter für Natural for VSAM
	VSIZE	Größe des Pufferbereichs für Natural for VSAM
NaturalONE	PDPSIZE	Daten-Pool für NaturalONE Profiler
	DBGAT	Debug Attach Server für NaturalONE
Natural Optimizer Compiler	OPT	Steuerung des Natural Optimizer Compiler
Natural RPC	RPC	Remote Procedure Call-Einstellungen
Software AG Editor	EDBP	Software AG Editor Buffer Pool-Definitionen
	EDPSIZE	Größe des Software AG Editor-Hilfs-Buffer-Pool
	SSIZE	Größe des Software AG Editor-Arbeitsbereichs
Natural Web I/O Interface	WEBIO	Web I/O Interface Rendering

Produkt	Parameter	Kurzbeschreibung
Natural zIIP Enabler	<a href="#">Z I I P</a>	zIIP-Verarbeitung (IBM's System z Integrated Information Processor, nur bei z/OS)

## Leistungsoptimierung

Die folgenden Profilparameter beeinflussen die Leistungsfähigkeit (Performance) des Systems:

Parameter	Kurzbeschreibung
<a href="#">OPT</a>	Steuerung des Natural Optimizer Compiler
<a href="#">Z I I P</a>	zIIP-Verarbeitung (IBM's System z Integrated Information Processor, nur bei z/OS)

## Ausgabe-Reports und Arbeitsdateien

Folgende Profilparameter steuern Standardattribute, die beim Erstellen von Natural-Reports zur Anwendung kommen:

Parameter	Kurzbeschreibung
<a href="#">DL</a>	Ausgabelänge
<a href="#">EJ</a>	Seitenvorschub
<a href="#">FAMSTD</a>	Überschreiben der Arbeitsdatei-Zugriffsmethoden-Zuweisungen
<a href="#">HCAM</a>	Hardcopy-Zugriffsmethode
<a href="#">HCDEST</a>	Ziel für Hardcopy-Ausgabe
<a href="#">INTENS</a>	Drucken von hervorgehobenen Feldern
<a href="#">LS</a>	Zeilenlänge
<a href="#">MAINPR</a>	Standard-Ausgabe-Report-Nummer
<a href="#">MP</a>	Maximale Seitenzahl eines Reports
<a href="#">PCNTRL</a>	Druckersteuerzeichen
<a href="#">PM</a>	Druck-Modus
<a href="#">PRINT</a>	Druckdatei-Zuweisungen
<a href="#">PS</a>	Länge einer Reportseite
<a href="#">SF</a>	Spaltenabstand
<a href="#">TQ</a>	Umsetzung von Anführungszeichen
<a href="#">TS</a>	Umsetzung von System-Library-Programmausgaben
<a href="#">WORK</a>	Arbeitsdatei-Zuweisungen
<a href="#">ZP</a>	Anzeige von Nullwerten

## Remote Procedure Call-Einstellungen (RPC)

---

Der folgende Profilparameter dient dazu, Remote Procedure Calls einzurichten und zu verarbeiten, die zwischen einem Client und einem Server unter Verwendung eines Natural RPC erfolgen.

Parameter	Kurzbeschreibung
RPC	Remote Procedure Call-Einstellungen für den Natural RPC.

## Steuerung der Natural-Session

---

Folgende Profilparameter steuern die Natural-Session:

Parameter	Kurzbeschreibung
CM	Kommandomodus
ETA	Fehlertransaktionsprogramm
FREEGDA	GDA im Utility-Modus freigeben
MENU	Menü-Modus
NC	Verwendung von Natural-Systemkommandos
SORT	Steuerung des Sortierprogramms
STEPLIB	Zusätzliche Steplib Library
SYNERR	Übergabe von Syntaxfehlern
UCONMAX	Maximale Anzahl gleichzeitiger Sessions für einen Benutzer
ULANG	Benutzersprache
ZD	Division durch Null

## Initialisierung und Beendigung einer Natural-Session

---

Folgende Profilparameter beeinflussen die Initialisierung und Beendigung einer Natural-Session:

Parameter	Kurzbeschreibung
AUTO	Automatische Anmeldung
ENDMSG	Session-Ende-Meldung anzeigen
IMSG	Nachricht bei Session-Initialisierungsfehler
ITERM	Session-Beendigung bei Initialisierungsfehler
MENU	Menü-Modus
NUCNAME	Name des gemeinsam genutzten Nukleus
PROGRAM	Kontrolle durch Nicht-Natural-Programm nach Session-Beendigung
STACK	Daten/Kommandos auf den Natural-Stack stellen
STEPLIB	Zusätzliche Steplib Library

## Source-Verwaltung

---

Die folgenden Profilparameter können Sie zur Natural-Source-Verwaltung benutzen:

Parameter	Kurzbeschreibung
RECAT	Dynamisches Rekatalogisieren
SL	Sourcecode-Zeilenlänge
SLOCK	Source-Sperrung
SYNERR	Übergabe von Syntaxfehlern

## Speicherverwaltung

---

Die folgenden Profilparameter beeinflussen die Natural-Pufferspeicher und die Buffer Pools:

Parameter	Kurzbeschreibung
BP82	Buffer Pool-Platzhalter wenn Objekt nicht gefunden
BPCSIZE	Cache-Größe für den Natural Buffer Pool
BPC64	Buffer Pool Cache-Speichertyp
BPI	Buffer Pool-Initialisierung
BPLIST	Name der Preload-Liste für den Natural Buffer Pool
BPMETH	Suchalgorithmus zur Zuweisung von Speicherplatz im Buffer Pool
BPNAME	Name des globalen Natural Buffer Pool
BPPROP	Global Buffer Pool-Weiterleitung
BPSFI	Objektsuche zuerst im Buffer Pool

Parameter	Kurzbeschreibung
BPSIZE	Größe des lokalen Natural Buffer Pool
BPTXT	Größe der Textsegmente im Natural Buffer Pool
CMPR	Standard-Algorithmus zur Komprimierungsoptimierung
DATSIZE	Minimale Größe des lokalen Daten-Puffers
DS	Größe der Pufferspeicher festlegen
DSIZE	Größe des DBLOG-Puffers
ESIZE	Größe des User Buffer-Erweiterungsbereichs
ISIZE	Größe des Puffers für Initialisierung
MAXROLL	Anzahl der CMROLL-Aufrufe vor vorübergehender Session-Unterbrechung
MONSIZE	Größe des SYSTP Monitor-Puffers
PDPSIZE	Daten-Pool für NaturalONE Profiler (nur beim NaturalONE-Server)
RJESIZE	Anfangsgröße des NATRJE-Puffers
RUNSIZE	Größe des Laufzeit-Puffers
THSIZE	Größe des Speicher-Thread
WPSIZE	Größe der Natural-Arbeitsspeicher

## Terminal-Kommunikation

Folgende Profilparameter beeinflussen die Benutzung von Natural an Video-Terminals:

Parameter	Kurzbeschreibung
ATTN	Unterstützung für Programm-Unterbrechungstaste
CLEAR	Verarbeitung der CLEAR-Taste im NEXT-Modus
DO	Anzeige-Reihenfolge von Ausgabedaten
DSC	Datenstromkomprimierung bei Terminals des IBM-Typs 3270
EMFM	Editiermasken-Frei-Modus
ESCAPE	Terminalkommandos %% und % . ignorieren
IKEY	Verarbeitung von PA- und PF-Tasten
IM	Input-Modus
KEY	PA-, PF- und CLEAR-Tastenbelegungen zuweisen
LC	Umsetzung von Klein- in Großbuchstaben
ML	Position der Meldungszeile
MSGSF	Anzeigeformat von Systemfehlermeldungen
OPF	Überschreiben geschützter Felder durch Helproutinen
REINP	Interner REINPUT bei ungültigen Daten

Parameter	Kurzbeschreibung
RM	Rückübertragung von geänderten Feldern
SA	Terminal-Warnton
SOSI	Shift-Out/Shift-In-Codes für Doppel-Byte-Zeichensätze
TMODEL	IBM 3270-Terminal-Modell
TTYTYPE	Terminaltyp

## Zeichenumsetzungstabellen

---

Folgende Profilparameter können Sie benutzen, um Zeichenumsetzungstabellen zu ändern, die von Natural verwendet werden:

Parameter	Kurzbeschreibung
CCHAR	Ausgabe-Steuerzeichen zulassen
CCTAB	Drucker-ESC-Folge-Definition
CP	Name der Standard-Codepage
SCTAB	SCTAB - Scanner-Zeichentabelle
TAB	Standard-Ausgabezeichen-Umsetzung
TABA1	EBCDIC/ASCII-Zeichenumsetzung
TABA2	ASCII/EBCDIC-Zeichenumsetzung
TABL	Translation table for output from SYS libraries
TAB1	Alternative Ausgabezeichen-Umsetzung
TAB2	Alternative Eingabezeichen-Umsetzung
UTAB1	Umsetzung von Klein- in Großschreibung
UTAB2	Umsetzung von Groß- in Kleinschreibung

## Unicode- und Codepage-Unterstützung

---

Den folgende Profilparameter können Sie für die Unicode- und Codepage-Unterstützung benutzen:

Parameter	Kurzbeschreibung
CFICU	Unicode- und Codepage-Unterstützung ein-/ausschalten
CPCVERR	Codepage-Umsetzungsfehler

## Nutzungs- und Konfigurationseinstellungen für Profilparameter

Folgende Profilparameter beeinflussen die Nutzungs- und Konfigurationseinstellungen für Profilparameter:

Parameter	Kurzbeschreibung
DYNPARM	Verwendung dynamischer Parameter
PARM	Alternatives Parametermodul
PLOG	Protokollieren von dynamischen Parametern
PROFILE	Dynamisches Parameterprofil aktivieren
SYS	Satz dynamischer Profilparameter definieren und aktivieren
USER	Verwendung von Profilparameterketten und Modulen einschränken

## Unterstützung von Statements für den Internet- und XML-Zugriff

Der folgende Profilparameter unterstützen den Zugang zum Internet und zu XML-Daten.

Parameter	Kurzbeschreibung
WEBIO	Web I/O Interface Rendering (nur bei installiertem Web I/O Interface)
XML	Einstellungen der Statements PARSE XML und REQUEST DOCUMENT aktivieren/deaktivieren

## TP-Monitor-Schnittstellen

Folgende Profilparameter gelten bei Verwendung von Natural mit einer TP-Monitor-Schnittstelle:

Parameter	Kurzbeschreibung
COMP	Parameter für Natural Com-plete/SMARTS Interface
IMSP	Allgemeine Parameter für das Natural IMS TM Interface
IMSPE	Umgebungsparameter für das Natural IMS TM Interface
IMSPT	Transaktionsdefinitionen für Natural IMS TM Interface
OUTDEST	Ausgabeziel für asynchrone Verarbeitung unter CICS und Com-plete
PSEUDO	Pseudo-konversationeller Modus (unter CICS)
SENDER	Bestimmungsort für Bildschirmausgabe bei asynchroner Verarbeitung unter CICS, Com-plete und IMS TM
SKEY	Speicherschutzschlüssel
SUBSID	Subsystemkennung (unter z/OS)
TSOP	Parameter für Natural TSO Interface

# 5

## ABLOG – Programmausführungsfehler protokollieren

---

Mit diesem Profilparameter können Sie festlegen, dass alle Natural-Programmausführungsfehler protokolliert werden, die eine Zeitüberschreitung oder einen Programmabbruch während der Ausführung eines Programms anzeigen (Fehlermeldungen NAT0953 bis NAT0956)

Die Protokoll-Aufzeichnungen enthalten folgende Informationen: Abbruchcode, Abbruchadresse im Speicher sowie aktuelle Natural Library, Programm- und Statement-Zeilenummer, wo der Fehler aufgetreten ist.

<b>Mögliche Werte</b>	ON	Die Natural-Fehler NAT0953 bis NAT0956 werden protokolliert und in die Protokolldatei für Meldungen geschrieben.
	OFF	Die Natural-Fehler NAT0953 bis NAT0956 werden nicht protokolliert.
<b>Standard-Einstellung</b>	OFF	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	



# 6 AD - Attribut-Definition

---

▪ AD-Parameter-Syntax .....	30
▪ Feldanzeige .....	31
▪ Ausrichtung der Feldwerte .....	32
▪ Feldeingabe/ausgabe-Charakteristika .....	32
▪ Auslegung alphanumerischer Felder .....	34
▪ Eingabezwang .....	34
▪ Mindestlänge der Eingabewerte .....	34
▪ Groß-/Kleinschreibung .....	35
▪ Füllzeichen .....	35

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

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

<b>Mögliche Werte</b>	Siehe <a href="#">AD-Parameter-Syntax</a> .	Sie können mehrere Attribute in beliebiger Reihenfolge angeben.
<b>Standard-Einstellung</b>	Siehe unten.	
<b>Gültige Statements</b>	FORMAT	
	DISPLAY INPUT NEWPAGE WITH TITLE PRINT REINPUT WRITE WRITE TITLE WRITE TRAILER	Parameter können auf Statement-Ebene und/oder Element-Ebene angegeben werden.
<b>Gültiges Kommando</b>	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.
	Keines.	

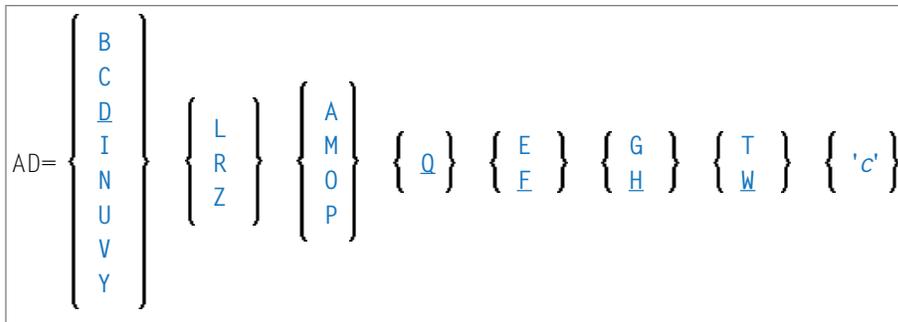
Die Parameter-Syntax und die Bedeutung der Attribute und die einzelnen Werte werden in den folgenden Abschnitten erklärt.

## 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	Der Feldwert wird unterstrichen angezeigt.
V	invers (*)	WRITE	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 Function Call	<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 <b>DY</b> -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 Gültigkeitsbereich des Profil- oder Session-Parameters <b>PM</b> (PM=I) 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. Dies ist die Voreinstellung.  <b>Anmerkung:</b> Um AD=W einzuschalten, müssen Sie den Wert ON für den Natural-Profilparameter <a href="#">LC</a> angeben.

## Füllzeichen

Wert	Bedeutung	Statements	Erklärung
' c '	Füllzeichen	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.

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.

# 7 ADAACBX – Verwendung des erweiterten Adabas Control Block

Dieser Natural-Profilparameter legt fest, ob bei Betrieb mit Adabas-Version 8 oder höher der Natural-Nukleus anstelle des klassischen Adabas Control Block (ACB) den erweiterte Adabas Control Block (ACBX) verwendet, um auf Natural-Systemdateien zuzugreifen.

<b>Mögliche Werte</b>	ON	Der erweiterte Adabas Control Block (ACBX) wird für den Zugriff auf Natural-Systemdateien verwendet.
	OFF	Der klassische Adabas Control Block (ACB) wird für den Zugriff auf Natural-Systemdateien verwendet.
<b>Standard-Einstellung</b>	OFF	
<b>Dynamische Angabe</b>	Ja	
<b>Angabe innerhalb der Session</b>	Nein	

 **Wichtig:** Wenn Sie Adabas-Link-Routinen-User Exits haben, bei denen ACB erwartet wird, müssen Sie ADAACBX=OFF setzen, um Probleme zu vermeiden.

ADAACBX hat keine Beeinträchtigung von Adabas-Aufrufen zur Folge, die von Natural-Programmen abgesetzt werden. Deren Verwendung des ACBX wird durch die Adabas-Version gesteuert, die als Datenbanktyp mit dem Profilparameter DB angegeben wird. Zum Beispiel: DB=(ADAV8) für Adabas Version 8. Seit Adabas Version 8 wird der erweiterte Adabas Control Block (ACBX) verwendet

Wir empfehlen Ihnen die Einstellung ADAACBX=ON. In diesem Fall werden Natural-Objekte in Segmenten von 64 KB anstelle von 32 KB von der Natural-Systemdatei geladen, wodurch die Anzahl der Adabas-Aufrufe bei großen Objekten verringert wird. Dazu müssen Sie Folgendes berücksichtigen:

- Die Speichererfordernisse im Natural-Adressraum erhöhen sich um 64 KB.

- Der Adabas-Parameter LU muss auf 64 KB (Standardwert) oder auf einen höheren Wert gesetzt werden.
- Die Anzahl der mit Adabas verbundenen Puffer muss eventuell erhöht werden.

# 8

## ADAMODE - Adabas-Call-Schnittstellenmodus

---

Dieser Natural-Profilparameter steuert den Adabas-Call-Schnittstellenmodus und die Anzahl der Adabas-Benutzer-Sessions, die von Natural verwendet werden, um Adabas-Aufrufe abzusetzen.

<b>Mögliche Werte</b>	Siehe weiter unten.
<b>Standard-Einstellung</b>	2
<b>Dynamische Angabe</b>	Ja
<b>Angabe innerhalb der Session</b>	Nein

Mögliche Werte:

Wert	Separate Adabas-Benutzer-Sessions für Nukleus- und Benutzeranwendungs-Datenbankaufrufe [1]	3GL Programm-Adabas-Aufrufe benutzen Natural's Adabas-Session für Benutzeranwendungsaufufe [2]	Image-Umschaltung in einer z/OS Parallel Sysplex-Umgebung unterstützt [3]
0	Nein	Ja	Nein
1	Nein	Nein	Ja
2	Ja	Nein	Ja
3	Ja	Ja	Nein

### Anmerkungen:

#### 1. Unterstützung von ADAMODE-Einstellungen

Wenn der für ADAMODE gesetzte Wert nicht von der in Ihrer Umgebung benutzten Adabas Link Routine unterstützt wird, erfolgt eine Fehlermeldung und der Wert von ADAMODE wird auf 0 gesetzt.

## 2. Separate Adabas-Benutzer-Sessions für Nukleus- und Benutzeranwendungs-Datenbankaufrufe

### Zwei getrennte Adabas-Benutzer-Sessions

Wenn Natural zwei getrennte Adabas-Benutzer-Sessions verwendet, um Adabas-Aufrufe abzusetzen, dann benutzt Natural die eine Adabas-Benutzer-Session, um vom Natural-Nukleus abgesetzte Adabas-Aufrufe zu behandeln (z.B. um Natural-Objekte aus der Systemdatei zu laden), und die andere Adabas-Benutzer-Session, um Adabas-Aufrufe abzusetzen, die von der Benutzeranwendung abgesetzt werden.

Eine Adabas-Zeitüberschreitung (die zum Natural-Fehler NAT3009 führt), die bei der Adabas-Benutzer-Session auftritt, die zur Handhabung von Adabas-Aufrufen durch den Natural-Nukleus verwendet wird, hat keine Auswirkung auf die Benutzeranwendung.

Ein separates Adabas User Queue Element (UQE) wird für jede Adabas-Benutzer-Session erzeugt, und es kann nötig sein, den Wert des Adabas-ADARUN-Parameters NU zu erhöhen.

### Nur eine Adabas-Benutzer-Session

Wenn Natural nur eine Adabas-Benutzer-Session verwendet, werden `END TRANSACTION-` und `BACKOUT TRANSACTION-` Statements entweder vom Natural-Nukleus abgesetzt oder die Benutzeranwendung veranlasst Transaktionen, die sowohl vom Natural-Nukleus als auch der Benutzeranwendung gestartet werden.

Eine Adabas-Zeitüberschreitung (die zum Natural-Fehler NAT3009 führt), die bei der Adabas-Benutzer-Session auftritt, wird immer an die Benutzeranwendung gemeldet, weil es nicht möglich ist zu prüfen, ob die Zeitüberschreitung den Transaktionsstatus der Anwendung betrifft.

Wenn Natural nur eine Adabas-Benutzer-Session verwendet, um Adabas-Aufrufe abzuwickeln, die vom Natural-Nukleus abgesetzt werden, ebenso wie Adabas-Aufrufe, die von der Benutzeranwendung abgesetzt werden, dann ist nur eine UQE nötig.

## 3. 3GL Programm-Adabas-Aufrufe benutzen Natural's Adabas-Session für Benutzeranwendungsaufrufe

### Aufrufe, die Natural's Adabas-Session benutzen

Wenn ein 3GL-Programm, das von der Benutzeranwendung aufgerufen wird, Adabas-Aufrufe absetzt, und wenn diese Adabas-Aufrufe nicht Natural's Adabas-Session für Benutzeranwendungsaufrufe verwenden, dann sind diese Adabas-Aufrufe an der Transaktionsabwicklung der Benutzeranwendung beteiligt (`END TRANSACTION-` und `BACKOUT TRANSACTION-` Statements), und sie werden von den Parametereinstellungen beeinflusst, die sich auf die Natural-Transaktionsabwicklung beziehen (siehe unten aufgeführte Parameter).

## Aufrufe, die Natural's Adabas-Session nicht benutzen

Wenn ein 3GL-Programm, das von der Benutzeranwendung aufgerufen wird, Adabas-Aufrufe absetzt, und wenn diese Adabas-Aufrufe nicht Natural's Adabas-Session für Benutzeranwendungsaufrufe verwenden, dann sind diese Adabas-Aufrufe nicht an der Transaktionsabwicklung der Adabas-Benutzer-Session beteiligt. Folglich müssen solche 3GL-Programme eine eigene Transaktionsabwicklung durchführen.

### 4. Image-Umschaltung in einer z/OS Parallel Sysplex-Umgebung wird unterstützt

Falls die Image-Umschaltung in einer z/OS Parallel Sysplex-Umgebung unterstützt wird, kann die Natural-Session nach einer Terminal-E/A-Operation nahtlos mit der Ausführung in einem z/OS Image weitermachen, das von dem z/OS Image verschieden ist, in dem Natural vor der Terminal-E/A-Operation ausgeführt hat. Für die Unterstützung in einer z/OS Parallel Sysplex-Umgebung ist die Installation des Natural Roll Server notwendig.

-  **Vorsicht:** Wenn der Wert von ADAMODE so gesetzt wird, dass die Image-Umschaltung in einer z/OS Parallel Sysplex-Umgebung nicht unterstützt wird, kann das zu unvorhersehbaren Ergebnissen führen, wenn die Natural-Session die Ausführung nach der Terminal-E/A-Operation in einem anderen z/OS Image mit der Ausführung fortsetzt. Abhängig von den Parametereinstellungen, die sich auf die Natural-Transaktionsverarbeitung beziehen (siehe unten aufgeführte Parameter) kann das Folgendes beinhalten:
- Nicht-Null-Adabas-Antwortcodes (die zum Beispiel zum Natural-Fehler NAT3021 führen),
  - Datenbankaktualisierungen, die noch nicht durch ein END TRANSACTION-Statement festgeschrieben sind, werden unbeabsichtigt rückgängig gemacht oder gegen die Datenbank ausgeführt.

Weitere Parameter mit Bezug zur Transaktionsverarbeitung: [DBCLOSE](#) | [DBOPEN](#) | [ENDBT](#) | [ET](#) | [ETDB](#) | [ETEOP](#) | [ETIO](#) | [ETSYNC](#)



# 9

## ADANAME - Name der Adabas-Link-Routine

---

Dieser Natural-Profilparameter gibt den Namen der zu verwendenden Namen der Adabas-Link-Routine an.

<b>Mögliche Werte</b>	1 - 8 Zeichen	Gültiger Modul- oder Eintragsname.
<b>Standard-Einstellung</b>	ADABAS	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	



### Anmerkungen:

1. ADANAME gilt nicht bei Com-plete.
2. Wenn die Adabas-Link-Routine mit dem Natural-Parametermodul verlinkt ist und ihr Eintragsname mit dem durch ADANAME in dem Parametermodul angegebenen Namen übereinstimmt, dann wird die verlinkte Routine verwendet. Falls sie nicht verlinkt ist, wird die verlinkte Routine dynamisch hinzu geladen. Dadurch entfällt die Notwendigkeit, das Adabas-Link-Modul statisch mit dem Natural-Nukleus zu verlinken.
3. Es ist möglich, denselben Natural-Nukleus mit verschiedenen Adabas-Link-Modulen zu betreiben.
4. Unter CICS darf die Adabas-Link-Routine nicht mit Natural verlinkt werden.



# 10 ADAPRM - Adabas Review-Unterstützung

---

Dieser Natural-Profilparameter dient dazu, Natural-Session-Daten innerhalb des siebten Adabas-Puffers an Adabas Review zu übergeben.

<b>Mögliche Werte</b>	ON	Natural-Session-Daten werden übergeben. <b>Anmerkung:</b> Setzen Sie ADAPRM auf ON, wenn Adabas Review installiert ist.
	OFF	Keine Natural-Session-Daten werden übergeben.
<b>Standard-Einstellung</b>	OFF	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	



# 11 ADASBV - Adabas Security durch Einstellung

---

Dieser Natural-Profilparameter kann benutzt werden, um ungültige Ergebnisse bei Zugriffen auf Adabas-Dateien zu verhindern, welche durch „Security-by-Setting“ geschützt sind. Wenn auf eine solche Datei zugegriffen wird, können in einigen Fällen, in denen kein Formatpuffer erzeugt und an Adabas übergeben wird, ungültige Ergebnisse zurückgeliefert werden.

<b>Mögliche Werte</b>	ON	Natural-Session-Daten werden übergeben.  <b>Anmerkung:</b> Es wird empfohlen, dass Sie die Einstellung ADASBV=ON verwenden, wenn Sie auf Adabas-Dateien zugreifen, die durch „Security-by-Setting“ geschützt sind. Dann wird immer ein Format-Puffer an Adabas für einen Datenzugriff an Adabas übergeben (selbst dann, wenn dies ein 2 Byte-Dummy-Puffer ist), wodurch ungültige Ergebnisse vermieden werden.
	OFF	Es werden keine Natural-Session-Daten übergeben.
<b>Standard-Einstellung</b>	OFF	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	



# 12 AL - 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.

<b>Mögliche Werte</b>	1 bis $n$	$n =$ Wert (Zeilenlänge) des Parameters <code>LS</code> minus 1.
<b>Standard-Einstellung</b>	Keine	
<b>Gültige Statements</b>	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.
<b>Gültiges Kommando</b>	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
```

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



# 13 ASIZE - Entire System Server-Hilfspuffer

Dieser Natural-Profilparameter bestimmt die Größe des Hilfspuffers für den Entire System Server.

<b>Mögliche Werte</b>	64 - 512	<p>Puffergröße in KB.</p> <p><b>Anmerkung:</b></p> <ol style="list-style-type: none"> <li>1. Wenn der Entire System Server verwendet werden soll, <i>muss</i> dieser Parameter gesetzt werden. Siehe <i>Entire System Server</i>-Dokumentation.</li> <li>2. Der Wert 64 wird empfohlen. Er sollte für die meisten Anwendungen ausreichend sein.</li> <li>3. Ein größerer ASIZE-Wert als 64 KB wird benötigt, wenn Aufrufe an den Entire System Server erfolgen sollen, bei denen größere Datenmengen in multiplen Feldern transportiert werden sollen. Beispiel: Die View SEND-EMAIL kann 3 x 20 E-mail-Adressen enthalten, wobei jede E-mail-Adresse mit 128 Zeichen definiert ist.</li> </ol>
	0	Wenn ASIZE=0 angegeben wird oder wenn der angeforderte Speicherplatz nicht zur Verfügung steht, kann der Entire System Server nicht aktiviert werden.
<b>Standard-Einstellung</b>	0	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	



## Anmerkungen:

1. ASIZE gilt nur, wenn der Entire System Server installiert ist.
2. Als Alternative zu ASIZE können Sie den äquivalenten Natural-Profilparameter [DS](#) oder das Parametermakro [NTDS](#) benutzen.



# 14 ASPSIZE (für interne Verwendung)

---

Dieser Natural-Profilparameter ist obsolet und wird nur noch aus Kompatibilitätsgründen akzeptiert.



# 15

## ATTN - Unterstützung für Programm-Unterbrechungstaste

---

<b>Mögliche Werte</b>	ON	Die Unterbrechungstaste bewirkt, dass die Natural-Verarbeitung unterbrochen wird.
	OFF	Die Unterbrechungstaste wird ignoriert.
<b>Standard-Einstellung</b>	ON	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	



### Anmerkungen:

1. Dieser Natural-Profilparameter steuert die Verwendung der Programm-Unterbrechungstaste bei IBM SNA Terminals. Durch Drücken dieser Taste kann die Natural-Verarbeitung mit einer entsprechenden Meldung (NAT1016) unterbrochen werden.
2. Ob eine Unterbrechungstaste vorhanden ist, hängt von der Umgebung und dem Terminaltyp ab.



# 16 AUTO - Automatische Anmeldung

---

Dieser Natural-Profilparameter bewirkt zu Beginn der Natural-Session eine automatische Anmeldung (Logon) für eine spezifische Library.

<b>Mögliche Werte</b>	ON	Eine automatische Anmeldung wird zu Beginn der Natural-Session ausgeführt.
	OFF	Es wird keine automatische Anmeldung zu Beginn der Natural-Session ausgeführt.
<b>Standard-Einstellung</b>	OFF	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	
<b>Anwendungsprogrammierschnittstelle</b>	USR1005N	Siehe <i>SYSEXT Utility</i> in der Dokumentation <i>Debugger und Dienstprogramme</i> .



## Anmerkungen:

1. Der Wert der Systemvariablen \*INIT-USER wird als Benutzerkennung für die Anmeldung verwendet.
2. Bei Verwendung mit Natural Security verhindert AUTO=ON eine Anmeldung mit einer anderen Benutzerkennung (weitere Informationen siehe *Natural Security*-Dokumentation).



# 17

## BP82 - Buffer Pool-Platzhalter wenn Objekt nicht gefunden

Dieser Natural-Profilparameter gibt an, ob ein Platzhalter in den Natural Buffer Pool gestellt wird, wenn ein Objekt nicht in einer Library gefunden worden ist.

<b>Mögliche Werte</b>	ON	Wenn ein Objekt nicht gefunden wurde, wird ein Platzhalter in den Buffer Pool gestellt.  <b>Anmerkung:</b>  1. Aus Sicht des Buffer Pool Managers ist dieser Platzhalter ein gewöhnliches Objekt. Für den Natural Object Loader ist er ein Hinweis darauf, dass keine Notwendigkeit besteht, nach diesem Objekt in der Systemdatei zu suchen.  2. Weitere Einzelheiten siehe <i>Beispiel</i> .
	OFF	Wenn ein Objekt nicht gefunden wurde, wird kein Platzhalter in den Buffer Pool gestellt.
<b>Standard-Einstellung</b>	OFF	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	

### Beispiel:

Angenommen der Natural Buffer Pool ist leer, Sie haben Natural mit dem Parameter STEPLIB=XYPSTEP gestartet, sich in der Library XYLIB angemeldet, und Sie versuchen das Programm XYPROG auszuführen, das sich in der Library SYSTEM in der Systemdatei FUSER befindet. Beim Laden des Programms XYPROG sucht Natural zuerst in der Library XYLIB, danach in der Library XYPSTEP und findet das Programm schließlich in der Library SYSTEM in FUSER.

Wenn der Profilparameter **BPSFI** (*Objektsuche zuerst im Buffer Pool*) auf OFF gesetzt ist, führt jeder Benutzer, der dasselbe macht, Datenbankaufrufe aus, um nach dem Objekt XYPROG in der Library XYLIB und in XYPSTEP zu suchen, aber diese Datenbankaufrufe sind niemals erfolgreich. Um diese

unnötigen Datenbankaufrufe zu verhindern, wird ein Platzhalter in den Buffer Pool gestellt. Wenn Sie mit der Funktion **List Objects** des Dienstprogramms SYSBPM in den Buffer Pool schauen, sehen Sie diesen Platzhalter als gewöhnliches Objekt:

```
14:34:39          ***** NATURAL SYSBPM UTILITY *****          2011-05-02
BPNAME QA82GBP          - List Objects -          Type Global Nat
BPPROP OFF          Loc DAEF QA82
C Library Object      DBID  FNR Loc RLD Use Max  Reuse      TotalUC ObjSize Sto
*
___ XYLIB      XYPROG      10   32 B          1          1      90   4
___ SYSTEM    XYPROG      10   32 B          1          1  2,656  4
___ XYPSTEP   XYPROG      10   32 B          1          1      90   4
```

# 18

## BPC64 - Buffer Pool Cache-Speichertyp

---

Dieser Profilparameter dient zur Angabe des Speichertyps für den Buffer Pool Cache eines lokalen Natural Buffer Pool.

<b>Mögliche Werte</b>	ON	Für den Buffer Pool Cache soll virtueller Speicher oberhalb der 2 GB-Grenze verwendet werden.
	OFF	Für den Buffer Pool Cache soll Datenspeicherplatz verwendet werden.
<b>Standard-Einstellung</b>	OFF	
<b>Dynamische Angabe</b>	ja	Dieser Parameter kann nur dynamisch angegeben werden.
<b>Angabe innerhalb der Session</b>	nein	



### Anmerkungen:

1. Dieser Natural-Profilparameter gilt nur unter z/OS (nicht bei Com-plete).
2. Er entspricht dem Subparameter **C64** des Profilparameters **BPI** bzw. des Makros **NTBPI**.
3. Intern wird die **BPC64**-Angabe in die entsprechende **BPI**-Angabe umgewandelt; zum Beispiel: **BPC64=ON** wird umgewandelt in **BPI=(TYPE=NAT,SEQ=0,C64=ON)**.
4. Der Profilparameter **BPC64** gilt nur bei einem primären Natural Buffer Pool (**TYPE=NAT, SEQ=0**). Im Falle eines globalen Buffer Pool wird er ignoriert. Bei Vorhandensein eines primären Natural Buffer Pool mit **SEQ=0** im Natural-Parametermodul wird nur die **C64**-Einstellung dieses Buffer Pool aktualisiert.
5. In einer Umgebung mit mehreren Benutzern (z.B. unter CICS) betrifft der Profilparameter **BPC64** nur die allererste Natural-Session, die den lokalen Buffer Pool initialisiert.
6. Allgemeine Informationen zum Natural Buffer Pool siehe *Natural Buffer Pools* in der *Operations*-Dokumentation.



# 19 BPCSIZE - Cache-Größe für den Natural Buffer Pool

Dieser Profilparameter dient zur Angabe der Größe des Natural Buffer Pool Cache (in KB) bei einem lokalen Natural Buffer Pool.

<b>Mögliche Werte</b>	0	Wenn BPCSIZE=0 gesetzt ist, wird kein Buffer Pool Cache verwendet.
	100 - 2097148 (2 GB - 4 KB)	Größe des Buffer Pool Cache in KB für Datenbereich-Cache, d.h., bei C64=OFF.  <b>Anmerkung:</b> Der angegebene Wert wird auf den nächsten 4 KB-Grenzwert gerundet.
	100 - 58720256 (56 GB)	Größe des Buffer Pool Cache in KB für Cache oberhalb der Grenze, d.h. bei C64=ON.  <b>Anmerkung:</b> Der angegebene Wert wird auf den nächsten 1 MB-Grenzwert gerundet.
<b>Standard-Einstellung</b>	0	Standardmäßig wird kein Buffer Pool Cache verwendet.
<b>Dynamische Angabe</b>	ja	Dieser Parameter kann nur dynamisch angegeben werden.
<b>Angabe innerhalb der Session</b>	nein	



## Anmerkungen:

1. Dieser Natural-Profilparameter gilt nur unter z/OS (nicht bei Com-plete und nicht bei IMS/TM).
2. Er entspricht dem Subparameter CSIZE des Profilparameters BPI bzw. des Makros NTBPI.
3. Intern wird die BPCSIZE-Angabe in die entsprechende BPI-Angabe umgewandelt; zum Beispiel: BPCSIZE=4000 wird umgewandelt in BPI=(TYPE=NAT, SEQ=0, CSIZE=4000).
4. Falls der angegebene Wert größer als der mögliche Maximalwert ist, wird stattdessen der mögliche Maximalwert verwendet.

5. Der Parameter `BPCSIZE` gilt nur für den primären Natural Buffer Pool (`TYPE=NAT`, `SEQ=0`). Im Falle eines globalen Buffer Pool wird er ignoriert. Bei Vorhandensein eines primären Natural Buffer Pool mit `SEQ=0` im Natural-Parametermodul wird nur die Cache-Größeneinstellung dieses Buffer Pool aktualisiert.
6. In einer Umgebung mit mehreren Benutzern (z.B. unter CICS) betrifft der Profilparameter `BPCSIZE` nur die Natural-Session, die als erste den lokalen Buffer Pool initialisiert.
7. Die für den Buffer Pool Cache zu verwendende Speicherart wird mit dem Profilparameter `BPC64` oder dem Subparameter `C64` des Profilparameters `BPI` bzw. des Makros `NTBPI` angegeben.
8. Weitere Informationen siehe *Buffer Pool Cache* in der *Operations*-Dokumentation.

# 20

## BPI - Buffer Pool-Initialisierung

---

- BPI-Parameter-Syntax ..... 66
- NTBPI-Makro-Syntax ..... 67
- Schlüsselwort-Subparameter ..... 67
- Beispiele für BPI-Parameter ..... 72
- Beispiele für NTBPI-Makros ..... 73

Dieser Natural-Profilparameter wird verwendet, um Buffer Pools zu einer Natural-Session zuzuweisen. Er entspricht dem Makro `NTBPI` im Natural-Parametermodul.

<b>Mögliche Werte</b>	Siehe <a href="#">BPI-Parameter-Syntax</a>	
<b>Standard-Einstellung</b>	Siehe <a href="#">Schlüsselwort-Subparameter</a> .	
<b>Dynamische Angabe</b>	ja	Der Parameter BPI kann nur dynamisch angegeben werden. Im Natural-Parametermodul wird das Makro <code>NTBPI</code> verwendet.
<b>Angabe innerhalb der Session</b>	nein	



#### Anmerkungen:

1. Es gibt mehrere Buffer Pools für verschiedene Zwecke. Es besteht die Möglichkeit, Backup Buffer Pools zu definieren (siehe [Beispiele](#) weiter unten).
2. Falls ein Buffer Pool nicht verfügbar ist, versucht Natural einen Buffer Pool desselben Typs mit der nächst höheren Folgennummer anzulegen.

Dieses Dokument behandelt folgende Themen:

## BPI-Parameter-Syntax

Der Profilparameter BPI wird wie folgt angegeben:

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

Oder Sie lassen `SEQ` weg, um alle Buffer Pool-Definitionen für einen bestimmten Typ zu löschen:

```
BPI=(TYPE=value,OFF)
```



#### Anmerkungen:

1. `BPI=(TYPE=NAT,OFF)` bewirkt, dass alle Buffer Pool-Definitionen für Natural gelöscht werden; d.h., für den Natural Buffer Pool werden die Standardwerte benutzt. Falls der Wert `OFF` verwendet wird, muss er nach `TYPE` und `SEQ` angegeben werden.
2. Wenn Sie den Profilparameter BPI benutzen, um eine vorhandene Buffer Pool-Definition im Parametermodul zu überschreiben, müssen Sie bei *allen* zu ändernden Subparametern neue Einstellungen angeben; andernfalls werden weiterhin die alten Einstellungen benutzt.
3. Wenn Sie z.B. von einem globalen auf einen lokalen Buffer Pool wechseln wollen, müssen Sie angeben: `NAME=' '`.

4. Wenn Sie den Profilparameter BPI benutzen, um dynamisch eine neue Backup Buffer Pool-Definition hinzuzufügen, müssen Sie dafür eine Folgenummer (SEQ) angeben. Falls Sie die SEQ-Angabe weglassen, wird die Definition des primären Buffer Pool (SEQ=0) überschrieben.
5. Die Angaben zu NAME, SIZE, LIST, TXTSIZE, CSIZE, METHOD und C64 für den primären Buffer Pool (SEQ=0) können auch dynamisch mit den Profilparametern BPNAME, BPSIZE, BPLIST, BPTXT, BPCSIZE, BPMETH und BPC64 definiert werden.

## NTBPI-Makro-Syntax

Das Makro NTBPI wird wie folgt angegeben:

```

NTBPI C64=value, *
      CSIZE=value, *
      LIST=value, *
      METHOD=value, *
      NAME=value, *
      SEQ=value, *
      SIZE=value, *
      TXTSIZE=value, *
      TYPE=value

```



**Anmerkung:** Der Wert OFF, der in der Syntax des Profilparameters BPI zur Verfügung steht, ist beim Makro NTBPI nicht erlaubt.

## Schlüsselwort-Subparameter

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



**Anmerkung:** Die Subparameter SIZE, CSIZE, TXTSIZE, METHOD und C64 gelten nicht für globale Buffer Pools. Diese Subparameter werden nur bei der Session berücksichtigt, die als erste einen lokalen Buffer Pool initialisiert.

### C64 - Typ des Buffer Pool Cache-Speichers

C64=*value* bestimmt die Speicherart, die für den Buffer Pool Cache verwendet werden soll.

Wert	Erklärung
ON	<p>Gibt an, dass ein Speicherobjekt oberhalb der Grenze (d.h. in 64-Bit-Speicher) für den Buffer Pool Cache verwendet werden soll.</p> <p>Beachten Sie, dass C64=ON nur dann akzeptiert wird, wenn die folgenden Voraussetzungen erfüllt werden:</p> <ul style="list-style-type: none"> <li>■ Hardware mit z/..-Architektur,</li> <li>■ Betriebssystem z/OS Version 1.2 oder höher</li> </ul> <p>Falls diese Voraussetzungen <i>nicht</i> erfüllt werden, wird die Standard-Einstellung genommen.</p>
OFF	<p>Gibt an, dass ein Datenspeicherbereich für den Buffer Pool Cache verwendet werden soll.</p> <p>Dies ist der Standardwert.</p>



**Anmerkungen:**

1. Gilt nur bei lokalen Buffer Pools des Typs TYPE=NAT unter z/OS (nicht bei Complete).
2. Ein Buffer Pool Cache kann nur dann benutzt werden, wenn der BPI-Subparameter CSIZE oder der Profilparameter BPCSIZE nicht auf Null gesetzt sind.
3. Die C64-Angabe kann dynamisch mit dem Profilparameter BPC64 überschrieben werden.

**CSIZE - Größe des lokalen Buffer Pool Cache**

CSIZE=value bestimmt die Größe des Buffer Pool Cache.

Wert	Erklärung
0	<p>Wenn BPCSIZE=0 gesetzt ist, wird kein Buffer Pool Cache verwendet.</p> <p>Dies ist der Standardwert.</p>
100 - 2097148 (2 GB - 4 KB)	<p>Größe des Buffer Pool Cache in KB für Datenbereich-Cache, d.h. bei C64=OFF.</p> <p><b>Anmerkung:</b> Der angegebene Wert wird auf den nächsten 4 KB-Grenzwert gerundet.</p>
100 - 58720256 (56 GB)	<p>Größe des Buffer Pool Cache in KB für Cache oberhalb der Grenze, d.h. bei C64=ON.</p> <p><b>Anmerkung:</b> Der angegebene Wert wird auf den nächsten 1 MB-Grenzwert gerundet.</p>



**Anmerkungen:**

1. Falls der angegebene Wert größer ist als der mögliche Maximalwert, wird stattdessen der mögliche Maximalwert genommen.
2. Die CSIZE-Angabe gilt nur bei lokalen Buffer Pools des Typs TYPE=NAT (nicht bei Complete).
3. Die CSIZE-Angabe kann dynamisch mit dem Profilparameter BPCSIZE überschrieben werden (nur bei TYPE=NAT).

4. Zum Festlegen des Speichertyps für den Buffer Pool Cache kann der Subparameter **C64** verwendet werden.
5. Weitere Informationen siehe Buffer Pool Cache in der *Operations*-Dokumentation.

### LIST - Name der zu verwendenden Preload-Liste

LIST=*value* bestimmt den Namen der Preload-Liste.

Wert	Erklärung
1 - 8 Zeichen oder ' ' (leer)	Der Name der Preload-Liste, die für diesen Buffer Pool verwendet werden soll. Bei LIST=' ' (leer) wird keine Preload-Liste verwendet.
' ' (leer)	Dies ist der Standardwert.



#### Anmerkungen:

1. Dieser Subparameter gilt nur bei lokalen Buffer Pools des Typs **TYPE=NAT**.
2. Die LIST-Angabe kann dynamisch mit dem Profilparameter **BPLIST** überschrieben werden.
3. Allgemeine Informationen zur Preload-Liste siehe *Preload List* in der *Operations*-Dokumentation.
4. Preload-Listen werden mit der Funktion **Preload List Maintenance** des Dienstprogramms **SYSBPM** gepflegt, siehe *Preload-Liste verwalten* in der *Debugger und Dienstprogramme (Utilities)*-Dokumentation.

### METHOD - Suchalgorithmus zur Zuweisung von Speicherplatz im Buffer Pool

METHOD=*value* bestimmt den Algorithmus für die Zuweisung von Speicherplatz im Buffer Pool.

Wert	Erklärung
S	Gibt an, dass für die Speicherzuordnung ein Auswahlverfahren verwendet werden soll. Dieses Verfahren besteht darin, dass das gesamte Verzeichnis des Buffer Pools durchsucht wird und verschiedene Einträge miteinander verglichen werden, um den geeignetesten Eintrag zu finden. Dieses Verfahren war früher bekannt als „Algorithmus 1+2“.  Dies ist der Standardwert.
N	Gibt an, dass der nächste verfügbare unbenutzte oder freie Speicherplatz verwendet werden soll. Die Suche nach dem nächsten verfügbaren Speicherplatz erfolgt ausgehend von einem Zeiger auf Verzeichniseinträge, der sich zyklisch umlaufend bewegt. Dieses Verfahren kann in Verbindung mit einem Buffer Pool Cache verwendet werden.



#### Anmerkungen:

1. Dieser Subparameter gilt für lokale Buffer Pools mit **TYPE=NAT**.
2. Die METHOD-Angabe kann dynamisch mit dem Profilparameter **BPMETH** überschrieben werden.

3. Weitere Informationen siehe *Buffer Pool Search Methods* in der *Operations*-Dokumentation.

### NAME - Name des globalen Buffer Pool

NAME=*value* bestimmt den Namen des globalen Buffer Pool.

Wert	Erklärung
1 - 8 Zeichen oder '' (leer)	Der Name des globalen Buffer Pool..
' ' (leer)	Dies ist der Standardwert (kein Name angegeben). Es wird ein lokaler Natural Buffer Pool benutzt.



#### Anmerkungen:

1. Dieser Subparameter gilt nur für globale Buffer Pools und für Pools des Typs `TYPE=SWAP` unter CICS.
2. Bei einem lokalen Buffer Pool bleibt *value* leer.
3. Unter IMS TM: Da eine Natural-Session in verschiedenen Regionen ausgeführt werden kann, ist kein lokaler Editor Buffer Pool möglich, sondern nur ein globaler Editor Buffer Pool.
4. Die Angabe im Subparameter NAME kann dynamisch mit dem Profilparameter `BPNAME` überschrieben werden (nur bei `TYPE=NAT`).

### SEQ - Buffer Pool-Folgenummer

SEQ=*value* bestimmt die Folgenummer des Buffer Pool.

Wert	Erklärung
0 oder 1 - 9	Die Folgenummer des Buffer Pool.
0	Dies ist der Standardwert.



#### Anmerkungen:

1. Der mit der niedrigsten Folgenummer definierte Buffer Pool wird als primärer Buffer Pool bezeichnet.
2. Zu jedem Buffer Pool-Typ können Sie einen primären Buffer Pool und einen oder mehrere Backup Buffer Pools (Reserve-Buffer Pools desselben Typs, aber mit anderer Folgenummer) definieren, die benutzt werden, wenn der primäre Buffer Pool bei der Session-Initialisierung nicht verfügbar ist oder nicht zugewiesen werden kann.

3. Buffer Pools desselben Typs werden in der Reihenfolge ihrer Folgenummern sortiert. Falls zwei Buffer Pools desselben Typs dieselbe Folgenummer haben, werden sie in der Reihenfolge sortiert, in der sie angegeben sind.
4. Falls der angeforderte Buffer Pool nicht verfügbar ist, wird stattdessen der Buffer Pool desselben Typs mit der nächst höheren Folgenummer benutzt. Wenn dieser auch nicht verfügbar ist, wird der mit der nächst höheren Folgenummer benutzt usw.

### SIZE - Größe des Buffer Pool

SIZE=*value* bestimmt die Größe des Buffer Pool.

Wert	Erklärung
256 - 2097151	Buffer Pool-Größe in KB bei Natural Buffer Pools.
100 - 2097151	Buffer Pool-Größe in KB bei den anderen Buffer Pool-Typen.
256	Dies ist der Standardwert.



#### Anmerkungen:

1. Dieser Subparameter gilt nur bei lokalen Buffer Pools.
2. Die SIZE-Angabe kann dynamisch mit dem Profilparameter **BPSIZE** überschrieben werden (nur bei TYPE=NAT).

### TXTSIZE - Größe der Buffer Pool Text-Segmente

TXTSIZE=*value* bestimmt die Größe der Segmente, in die der Text-Pool-Bereich des Natural Buffer Pool unterteilt wird.

Wert	Erklärung
1, 2, 4, 8, 12 oder 16	Größe der Buffer Pool-Textsegmente in KB.
4	Dies ist der Standardwert.



#### Anmerkungen:

1. Dieser Subparameter gilt für lokale Buffer Pools mit TYPE=NAT, TYPE=SORT und TYPE=DLI.
2. In Umgebungen mit mehreren Benutzern (z.B. unter CICS) betrifft die TXTSIZE-Angabe nur die Natural-Session, die als erste den lokalen Buffer Pool initialisiert.
3. Bei TYPE=NAT kann die TXTSIZE-Angabe kann dynamisch mit dem Profilparameter **BPTXT** überschrieben werden.

## TYPE - Buffer Pool-Typ

TYPE=*value* bestimmt den Buffer Pool-Typ.

Wert	Erklärung
NAT	Natural Buffer Pool.  Dies ist der Standardwert.
DLI	DL/I Buffer Pool; siehe <i>Control Blocks in Separate Buffer Pool</i> in der <i>Natural for DL/I</i> -Dokumentation.
EDIT	Software AG Editor Buffer Pool; siehe <i>Editor Buffer Pool</i> in the <i>Operations</i> -Dokumentation.  Alternativ kann ein Editor-Hilfs-Buffer-Pool pro Session definiert werden; siehe Profilparameter <a href="#">EDPSIZE</a> .
SORT	Sortier-Buffer Pool; siehe auch Schlüsselwort-Subparameter <a href="#">STORAGE</a> beim Profilparameter SORT bzw. beim Macro NTSORT.
MON	Buffer Pool für die Natural Monitoring-Funktion (SYSMON) der SYSTP-Utility; siehe <i>Natural Monitoring (SYSMON)</i> in der <i>Utilities</i> -Dokumentation.
MSG	Message Buffer Pool; siehe <i>Message Buffer Pool</i> in der <i>Operations</i> -Dokumentation.  <b>Anmerkung:</b> Bei diesem Buffer Pool-Typ wird nur der Subparameter <a href="#">NAME</a> ausgewertet.



### Anmerkungen:

1. Buffer Pools der Typen NAT, DLI oder SORT können mit der Buffer Pool Management Utility SYSBPM verwaltet werden, siehe *Debugger und Dienstprogramme (Utilities)*-Dokumentation.
2. Allgemeine Informationen zum Natural Buffer Pool siehe *Natural Buffer Pools* in der *Operations*-Dokumentation.

## Beispiele für BPI-Parameter

### Beispiel 1:

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

Der primäre Buffer Pool wird durch einen lokalen Buffer Pool mit 2000 KB ersetzt. Die Definition ist gleichbedeutend mit:

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

**Beispiel 2:**

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

Zunächst versucht Natural, einen globalen Buffer Pool mit dem Namen „LBP1“ zuzuweisen. Wird dieser Buffer Pool nicht gefunden, versucht Natural „LBP2“ zuzuweisen. Wenn dieser nicht gefunden wird, weist Natural einen lokalen Buffer Pool mit einer Größe von 500 KB zu.

**Beispiel 3:**

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

Zunächst versucht Natural, einen globalen Buffer Pool mit dem Namen „LBPE1“ zuzuweisen. Wird dieser nicht gefunden, weist Natural einen lokalen Editor Buffer Pool mit einer Größe von 500 KB zu.

**Beispiel 4:**

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

Ein lokaler Natural-Swap Pool mit dem Namen „SWAPPOOL“ mit einer Größe von 500 KB und einer Cache-Größe von 2000 KB wird zugewiesen.

**Beispiele für NTBPI-Makros**

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

In diesen Beispielen werden mehrere Natural Buffer Pools definiert. Falls der globale Buffer Pool NATBP1 nicht verfügbar ist, wird stattdessen der globale Buffer Pool NATBP2 benutzt. Falls letzterer auch nicht verfügbar ist, wird ein lokaler Buffer Pool mit einer Größe von 1000 KB verwendet.



# 21 BPLIST - Name der Preload-Liste für den Natural Buffer Pool

Dieser Natural-Profilparameter gibt den Namen der Preload-Liste an, die für den Natural Buffer Pool verwendet werden soll.

<b>Mögliche Werte</b>	1 - 8 Zeichen oder ' ' (leer)	Name einer Preload-Liste, die für den Natural Buffer Pool verwendet werden soll.  Wenn BPLIST=' ' (leer) angegeben wird, dann wird keine Preload-Liste verwendet.
<b>Standard-Einstellung</b>	leer	
<b>Dynamische Angabe</b>	ja	Dieser Parameter kann nur dynamisch angegeben werden.
<b>Angabe innerhalb der Session</b>	nein	



## Anmerkungen:

1. Dieser Natural-Profilparameter Parameter entspricht der **LIST**-Angabe im Profilparameter **BPI** bzw. im Makro **NTBPI**.
2. Intern wird die **BPLIST**-Angabe in eine gleichwertige **BPI**-Angabe umgewandelt; zum Beispiel: **BPLIST=LIST3** wird umgewandelt in **BPI=(TYPE=NAT, SEQ=0, LIST=LIST3)**.
3. Der Parameter gilt nur für den primären Natural Buffer Pool (**TYPE=NAT, SEQ=0**). Falls im Natural-Parametermodul ein primärer Natural Buffer Pool mit **SEQ=0** vorhanden ist, wird nur die **LIST**-Einstellung dieses Buffer Pool aktualisiert.
4. Allgemeine Informationen zum Natural Buffer Pool siehe *Natural Buffer Pool* in der *Operations*-Dokumentation.



# 22

## BPMETH - Suchalgorithmus zur Zuweisung von Speicherplatz im Buffer Pool

---

Dieser Natural-Profilparameter gibt den für die Zuordnung von Speicherplatz im Natural Buffer Pool zu verwendenden Algorithmus an.

<b>Mögliche Werte</b>	S	Gibt an, dass für die Speicherzuweisung ein Auswahlverfahren verwendet werden soll. Dieses Verfahren besteht darin, dass das gesamte Verzeichnis des Buffer Pools durchsucht wird und verschiedene Einträge miteinander verglichen werden, um den geeignetsten Eintrag zu finden. Dieses Verfahren war früher bekannt als Algorithmus 1+2.
	N	Gibt an, dass der nächste verfügbare unbenutzte oder freie Speicherplatz verwendet werden soll. Die Suche nach dem nächsten verfügbaren Platz erfolgt ausgehend von einem Zeiger auf Verzeichniseinträge, der sich zyklisch umlaufend bewegt. Dieses Verfahren kann in Verbindung mit einem Buffer Pool Cache verwendet werden.
<b>Standard-Einstellung</b>	S	
<b>Dynamische Angabe</b>	ja	Dieser Parameter kann nur dynamisch angegeben werden.
<b>Angabe innerhalb der Session</b>	nein	



### Anmerkungen:

1. Dieser Natural-Profilparameter entspricht dem Subparameter **METHOD** des Profilparameters **BPI** bzw. des Makros **NTBPI**.
2. Intern wird die **BPMETH**-Angabe in die entsprechende **BPI**-Parameterangabe umgewandelt; zum Beispiel: **BPMETH=S** wird umgewandelt in **BPI=(TYPE=NAT, SEQ=0, METHOD=S)**.
3. Der Profilparameter **BPMETH** gilt nur für den primären Natural Buffer Pool (**TYPE=NAT, SEQ=0**). Im Fall eines globalen Buffer Pool wird er ignoriert. Wenn ein primärer Natural Buffer Pool mit **SEQ=0** im Natural-Parametermodul vorhanden ist, wird nur die **METHOD**-Einstellung für diesen Buffer Pool aktualisiert.

4. In einer Umgebung mit mehreren Benutzern (z.B. unter CICS) betrifft der Profilparameter BPMETH nur die allererste Natural-Session, die den lokalen Buffer Pool initialisiert.
5. Allgemeine Informationen zum Natural Buffer Pool siehe *Natural Buffer Pools* in der *Operations*-Dokumentation.

# 23

## BPNAME - Name des globalen Natural Buffer Pool

---

Dieser Natural-Profilparameter gibt den Namen des globalen Natural Buffer Pool an.

<b>Mögliche Werte</b>	1 - 8 Zeichen oder ' ' (leer)	Name des globalen Natural Buffer Pool.  Wenn BPNAME=' ' (leer) gesetzt ist, wird ein <i>lokaler</i> Natural Buffer Pool verwendet.
<b>Standard-Einstellung</b>	leer	
<b>Dynamische Angabe</b>	ja	Dieser Parameter kann nur dynamisch angegeben werden.
<b>Angabe innerhalb der Session</b>	nein	



### Anmerkungen:

1. Der Profilparameter BPNAME kann nur dynamisch angegeben werden. Er entspricht der NAME-Angabe des Profilparameters BPI bzw. des Makros NTBPI.
2. Intern wird die BPNAME-Angabe in die entsprechende BPI-Parameterangabe umgewandelt; zum Beispiel: BPNAME=GBP1 wird umgewandelt in BPI=(TYPE=NAT, SEQ=0, NAME=GBP1) .
3. Der Profilparameter BPNAME gilt nur für den primären Natural Buffer Pool (TYPE=NAT, SEQ=0). Wenn ein primärer Natural Buffer Pool mit SEQ=0 im Natural-Parametermodul vorhanden ist, wird nur die NAME-Einstellung für diesen Buffer Pool aktualisiert.
4. Allgemeine Informationen siehe *Natural Global Buffer Pool* in der *Operations*-Dokumentation.



# 24

## BPPROP - Global Buffer Pool-Weiterleitung

Dieser Natural-Profilparameter steuert die Weitergabe von Änderungen an ein Objekt in einem Buffer Pool. Wenn eine Änderung erfolgt, die ein Objekt betrifft, das in einem (globalen oder lokalen) Buffer Pool gespeichert ist, kann diese Änderung an andere globale Buffer Pools weitergeleitet werden, wodurch die Konsistenz der Buffer Pool gewährleistet wird.

<b>Mögliche Werte</b>	OFF	Änderung werden an keinen anderen globalen Buffer Pool weitergeleitet. <b>Anmerkung:</b> Unter z/OS: Bei allen Einstellungen außer OFF muss der Authorized Services Manager aktiv sein.
	GLOBAL	Änderung werden an alle anderen globalen Buffer Pools weitergeleitet. <b>Anmerkung:</b> In einer z/OS Parallel Sysplex-Umgebung: Die Änderungen werden nur innerhalb des aktuellen z/OS Image weitergeleitet (s. <b>Anm. *</b> ).
	PLEX	Änderung werden innerhalb der gesamten z/OS Parallel Sysplex-Umgebung an alle anderen globalen Buffer Pools mit demselben Namen weitergeleitet (s. <b>Anm. *</b> ).
	GPLEX	Änderung werden innerhalb der gesamten z/OS Parallel Sysplex-Umgebung an alle anderen globalen Buffer Pools weitergeleitet (s. <b>Anm. *</b> ).
<b>Standard-Einstellung</b>	OFF	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	



### Anmerkungen:

1. Dieser Natural-Profilparameter gilt nur unter z/OS.
2. \* Unter z/OS ist die Weiterleitung immer eingeschränkt auf das Natural-Subsystem, in dem die Änderung erfolgt ist; d.h., der mit dem Parameter BPPROP eingestellte Geltungsbereich für die

Weiterleitung gilt nur innerhalb dieses Subsystems, jedoch nicht für andere Subsysteme. Weitere Informationen siehe *Natural Subsystem* in der *Operations-Dokumentation*.

3. Weitere Informationen zur Weiterleitung siehe *Natural Global Buffer Pools* in der *Operations-Dokumentation*.

# 25

## BPSFI - Objektsuche zuerst im Buffer Pool

Dieser Natural-Profilparameter bestimmt die Reihenfolge, in der nach einem angeforderten Objekt, das ausgeführt werden soll, im Buffer Pool und in der (oder den) Systemdatei(en) gesucht werden soll.

Sie können zwischen drei Suchreihenfolgen wählen:

Mögliche Werte		
	ON	<p>Suchreihenfolge 1 wird verwendet: Die Suche erfolgt zuerst im Buffer Pool bei allen Libraries, danach in der (oder den) Systemdatei(en).</p> <p>Natural sucht solange in der folgenden Reihenfolge nach dem Objekt, bis es gefunden wird:</p> <ol style="list-style-type: none"><li>1. Im Buffer Pool, zuerst in der aktuellen Library, danach sukzessive in den Steblibs, dann in den beiden SYSTEM Libraries;</li><li>2. in der (oder den) Systemdatei(en), zuerst in der aktuellen Library, danach sukzessive in den Steblibs, dann in den beiden SYSTEM Libraries.</li></ol> <p>Aus Performance-Gründen wird empfohlen, in einer Produktionsumgebung die Einstellung BPSFI=ON zu wählen.</p> <p><b>Vorsicht:</b> Wenn Sie die Einstellung BPSFI=ON wählen, müssen Sie sich vergewissern, dass die Objektnamen in allen Libraries, die in die Suche einbezogen werden, eindeutig sind. Falls Objekte mit demselben Namen in verschiedenen von der Suche betroffenen Libraries existieren, können unvorhersehbare Ergebnisse auftreten.</p>
	OFF	<p>Suchreihenfolge 2 wird verwendet (bei jeder Datei abwechselnde Suche im Buffer Pool und in der (oder den) Systemdatei(en)).</p> <p>Natural sucht solange in der folgenden Reihenfolge nach dem Objekt, bis es gefunden wird:</p> <ol style="list-style-type: none"><li>1. in der aktuellen Library, zuerst im Buffer Pool, danach in der (oder den) Systemdatei(en);</li></ol>

		<p>2. sukzessive in den Steblibs, zuerst im Buffer Pool, danach in der (oder den) Systemdatei(en) für jede Steplib;</p> <p>3. in den beiden SYSTEM Libraries, zuerst im Buffer Pool, danach in der (oder den) Systemdatei(en) für jede Library.</p> <p>Die Einstellung BPSFI=OFF wird für Entwicklungsumgebungen empfohlen, damit Sie immer das aktuellste Objekt aus Ihrer aktuellen Library zu erhalten.</p>
	LIB	<p>Suchreihenfolge 3 wird verwendet.</p> <p>Natural sucht solange in der folgenden Reihenfolge nach dem Objekt, bis es gefunden wird:</p> <ol style="list-style-type: none"> <li>1. in der aktuellen Library, zuerst im Buffer Pool, danach in der (oder den) Systemdatei(en);</li> <li>2. im Buffer Pool sukzessive in den Steblibs, dann in den beiden SYSTEM Libraries;</li> <li>3. in der (oder den) Systemdatei(en) sukzessive in den Steblibs, dann in den beiden SYSTEM Libraries.</li> </ol>
<b>Standard-Einstellung</b>	OFF	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	



**Anmerkungen:**

1. Weitere Informationen siehe *Steplib Libraries* und *Suchreihenfolge bei der Objekt-Ausführung* in der *Natural benutzen*-Dokumentation.
2. Siehe auch Profilparameter [BP82](#) (*Buffer Pool-Platzhalter wenn Objekt nicht gefunden*).

# 26

## BPSIZE - Größe des lokalen Natural Buffer Pool

---

Dieser Natural-Profilparameter gibt die Größe des lokalen Natural Buffer Pool an.

<b>Mögliche Werte</b>	256 - 2097151	Größe des lokalen Natural Buffer Pool in KB.
<b>Standard-Einstellung</b>	256	
<b>Dynamische Angabe</b>	ja	Dieser Parameter kann nur dynamisch angegeben werden.
<b>Angabe innerhalb der Session</b>	nein	



### Anmerkungen:

1. Dieser Natural-Profilparameter entspricht dem Subparameter **SIZE** des Profilparameters **BPI** bzw. des Makros **NTBPI**.
2. Intern wird die **BPSIZE**-Angabe in eine gleichwertige **BPI**-Angabe umgewandelt; zum Beispiel: **BPSIZE=1500** wird umgewandelt in **BPI=(TYPE=NAT, SEQ=0, SIZE=1500)**.
3. Der Parameter **BPSIZE** gilt nur für den primären Natural Buffer Pool (**TYPE=NAT, SEQ=0**). Bei einem globalen Buffer Pool wird er ignoriert. Falls im Natural-Parametermodul ein primärer Natural Buffer Pool mit **SEQ=0** vorhanden ist, wird nur die **SIZE**-Einstellung dieses Buffer Pool aktualisiert.
4. In einer Umgebung mit mehreren Benutzern (z.B. unter **CICS**) betrifft der Profilparameter **BPSIZE** nur die Natural-Session, die als erste den lokalen Buffer Pool initialisiert.
5. Unter **Com-plete** wird die Größe eines lokalen Natural Buffer Pool gemäß der Anleitung in der Natural *Installation*-Dokumentation eingestellt.
6. Allgemeine Informationen zum Natural Buffer Pool siehe *Natural Buffer Pool* in der *Operations*-Dokumentation.



# 27

## BPTTEXT - Größe der Textsegmente im Natural Buffer Pool

---

Dieser Natural-Profilparameter gibt die Größe der Segmente an, in die der Text-Pool-Bereich des Natural Buffer Pool unterteilt wird.

<b>Mögliche Werte</b>	1, 2, 4, 8, 12 oder 16	Größe der Segmente in KB.
<b>Standard-Einstellung</b>	4	
<b>Dynamische Angabe</b>	ja	Dieser Parameter kann nur dynamisch angegeben werden.
<b>Angabe innerhalb der Session</b>	nein	



### Anmerkungen:

1. Dieser Natural-Profilparameter entspricht dem Subparameter `TXTSIZE` des Profilparameters `BPI` bzw. des Makros `NTBPI`.
2. Intern wird die `BPTTEXT`-Angabe in eine gleichwertige `BPI`-Angabe umgewandelt; zum Beispiel: `BPTTEXT=4` wird umgewandelt in `BPI=(TYPE=NAT,SEQ=0,TXTSIZE=4)`.
3. Der Parameter `BPTTEXT` gilt nur für den primären Natural Buffer Pool (`TYPE=NAT, SEQ=0`). Bei einem globalen Buffer Pool wird er ignoriert. Falls im Natural-Parametermodul ein primärer Natural Buffer Pool mit `SEQ=0` vorhanden ist, wird nur die `TXTSIZE`-Einstellung dieses Buffer Pool aktualisiert.
4. In einer Umgebung mit mehreren Benutzern (z.B. unter CICS) betrifft der Profilparameter `BPTTEXT` nur die Natural-Session, die als erste den lokalen Buffer Pool initialisiert.
5. Allgemeine Informationen zum Natural Buffer Pool siehe *Natural Buffer Pools* in der *Operations*-Dokumentation.



# 28

## BSIZE - Größe des EntireX Broker Buffer

---

Dieser Natural-Profilparameter gibt die Größe des EntireX Broker Buffer an.

<b>Mögliche Werte</b>	0 - 64	Buffer-Größe in KB.
<b>Standard-Einstellung</b>	0	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	



### Anmerkungen:

1. Dieser Natural-Profilparameter gilt nur, wenn der EntireX Broker installiert ist.
2. Alternativ können Sie den äquivalenten Natural-Profilparameter [DS](#) oder das Parametermakro [NTDS](#) benutzen.
3. Zurzeit gibt der EntireX Broker, falls er verwendet wird, die Buffer-Größe automatisch an.



# 29

## BX - Feldumrahmung (Box-Definition)

Mit diesem Parameter bestimmen Sie, welcher Teil einer Feldumrahmung angezeigt werden soll.



### Anmerkungen:

1. „Outlining (Boxing)“ bezeichnet die Möglichkeit, bestimmte Felder auf dem Bildschirm eingeraht anzuzeigen. Diese Form der Anzeige ist eine zusätzliche 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 Terminaltyp kein Outlining unterstützt, wird dieser Parameter zur Ausführungszeit ignoriert.
2. Siehe auch Terminalkommando %D=.

<b>Mögliche Werte</b>	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=TBRL.
	OFF	Bewirkt, dass keine Kästchen um die betreffenden Felder gezogen werden.
<b>Standard-Einstellung</b>	OFF	
<b>Gültige Statements</b>	FORMAT	
	DISPLAY INPUT WRITE	Parameter können auf Statement-Ebene und/oder Element- Ebene angegeben werden.
<b>Gültiges Kommando</b>	Keines.	



### Anmerkungen:

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)
```

# 30

## CANCEL - Abbruch der Session mit Dump

---

Mit diesem Natural-Profilparameter können Sie eine Zeichenkette angeben, die dazu führt, dass eine Natural-Session mit einem Dump beendet wird. Das kann für Debugging-Zwecke hilfreich sein.

<b>Mögliche Werte</b>	1 - 8 Zeichen	Wenn Sie in der Natural-Session diese Zeichenkette in einem beliebigen Eingabefeld (ab dem ersten Eingabefeld) angeben, wird die Session sofort beendet, und es wird ein Dump erzeugt.
<b>Standard-Einstellung</b>	*CANCEL	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	



# 31 CC - 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.

<b>Mögliche Werte</b>	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, das nächste Programm (oder Kommando) im Eingabedatenstrom auszuführen.	
<b>Standard-Einstellung</b>	OFF		
<b>Dynamische Spezifikation</b>	ja		
<b>Spezifikation in Session</b>	ja	Gültige Statements:	SET GLOBALS
		Gültiges Kommando:	GLOBALS
<b>Programmierschnittstelle (API)</b>	USR1005N	Siehe <i>SYSEXT Utility</i> in der Dokumentation <i>Debugger und Dienstprogramme</i> .	



## Anmerkungen:

1. Dieser Profil- und Session-Parameter gilt nicht, wenn vom Benutzer geschriebene Fehlerbehandlungsroutinen benutzt werden.
2. Innerhalb einer Natural-Session kann der Profilparameter CC vom Session-Parameter CC überschrieben werden.
3. 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).



# 32

## CCHAR - Ausgabe-Steuerzeichen zulassen

Um Fehler bei der Bildschirm-Ein-/Ausgabe zu vermeiden, setzt Natural die Ausgabe-Steuerzeichen x'01' bis x'3F' automatisch auf '?' um. Es gibt jedoch Fälle, in denen bestimmte Steuerzeichen für spezielle Zwecke benötigt werden.

Mit diesem Natural-Profilparameter können Sie definieren, dass hexadezimale Ausgabe-Steuerzeichen für die primäre Ein-/Ausgabe unverändert durchgelassen werden. Dadurch werden die Definitionen, die in den Ausgabe-Umsetzungstabellen `NTTAB`, `NTTAB1` und `NTTABL` im Konfigurationsmodul enthalten sind bzw. die durch den entsprechenden dynamischen Profilparameter oder durch das entsprechende Makro im Natural-Parametermodul definiert sind, überschrieben. Damit wird vermieden, dass während der Session-Initialisierung die Fehlermeldung NAT7021 erscheint, wenn der Profilparameter `CFICU` auf ON gesetzt ist.

<b>Mögliche Werte</b>	<i>a1</i> ( <i>a1</i> , <i>a2</i> , ...)	Angegeben werden kann ein einzelnes Hexadezimalzeichen oder eine Liste mit Hexadezimalzeichen in Klammern. Die Hexadezimalzeichen müssen im Bereich x'01' bis x'3F' liegen. Anstelle eines Hexadezimalzeichens ist auch ein Bereich von Hexadezimalzeichen zulässig.
	OFF	Mit OFF werden eventuelle frühere CCHAR-Definitionen zurückgesetzt.
<b>Standard-Einstellung</b>	nein	
<b>Dynamische Spezifikation</b>	ja	
<b>Spezifikation in Session</b>	nein	



**Anmerkung:** Für zusätzliche Druckdateien können Sie den Subparameter `CCHAR` des Profilparameters `PRINT` benutzen.

**Beispiele:**

```
CCHAR=17  
CCHAR=19-1B  
CCHAR=(03-06,0A,1B,3A-3F)
```

# 33

## CCTAB - Drucker-ESC-Folge-Definition

---

■ CCTAB-Parameter-Syntax .....	100
■ NTCCTAB-Makro-Syntax .....	102
■ Syntax der Zeichenketten bei OPN, CLS, CODE, CS, CSS bzw. CSE .....	102
■ Proportionale Schriftarten .....	103
■ Beispiele für CCTAB-Parameter .....	103
■ Beispiele für NTCCTAB-Makros .....	103

Dieser Natural-Profilparameter dient dazu, eine Tabelle mit Druckersteuerzeichenfolgen (Escape Sequences) einzurichten, die zum Ausdrucken von zusätzlichen Berichten und gedruckten Maschinenausgaben verwendet wird. Er entspricht dem Makro `NTCCTAB` im Natural-Parametermodul.

- Es besteht die Möglichkeit, entweder Natural-Feldattribute in ESC-Folgen umzusetzen oder Sonderzeichen zu definieren, die in ESC-Folgen umgesetzt werden.
- Darüber hinaus können Zeichenketten definiert werden, die immer als erster Ausgabedatensatz nach einer Open-Operation oder als letzter Ausgabedatensatz vor einer Close-Operation gesendet werden.
- Das bedeutet, dass Sie durch Benutzung des richtigen Profilnamens Ihren Ausdruck entweder im Hoch- oder im Querformat-Modus aktivieren können. Dann können Sie alle Druckmerkmale des Geräts benutzen, indem Sie in Natural einfach Attribute verwenden. Dadurch haben Sie die Möglichkeit, Ausdrücke von Balkencodes oder mit doppelter Zeichengröße zu erstellen.
- Der Profilparameter `CCTAB` definiert Tabellen, die zum Erkennen von Sonderzeichen in Ausgabefeldern und zum Ersetzen dieser Sonderzeichen durch die definierten Steuerzeichenfolgen verwendet werden.

<b>Mögliche Werte</b>	Siehe <a href="#">CCTAB Parameter-Syntax</a> .	
<b>Standard-Einstellung</b>	Gemäß den Angaben im Makro <code>NTCCTAB</code> in <code>NATCONFIG</code> .	
<b>Dynamische Angabe</b>	ja	Dieser Parameter kann nur dynamisch angegeben werden. Im Natural-Parametermodul wird stattdessen das Makro <code>NTCCTAB</code> verwendet.
<b>Angabe innerhalb der Session</b>	nein	

Dieses Dokument behandelt folgende Themen:

## CCTAB-Parameter-Syntax

Zu jedem Profil muss ein separater `CCTAB`-Parameter angegeben werden. Für die Angabe des Parameters `CCTAB` sind drei Varianten möglich:

## 1. Variante

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

Dabei ist:

- name* Der Name des Profilformulars, d.h. die erforderliche Definition DEFINE PRINTER (*n*) OUTPUT 'nnnnn' PROFILE 'name', die eine maximale Länge von 8 Bytes hat.
- OPN='xxxxx' Optional, definiert eine Datenzeichenkette (bis zu 250 Bytes), die bei jeder Open-Operation an den Drucker gesendet wird.
- CLS='yyyyy' Optional, definiert eine Datenzeichenkette (bis zu 250 Bytes), die vor jeder Close-Operation an den Drucker gesendet wird.



**Anmerkung:** OPN und CLS können in beliebiger Reihenfolge angegeben werden.

## 2. Variante

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

Dabei ist:

- CODE='n' Ein Zeichen, das von Natural erkannt wird, sobald es in der Ausgabe-Zeichenkette erscheint.
- CS='xxxx' Die Zeichenkette, durch die das mit CODE angegebene Zeichen ersetzt werden soll.



**Anmerkung:** Der Subparameter CS muss nach dem Subparameter CODE stehen.

## 3. Variante

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

Dabei ist:

- ATR='nnnn' Das Natural-interne Feldattribut. Der Name wird mit dem Makro NAMATR definiert.
- CSS='xxxx' Die Zeichenkette (bis zu 20 Bytes), die vor dem Feld eingefügt wird. Die CSS-Angabe ist zwingend erforderlich.
- CSE='yyyy' Die Zeichenkette (bis zu 20 Bytes), die hinter dem Feld eingefügt wird. Die CSE-Angabe ist zwingend erforderlich.



**Anmerkung:** Die Subparameter CSS und CSE müssen nach dem Subparameter ATR stehen.

## NTCCTAB-Makro-Syntax

---

Für die Angabe des Makros NTCCTAB sind drei Varianten möglich:

### 1. Variante

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

Einzelheiten siehe CCTAB Parameter-Syntax, [1. Variante](#).

### 2. Variante

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

Einzelheiten siehe CCTAB Parameter-Syntax, [2. Variante](#).

### 3. Variante

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

Einzelheiten siehe CCTAB Parameter-Syntax, [3. Variante](#).

## Syntax der Zeichenketten bei OPN, CLS, CODE, CS, CSS bzw. CSE

---

Sie können die Zeichenketten entweder als Zeichen (in Hochkommas gesetzt) oder als entsprechende hexadezimale Darstellung der Zeichen (ohne Hochkommas) angeben.

## Proportionale Schriftarten

Wenn Sie proportionale Schriftarten verwenden, sollten Sie sicherstellen, dass Sie auf eine Schriftart mit festem Zeichenabstand zurückschalten, ehe Sie Tabellen verwenden, bei denen Sie eine korrekte Positionierung der Zeichen benötigen.

## Beispiele für CCTAB-Parameter

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

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

## Beispiele für NTCCTAB-Makros

```
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(S0B'
NTCCTAB CODE='>', CS='B(S3B '
NTCCTAB CODE='(', CS=' B(S1S'
NTCCTAB CODE=')', CS='B(S0S '
NTCCTAB ATR=P2UL, CSS=' B&&DD', CSE='B&&D$'
NTCCTAB ATR=P2UL, CSS=405FF1C25084C4, CSE=5FF1C250847C
NTCCTAB ATR=P2ITAL, CSS=' B(S1S', CSE='B(S0S'
NTCCTAB ATR=P1HIGH, CSS=' B(S3B', CSE='B(S0B'
NTCCTAB ATR=P2RVID, CSS=' B(S-3B', CSE='B(S0B'
```



# 34 CD - 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](#) - Attribut-Definition.

<b>Mögliche Werte</b>	BL	blau
	GR	grün
	NE	neutral
	PI	rosa
	RE	rot
	TU	türkis
	YE	gelb
<b>Standard-Einstellung</b>	NE	
<b>Gültige Statements</b>	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.
<b>Gültiges Kommando</b>	Keines	

**Beispiel:**

```
INPUT (CD=RE) #A #B
```

# 35 CDYNAM - Dynamisches Laden von

## Nicht-Natural-Programmen

---

Mit diesem Natural-Profilparameter können Sie das Limit für die Anzahl der Nicht-Natural-Programme, die gleichzeitig während einer Natural-Session geladen werden dürfen, festlegen. Wird dieses Limit erreicht, gibt Natural die Fehlermeldung NAT0920 aus, die anzeigt, dass das angeforderte Nicht-Natural-Programm nicht dynamisch geladen werden kann.

Das Laden von Nicht-Natural-Programmen ist nur wieder möglich, nachdem die zuvor geladenen Nicht-Natural-Programme gelöscht sind. Wann Nicht-Natural-Programme nach dem dynamischen Laden gelöscht werden, wird mit dem Profilparameter **DELETE** festgelegt.

<b>Mögliche Werte</b>	0	Dynamisches Laden von Nicht-Natural-Programmen ist nicht erlaubt.
	1 - 1024	Nicht-Natural-Programme können mindestens bis zu der hier angegebenen Anzahl dynamisch geladen werden.  Die Anzahl der tatsächlich geladenen Programme kann je nach Platz in den internen Programmtabellen größer als die angegebene Anzahl sein.
<b>Standard-Einstellung</b>	5	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	

---

# 36

## CF - 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.

<b>Mögliche Werte</b>	beliebiges Sonderzeichen	Ein Terminalkommando muss mit dem angegebenen Zeichen anfangen. Das mit dem CF-Parameter angegebene Zeichen <ul style="list-style-type: none"> <li>■ darf nicht dasselbe sein wie das mit dem HI-Parameter (Hilfezeichen) oder IA-Parameter (Input-Zuweisungszeichen) angegebene Zeichen.</li> <li>■ sollte nicht dasselbe sein wie das mit dem DC-Parameter (Dezimalzeichen) oder ID-Parameter (Input-Delimiterzeichen) angegebene Zeichen.</li> <li>■ Im Masken-Editor (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.</li> </ul>	
	OFF	Es steht kein Steuerzeichen für Terminalkommandos zur Verfügung. Mit SET CONTROL angegebene Terminalkommandos können jedoch nach wie vor ausgeführt werden.	
<b>Standard-Einstellung</b>	%	Ein Terminalkommando muss mit dem Zeichen % anfangen.	
<b>Dynamische Spezifikation</b>	ja		
<b>Spezifikation in Session</b>	ja	Gültige Statements:	SET GLOBALS
		Gültiges Kommando:	GLOBALS
<b>Programmierschnittstelle (API)</b>	USR0350N, USR1005N *	Siehe <i>SYSEXT Utility</i> in der Dokumentation <i>Debugger und Dienstprogramme</i> .  * Empfohlen.	



**Anmerkungen:**

1. Innerhalb einer Natural-Session kann der Profilparameter CF vom Session-Parameter CF überschrieben werden.
2. Unter Natural Security: Die Einstellung dieses Parameters kann von der *Session Parameters*-Option des Library-Profiles überschrieben werden.

# 37

## CFICU - Unicode- und Codepage-Unterstützung

---

- CFICU-Parameter-Syntax ..... 112
- NTCFICU-Makro-Syntax ..... 113
- Schlüsselwort-Subparameter ..... 113
- Beispiel für CFICU-Parameter ..... 118
- Beispiel für NTCFICU-Makro ..... 118

Dieser Natural-Profilparameter wird benötigt, um die Unicode- und Codepage-Unterstützung für verschiedene Unicode-Einstellungen einzuschalten, z.B. wenn Variablen mit Format U oder das Statement `MOVE ENCODED` benutzt werden sollen. Er entspricht im Natural-Parametermodul dem Makro `NTCFICU`.

<b>Mögliche Werte</b>	Siehe <i>CFICU-Parameter-Syntax</i> .	
<b>Standard-Einstellung</b>	ON oder OFF	Schaltet die Unicode- und Codepage-Unterstützung ein bzw. aus.  Die Standard-Einstellung ist OFF, wenn der Profilparameter CP auf OFF gesetzt ist. Andernfalls ist die Standard-Einstellung ON.
<b>Dynamische Angabe</b>	ja	Der Parameter CFICU kann nur dynamisch angegeben werden. Im Natural-Parametermodul wird das Makro <code>NTCFICU</code> verwendet.
<b>Angabe innerhalb der Session</b>	nein	



#### Anmerkungen:

1. `CFICU=ON` wird zwangsweise gesetzt, wenn der Profilparameter `CP` auf einen anderen Wert als `OFF` gesetzt wird.
2. Weitere Informationen siehe *Natural-Profilparameter und Parameter-Makros in der Unicode- und Codepage-Unterstützung-Dokumentation*.

Dieses Dokument behandelt folgende Themen:

## CFICU-Parameter-Syntax

Der Profilparameter `CFICU` wird wie folgt angegeben:

```
CFICU=(ON,keyword-subparameter=value,keyword-subparameter=value,...)
```

Oder:

```
CFICU=ON
```

Oder:

```
CFICU=(OFF,keyword-subparameter=value,keyword-subparameter=value,...)
```

Oder:

```
CFICU=OFF
```

Siehe [Schlüsselwort-Subparameter](#).

## NTCFICU-Makro-Syntax

Das Makro NTCFICU wird wie folgt angegeben:

```
NTCFICU ON, *
    CNVNORM=value, *
    COLLATE=value, *
    COLNORM=value, *
    CPOPT=value, *
    DATITEM=value, *
    LOCALE=value1_value2, *
    STEPLIB='value'
```

Oder:

```
NTCFICU ON
```

Oder:

```
NTCFICU OFF
```

Siehe [Schlüsselwort-Subparameter](#).

## Schlüsselwort-Subparameter

CNVNORM | COLLATE | COLNORM | CPOPT | DATITEM | LOCALE | STEPLIB

## CNVNORM - Normalisierung vor der Umwandlung

CNVNORM=*value* aktiviert bzw. deaktiviert die Normalisierung vor der Umwandlung.

Wert	Erklärung
ON	Eine Normalisierung vor der Umwandlung findet statt.
OFF	Keine Normalisierung vor der Umwandlung.  <b>Anmerkung:</b> Wenn CNVNORM=OFF ist, kann das Statement MOVE NORMALIZED zum Normalisieren von ausgewählten Zeichenketten verwendet werden.  Dies ist die Standard-Einstellung.



### Anmerkungen:

1. Das deutsche Zeichen „ä“ zum Beispiel kann in Unicode als U+00E4 oder unter Verwendung eines zusammengesetzten Zeichens als U+0061, U+0308 dargestellt werden. Bei der Umwandlung in eine Codepage wird das zusammengesetzte Zeichen „ä“ (U+0061 U+0308) als zwei Codepoints betrachtet und ein Ersatzzeichen erzeugt, wenn U+0308 kein gültiges Zeichen der Ziel-Codepage ist. Bei der Normalisierung vor der Umwandlung wird aus den zusammengesetzten Codepoints U+0061 U+0308 ein Codepoint U+00E4 erzeugt, und die anschließende Umwandlung liefert das „ä“ als Ergebnis.
2. Der Parameter wird immer dann berücksichtigt, wenn eine Umwandlung vom Format U nach Format A erfolgt, zum Beispiel: MOVE U TO A oder DISPLAY U, und das Ausgabegerät eine Terminalemulation ist. Diese zusätzliche Operation benötigt natürlich zusätzlichen Speicher sowie zusätzliche CPU-Zeit.

## COLLATE - Collation Services

COLLATE=*value* bestimmt den zu verwendenden Collation Service.

Wert	Erklärung
ON	Verwendung der Locale ID der ICU Collation Services, um Unicode-Zeichensätze zu vergleichen.  Dies ist die Standard-Einstellung.
OFF	Verwendung des ICU Simple Unicode für den Vergleich.



**Anmerkung:** Als „Collation“ bezeichnet man den Vorgang, Einheiten mit textuellen Informationen in einer bestimmten Sortierfolge zu verarbeiten (alphabetisches Sortieren). Diese Sortierfolge ist üblicherweise spezifisch für eine bestimmte Sprache.

### Beispiele:

- Das Zeichen „Ä“ wird im deutschen Gebietsschema („Locale“) zwischen „A“ und „B“ einsortiert, aber im schwedischen Gebietsschema nach dem „Z“.

- Im Litauischen wird das „y“ zwischen dem „i“ und dem „k“ einsortiert.

## COLNORM - Normalisierungsprüfung der Collation Services

COLNORM=*value* dient zum Ein- bzw. Ausschalten der Normalisierungsprüfung.

Wert	Erklärung
ON	Die Prüfung auf nicht-normalisierten Text erfolgt.
OFF	Es erfolgt keine Prüfung auf nicht-normalisierten Text. Dies ist die Standard-Einstellung.



### Anmerkungen:

1. Als Normalisierung bezeichnet man den Vorgang, alternative Darstellungen von gleichbedeutenden Abfolgen aus Textdaten zu entfernen, um die Daten in eine Form umzuwandeln, die binär auf gleiche Bedeutung verglichen werden kann. Der ICU Collation Service handhabt nicht-normalisierten Text korrekt und erzeugt so dieselben Ergebnisse, wie wenn der Text normalisiert wäre. Dadurch wird bei den meisten Texten, die eine Normalisierung erfordern, eine Performance-Maximierung erreicht. Wenn man bei Unicode-Daten mit Gewissheit davon ausgehen kann, dass sie keinen nicht-normalisierten Text enthalten, kann man sich den zusätzlichen Aufwand für die Normalisierungsprüfung ersparen.
2. Dieser Subparameter wird nur dann berücksichtigt, wenn COLLATE=ON gesetzt ist.

## CPOPT - Schnellere Codepage-Umsetzung

CPOPT=*value* kann zur Optimierung der Umwandlungsleistung angegeben werden.

Wert	Erklärung
ON	Verwendung interner Umwandlungstabellen anstelle von ICU-Funktionen, falls möglich.
OFF	In jedem Fall Verwendung von ICU-Funktionen. Dies ist die Standard-Einstellung.



**Anmerkung:** Standardmäßig erfolgt die Umsetzung vom alphanumerischen in das Unicode-Format und in umgekehrter Richtung durch Aufrufen von Unicode-Funktionen. Bestimmte Codepages bilden Zeichen mit einer 1:1-Beziehung in Unicode ab (per „Mapping“). In diesem Fall kann die Umwandlungsleistung verbessert werden, wenn man statt der ICU-Funktionen interne Umwandlungstabellen benutzt. Siehe *Translation Tables* in der *Operations*-Dokumentation.

## DATITEM - Methode für das Laden von ICU-Datenelementen (Data Items)

DATITEM=*value* kann zur Festlegung der Methode für das Laden von ICU-Datenelementen (Data Items) unter CICS und Com-plete angegeben werden. Weitere Informationen siehe Abschnitt *ICU-Datenelemente* in der *Unicode- und Codepage-Unterstützung*-Dokumentation.

Wert	Erklärung
SVC	Die ICU-Datenelemente (Data Items) werden mit der SVC-Anweisung des Betriebssystems geladen.
NONE	Die ICU-Datenelemente (Data Items) werden mit den vom TP-System zur Verfügung gestellten Funktionen geladen.  Dies ist die Standard-Einstellung.



### Anmerkungen:

1. Wenn Sie DATITEM=NONE unter Com-plete benutzen, müssen Sie in den Startup-Optionen für Ihr Com-plete den Schlüsselwort-Parameter THREAD-ESQA-SIZE=15K (oder eine Größe von mehr als 15 KB) setzen.
2. Wenn Sie DATITEM=NONE unter CICS benutzen, müssen Sie pro ICU-Datenelement (Data Item) einen PPT-Eintrag hinzufügen.

## LOCALE - Locale-Kennung

LOCALE=*value1\_value2* bestimmt die „Locale“-Kennung. (Eine „Locale“ ist ein Satz Einstellungen, der die Gebietsschemaparameter enthält.)

Wert	Erklärung
<i>value1_value2_RRR</i>	<i>value1</i> ist ein 2- oder 3-Byte-Sprachcode mit Kleinbuchstabenzeichen. Bei Angabe in Großbuchstaben wird er automatisch in Kleinbuchstaben umgewandelt.  <i>value2</i> ist ein 2- oder 3-Byte-Regionalcode mit Großbuchstabenzeichen zur Klassifizierung der Sprache.
en_US	Dies ist die Standard-Einstellung.



**Anmerkung:** Die Locale-Kennung wird vom ICU Collation Service verwendet, um sprachabhängige und auch regionsabhängige Merkmale der Sortierfolge zu berücksichtigen. Der Sprachcode der Locale-Kennung richtet sich nach ISO639, und der Regionalcode nach ISO3166.

**Beispiele für Sprachcode- und Regionalcode-Paare:**

en_US	Englisch (Vereinigte Staaten)
en_UK	Englisch (Vereinigtes Königreich)
de_DE	Deutsch (Deutschland)
de_AT	Deutsch (Österreich)
de_CH	Deutsch (Schweiz)
sv_SE	Skandinavisch (Schweden)

**STEPLIB - Name des dynamisch zuzuweisenden Dataset (der die ICU-Datenelemente enthält)**

STEPLIB='dataset.name' kann angegeben werden, um den Namen der ICU Library zu definieren. Siehe ICS 311 im Dokument *Unicode- und Codepage-Unterstützung*.

Wert	Erklärung
'dataset.name'	Der Name des Dataset, der die ICU-Datenelemente enthält. Länge: 3 bis 44 Zeichen.

**Validierungsregeln für den mit dem STEPLIB-Subparameter angegebenen Dataset-Namen**

- Der Dataset muss existieren.
- Der Name des Dataset muss mit Hochkommas ( ' ' ) umschlossen sein, siehe Beispiel weiter oben.
- Der Name des Dataset muss den z/OS-Namenskonventionen entsprechen:
  - Maximale Länge: 44 Zeichen.
  - Darf keine Sonderzeichen enthalten.
  - Darf kein High Level Qualifier sein, d.h., er muss mindestens einen Punkt ( . ) enthalten.

Ab ICS 311 werden von der Software AG zur Verfügung gestellte Datendateien nicht unterstützt und es werden nur die minimalen Lademodule SAGICU und SAGICUA9 ausgeliefert. Diese Module enthalten keine statisch verlinkten Lokalisierungsdaten. ICU-Lokalisierungsdaten werden lediglich dynamisch aus einem Dataset geladen, der die Datenelemente (Collators, Converters usw.) enthält.

Der in dem Parameter CFICU STEPLIB angegebene Dataset wird dynamisch nur einmal durch die erste Natural-Sitzung in einem gegebenen TP-System unter der DD Card ICS<sub>xxx</sub>DD (dabei entspricht xxx der ICS-Version) zugewiesen und danach von allen Natural-Sitzungen benutzt.



**Anmerkung:** Diese Funktionalität ist nur ab ICS 311 verfügbar.

## Beispiel für CFICU-Parameter

---

```
CFICU=(COLNORM=ON,LOCALE='de_DE',STEPLIB='DATAITEMS.LOAD')
```

## Beispiel für NTCFICU-Makro

---

```
NTCFICU COLNORM=ON, *  
        LOCALE=de_DE, *  
        STEPLIB='DATAITEMS.LOAD'
```

# 38

## CFWSIZE (für interne Verwendung)

---

Dieser Parameter ist für die interne Nutzung durch Natural reserviert.



**Vorsicht:** Ändern Sie seine Einstellung nicht.



# 39

## CICSP - Umgebungsparameter für Natural CICS-Interface

---

- NTCICSP-Makro-Syntax ..... 122
- Schlüsselwort-Subparameter ..... 123
- Beispiel für ein NTCICSP-Makro ..... 142

Dieser Natural-Profilparameter kann nur mit dem Makro `NTCICSP` angegeben werden. Die Angabe als dynamischer Parameter ist noch nicht möglich.



**Anmerkung:** Die Verfügbarkeit des Makros `NTCICSP` ist abhängig von der bei Ihnen installierten Version der Natural CICS Schnittstelle.

Das Makro `NTCICSP` dient zur Definition umgebungsspezifischer Parameter für Natural-Session-Optionen, die in einer CICS-Umgebung relevant sind.

<b>Mögliche Werte</b>	Siehe unten.
<b>Standard-Einstellung</b>	Siehe unten.
<b>Dynamische Angabe</b>	nein
<b>Angabe innerhalb der Session</b>	nein

## NTCICSP-Makro-Syntax

Das Makro `NTCICSP` wird wie folgt angegeben:

```

NTCICSP BACKEND=value, *
      BACKOUT=value, *
      BACKRPL=value, *
      CALLRPL=value, *
      CHAP=value, *
      CNTCALL=value, *
      COMARET=value, *
      DIRNAME=value, *
      DUPTID=value, *
      FDTPX=value, *
      LOGDEST=value, *
      MEMOBJR=value, *
      MSGDEST=value, *
      MSGPFX=value, *
      MSGTRAN=value, *
      PREFIX=value, *
      PRMDEST=value, *
      PSTRNID=value, *
      RCVASYN=value, *
      RESENDC=value, *
      RESENDS=value, *
      RJEDEST=value, *
      SLCALL=value, *
      SLNOHLD=value, *
      SNDLAST=value, *
      STORVIO=value, *
      TERMVAR=value, *
      TIOBSZ=value, *

```

TRANCHK= <i>value</i> ,	*
TTYCNLS= <i>value</i> ,	*
UCTRAN= <i>value</i> ,	*
UNITID= <i>value</i> ,	*
USERID= <i>value</i>	

Siehe [Schlüsselwort-Subparameter](#).

## Schlüsselwort-Subparameter

BACKEND | BACKOUT | BACKRPL | CALLRPL | CHAP | CNTCALL | COMARET | DIRNAME | DUPTID | FDTPIX  
 | LOGDEST | MEMOBJR | MSGDEST | MSGPFX | MSGTRAN | PREFIX | PRMDEST | PSTRNID | RCVASYN |  
 RESENDC | RESENDS | RJEDEST | SLCALL | SLNOHLD | SNDLAST | STORVIO | TERMVAR | TIOBSZ | TRANCHK  
 | TTYCNLS | UCTRAN | UNITID | USERID

### BACKEND - Back-End-Programm-Aufrufsteuerung

BACKEND=*value* definiert, ob ein angegebenes Back-End-Programm oder eine angegebene Back-End-Transaktion aufgerufen werden soll, nachdem die Session (normal oder abnormal) beendet worden ist.

Der BACKEND-Parameter hat zwei Subparameter. Der zweite Subparameter ist optional. Er legt fest, ob im Fall eines Terminal-Fehlers ein Back-End-Programm aufgerufen werden soll. Dies schließt auch Session-Bereinigungs-Tasks ein, die von NEP gestartet werden.

Mögliche Werte für beide Subparameter sind ON/OFF, die Standard-Einstellungen sind jedoch verschieden.

Wert	Erklärung
ON	<p>Gleich wie BACKEND=(ON,OFF).</p> <p>Dies ist die Standard-Einstellung, wenn der BACKEND-Parameter weggelassen wird. Ein potenzielles Back-End-Programm oder eine potenzielle Back-End-Transaktion wird immer aufgerufen, insbesondere nach einer abnormalen Beendigung eines Task, jedoch nicht bei Terminal-Fehlern.</p> <p>Wird ein Back-End-Programm aufgerufen, dann werden die Natural-Beendigungsmeldung und ein Rückgabecode an den CICS-Transaktionsarbeitsbereich (TWA) übergeben. Zusätzlich können die gleichen Informationen, wie beim BACKRPL-Parameter beschrieben, an eine CICS COMMAREA übergeben werden.</p>
(ON,ON)	<p>Gleich wie BACKEND=(,ON).</p> <p>Ein potenzielles Back-End-Programm oder eine potenzielle Back-End-Transaktion wird immer aufgerufen, insbesondere nach abnormalen Task-Beendigungen (ABEND). Dies schließt auch Terminal-Fehler ein.</p>

Wert	Erklärung
OFF	Erzwingt die Einstellung <code>BACKEND=(OFF,OFF)</code> .  Ein potenzielles Back-End-Programm oder eine potenzielle Back-End-Transaktion wird nur dann aufgerufen, wenn die Natural-Session normal, d.h. mit einer Natural-Beendigungsmeldung, beendet worden ist.

### BACKOUT - Rücksetzen einer Transaktion bei nicht behebbaren abnormalen Beendigungen

`BACKOUT=value` definiert, ob das Natural CICS Interface eine Transaktionsrücksetzung mittels eines `EXEC CICS SYNCPOINT ROLLBACK`-Aufrufs durchführen soll oder nicht.

Wert	Erklärung
ON	Alle anstehenden Datei-Aktualisierungen werden zurückgesetzt.  Dies ist die Standard-Einstellung.
OFF	Alle anstehenden Datei-Aktualisierungen werden bereitgestellt.

Wegen seines Exits für abnormale Beendigungen fängt das Natural CICS Interface alle abnormalen Beendigungen ab. Falls eine abnormale Beendigung nicht behackbar ist, werden alle Ressourcen für die abnormal beendete Session freigegeben und die Session wird per `EXEC CICS RETURN` beendet, d.h., sie wird im Sinne von CICS „normal“ beendet. Somit werden am Ende des Task anstehende Datei-Aktualisierungen nicht automatisch durch CICS zurückgesetzt.

### BACKRPL - Übergabeort der Parameterliste für das Back-End-Programm

`BACKRPL=value` steuert, wo und wie die Back-End-Parameter an das Back-End-Programm übergeben werden.

Wert	Erklärung
ALL	Dies ist die Standard-Einstellung.  Der durch das Makro <code>NAMBCKP</code> abgebildete Natural-Back-End-Parameterbereich wird sowohl in der CICS TWA als auch in einer CICS COMMAREA übergeben (einschließlich potenzieller Beendigungsdaten).
COMA	Der durch das Makro <code>NAMBCKP</code> abgebildete Natural-Back-End-Parameterbereich wird (einschließlich potenzieller Beendigungsdaten) nur in einer CICS COMMAREA und nicht in der CICS TWA übergeben.
DATA	Der durch das Makro <code>NAMBCKP</code> abgebildete Natural-Back-End-Parameterbereich wird nur in der CICS TWA übergeben. Eine CICS COMMAREA hält lediglich potenzielle Beendigungsdaten fest. Sind keine Beendigungsdaten verfügbar, erfolgt keine COMMAREA-Übergabe.
TWA	Der durch das Makro <code>NAMBCKP</code> abgebildete Natural-Back-End-Parameterbereich wird nur in der CICS TWA übergeben. Es erfolgt keine COMMAREA-Übergabe.



**Anmerkung:** Dieser Parameter gilt nur für Back-End-Programme, nicht für Back-End-Transaktionen.

## CALLRPL - Übergabeort der Parameterliste für externe Subroutinen-Programme, aufgerufen von Natural per EXEC CICS LINK

CALLRPL=*value* steuert, wo und wie die CALL-Parameterlisten an externe Subroutinen-Programme übergeben werden.

Wert	Erklärung
ALL	Dies ist die Standard-Einstellung.  Die Natural-Parameterlistenadressen werden sowohl in der CICS TWA als auch in einer CICS COMMAREA übergeben. Die Länge der übergebenen COMMAREA wird durch den zweiten Subparameter gesteuert.
COMA	Die Natural-Parameterlistenadressen werden nur in einer CICS COMMAREA und nicht in der CICS TWA übergeben. Die Länge der übergebenen COMMAREA wird durch den zweiten Subparameter gesteuert.
TWA	Die Natural-Parameterlistenadressen werden nur in der CICS TWA und nicht in einer CICS COMMAREA übergeben, d.h., die Länge der COMMAREA ist dann 0.

Mögliche Werte für den zweiten Subparameter:

Wert	Erklärung
2	Dies ist die Standard-Einstellung.  Nur die Adresse der Parameteradressenliste und die Adresse der Feldbeschreibungsliste (R1 und R2, wie bei CALL-Statement beschrieben) werden in einer CICS COMMAREA übergeben, d.h., die Länge der COMMAREA ist dann 8.
3	Die Adresse der Feldlängenliste (R3, wie beim CALL-Statement beschrieben) wird zusätzlich in einer COMMAREA übergeben, d.h., die Länge der COMMAREA ist 12.
4	Die Adresse der Feldlängenliste und die Adresse der Liste der großen Feldlängen (R4, wie bei CALL-Statement beschrieben) werden zusätzlich in einer CICS COMMAREA übergeben, d.h., die Länge der COMMAREA ist dann 16.

Beispiel:

```
CALLRPL=(ALL,2)
```

Dies ist die Standard-Einstellung.



**Anmerkungen:**

1. Der zweite Subparameter gilt nur, wenn der erste Subparameter auf ALL oder COMA gesetzt ist.
2. Wenn die CICS TWA benutzt werden soll, enthält diese immer alle vier Parameterlistenadressen.

3. Wenn die Länge der CICS COMMAREA größer als 0 ist, erhält die zuletzt übergebene Parameteradresse eine Markierung, die besagt, dass es sich um die letzte Adresse in der Liste handelt. Diese Markierung wird im höchstwertigen Bit im Adressfeld gesetzt.
4. Der CALLRPL-Parameter gilt nicht, wenn Parameterwerte in einer CICS COMMAREA übergeben werden (%P=C). Dann wird unabhängig von der Einstellung des CALLRPL-Parameters eine CICS COMMAREA benutzt.

### CHAP - Vorrang der Task-Zuteilung ändern

CHAP=*value* definiert, wie das Natural CICS Interface lang laufende Tasks behandeln soll, die die Aufruf-Höchstgrenzen von DBROLL und/oder MAXROLL erreichen.

Wert	Erklärung
ON	Jedes Mal, wenn der Task die mit DBROLL und/oder MAXROLL festgelegten Aufruf-Höchstgrenzen erreicht, wird die Zuteilungspriorität für den Task um eins verringert. Die ursprüngliche Task-Zuteilungspriorität wird bei der nächsten Bildschirm-Ein-/Ausgabe wiederhergestellt.
OFF	Die Session wird unterbrochen.  Dies ist die Standard-Einstellung.

### CNTCALL - CICS-Aufruf übergibt Daten automatisch im Transportbehälter

Mit SET CONTROL 'P=C' werden statt Parameterdatenzeiger die CALL-Statement-Parameterdaten beim EXEC CICS LINK in einem CICS COMMAREA übergeben. Da die Länge einer CICS COMMAREA auf 32 KB begrenzt ist, wird EXEC CICS LINK bei einer COMMAREA, die größer als 32 KB ist, aufgrund einer LENGERR-Bedingung fehlschlagen.

CNTCALL=*value* gestattet es Ihnen, bei EXEC CICS LINK einen Transportbehälter zu benutzen, wenn die zu übergebenden Daten die maximale COMMAREA-Länge von 32 KB überschreiten. Diese Funktionalität funktioniert nur, wenn der CICS Transaction Server in Ihrer z/OS-Umgebung Kanäle (Channels) und Transportbehälter (Containers) unterstützt.

Standardmäßig lautet dann der Name des Transportbehälters NCI - COMMAREA, sofern nicht vor dem Natural-CALL-Statement ausdrücklich ein anderer Name über die Anwendungsprogrammierschnittstelle USR4204N angegeben wird.

Wert	Erklärung
ON	Wenn die COMMAREA-Daten 32 KB überschreiten würden, verwendet das Natural CICS Interface automatisch einen CICS-Transportbehälter für das EXEC CICS LINK-Kommando und benutzt standardmäßig NCI - COMMAREA als Namen.
OFF	Wenn die COMMAREA-Daten 32 KB überschreiten würden, schlägt die Ausführung des Natural-CALL-Statement mit Fehlermeldung NAT0920 und Ursachencode LENGERR (hexadezimal 16) fehl.

## COMARET - CICS COMMAREA-Nutzung zur Task-Steuerung

COMARET=*value* definiert, ob das Natural CICS Interface beim Beenden und Neustarten von pseudo-konversationellen Tasks die Vorteile der Kommandoebene der CICS-COMMAREA nutzt.

Wert	Erklärung
ON	Ein pseudo-konversationeller Task speichert seine Neustart-Informationen in einer CICS COMMAREA, sofern er nicht per EXEC CICS LINK aufgerufen worden ist.  Dies ist die Standard-Einstellung.
OFF	Zwingt Natural, seine Neustart-Informationen in einen temporären CICS-Hauptspeicher zu stellen, was Mehraufwand zur Folge hat, der durch zusätzliche Dienstaufrufe bedingt ist, die zum Abstellen und Zurückholen dieser Informationen benötigt werden.  Der Schlüssel für den temporären CICS-Speicher besteht aus einer Präfix-Zeichenkette (Definition erfolgt mit dem NCMDIR-Parameter TSKEY, siehe <i>TP Monitor Interfaces</i> -Dokumentation) und aus der Terminal-Kennung. Bei Betrieb in einer CICSplex-Umgebung muss der Schlüssel für den temporären CICS-Speicher in einem CICS TST als REMOTE/SHARED definiert werden, damit aus allen teilnehmenden CICS-Regionen auf ihn zugegriffen werden kann.

## DIRNAME - Name des Natural CICS Interface-Systemverzeichnismoduls

DIRNAME=*value* gibt den Namen des Natural CICS Interface-Systemverzeichnismoduls an.

Wert	Erklärung
(siehe unten)	Ein beliebiger, gültiger Modulname.
<i>prefixCB</i>	<i>prefix</i> ist das gemeinsame Präfix für Programme und Dateien, siehe <a href="#">PREFIX</a> -Parameter.  Dies ist die Standard-Einstellung.

Die ersten 5 Zeichen des Verzeichnisnamens werden auch als Teil der temporären CICS-Speicher-Warteschlangennamen verwendet, die sich auf die relevante NCI-Umgebung beziehen. Daher müssen sich die Namen der relevanten Systemverzeichnismodule in den ersten fünf Zeichen unterscheiden, wenn mehr als eine CICS-Umgebung in einer CICS-Region betrieben wird.

Bitte beachten Sie, dass der angegebene oder standardmäßig verwendete Name des Natural CICS Interface-Systemverzeichnismoduls zur Laufzeit über das Exit-Interface NCIDIREX (für den Namen des Natural CICS Interface-Systemverzeichnismoduls) geändert werden kann (siehe *TP Monitor Interfaces*-Dokumentation). Dies ermöglicht es, dasselbe NCI-Treiber-/Natural-Parametermodul zu benutzen, aber verschiedene NCI-Umgebungen (Thread-Gruppen, Tread-Größen usw) in Abhängigkeit z.B. von CICS-Systemkennung, Transaktionskennung zu benutzen.

## DUPTID - Umgang mit doppelten Terminal-Kennungen

Das Natural CICS Interface erfordert eindeutige Terminal-Kennungen, weil die Terminal-Kennung der Schlüssel zu seinen Session-Informationsdatensätzen (SIRs) ist. Normalerweise ist dies in einer einzelnen CICS-Region gewährleistet, jedoch nicht notwendigerweise über mehrere CICS-Regionen hinweg, die gemeinsam denselben SIP Server benutzen.

`DUPTID=value` bestimmt, wie das Natural CICS Interface mit doppelten Terminal-Kennungen umgeht, d.h., wenn eine neue Session gestartet werden soll und für diese Terminal-Kennung schon ein SIP existiert.

Wert	Erklärung
ON	Bei Feststellung einer doppelten Terminal-Kennung erzwingt das Natural CICS Interface intern die Beendigung der alten Session und startet danach eine neue Session.  Dies ist die Standard-Einstellung.
OFF	Wenn für die Terminal-Kennung der neuen Session schon ein SIR existiert, beendet das Natural CICS Interface die neue Session und gibt die Meldung NS19 aus. Erläuterungen und Abhilfemaßnahmen siehe <i>Natural under CICS Messages, SCP Processing Errors</i> in der <i>Messages and Codes</i> -Dokumentation.

Es steht ein Terminal-Kennungs-Exit-Interface zur Verfügung, um eindeutige, aus 8 Zeichen bestehende Terminal-Kennungen zu erstellen, z.B. durch Anhängen der 4 Zeichen langen CICS-Systemkennung an die 4 Zeichen lange CICS-Terminal-Kennung, was eine logische Natural-Terminal-Kennung ergibt.

## FDTPX - Verwendung von NCIDTPEX Exit für alle Terminal-Typen erzwingen

`FDTPX=value` bestimmt, ob das Terminal-Ein-/Ausgabe-Interface NCIDTPEX (siehe *TP Monitor Interfaces*-Dokumentation) für alle in Ihrer Umgebung benutzten Terminal-Typen aufgerufen wird.

Wert	Erklärung
ON	Das Interface NCIDTPEX wird bei allen Terminal-Typen aufgerufen.
OFF	Das Interface NCIDTPEX wird nur für verteilte Transaktionsverarbeitung (DTP) mit APPC- oder MRO-Umwandlungen aufgerufen.  Dies ist die Standard-Einstellung.

## LOGDEST - Bestimmungsort für Natural-CICS-Protokollierung

LOGDEST=*value* gibt den Namen des CICS-Bestimmungsorts an, wohin das Natural CICS Interface seine Session-Protokoll-Datensätze schreibt.

Wert	Erklärung
<i>name</i>	Ein beliebiger, gültiger Bestimmungsortname.
NLOG	Dies ist die Standard-Einstellung.

Ein Eintrag in die CICS-Destinations-Steuerungstabelle muss für den optionalen Natural-CICS-Protokoll-Dataset definiert werden.

## MEMOBJR - Bestimmungsort in Speicherobjekt, falls möglich

In älteren CICS TS-Versionen werden temporäre Hauptspeicher-Warteschlangen oberhalb der Grenze zugeordnet. Bei Benutzung von temporärem CICS-Hauptspeicher werden Auslagerungsdaten in temporäre Speicher-Warteschlangen-Elemente mit maximaler Größe von jeweils 32 KB aufgespalten.

Abhängig von der CICS TS-Version, die an Ihrem Standort installiert ist, unterstützt CICS auch die Verwendung von Speicherobjekten für CICS-Anwendungen. Die Zuordnung von CICS-Speicherobjekten für Auslagerungsdaten bedeutet weniger zusätzlichen Aufwand, weil die Auslagerungsdaten nicht aufgespalten werden müssen.

Mögliche Werte für MEMOBJR=*value*:

Wert	Erklärung
ON	Daten werden unter Verwendung von CICS-Speicherobjekten ausgelagert, falls dies durch die an Ihrem Standort installierte CICS-Version unterstützt wird.  Dies ist die Standard-Einstellung.
OFF	Daten werden in den temporären CICS-Hauptspeicher ausgelagert.

## MSGDEST - Bestimmungsort-Kennung für Natural-Fehlermeldung-Protokollierung

MSGDEST=*value* gibt den Namen des CICS-Bestimmungsorts an, der vom Natural CICS Interface für informative NCI-Nachrichten und zum Protokollieren der Natural-Session-Beendigungsmeldung im Falle einer abnormalen Beendigung der Session benutzt werden soll.

Wert	Erklärung
<i>name</i>	Ein beliebiger, gültiger Bestimmungsortname.
NERR	Dies ist die Standard-Einstellung.

Da diese Nachrichten bzw. Meldungen in Zeichenformat vorliegen, können Sie einen beliebigen, schon zur Verfügung stehenden CICS-Bestimmungsort (z.B. CSSL) benutzen, anstatt einen neuen Bestimmungsort zu definieren.

### MSGPFX - NCI-Meldungspräfix für WTL-Meldungen generieren

Das Natural CICS Interface verwendet ein Präfix für alle Meldungen, die es an den Bestimmungsort MSGDEST sendet. Das Präfix hat eine Länge von ca. 48 Bytes und enthält folgende Informationen:

- NCI-Meldungsnummer
- CICS-Region-SYSID
- Terminal-Kennung oder die Zeichenkette ASYN bei Nicht-Terminal-Tasks
- Benutzerkennung
- Transaktionskennung
- Datum und Uhrzeit

Das Meldungspräfix wird standardmäßig auch an die Meldungen angehängt, die über CMWTL ausgegeben werden.

Mögliche Werte für `MSGPFX=va lue`:

Wert	Erklärung
ON	Das NCI-Meldungspräfix wird an alle Meldungen angehängt, die über CMWTL ausgegeben werden. Dies ist die Standard-Einstellung.
OFF	Das NCI-Meldungspräfix wird nicht an die Meldungen angehängt, die über CMWTL ausgegeben werden. Die Meldungen werden unverändert ausgegeben.

### MSGTRAN - Interne Transaktionskennung für die Meldungsumschaltung

`MSGTRAN=va lue` gibt die Transaktionskennung an, die intern von den Natural-Funktionen zur Meldungsumschaltung und zum asynchronen Session-Flush-Abschluß benutzt wird.

Dieser Parameter hat dieselbe Bedeutung wie der `MSGTRAN`-Parameter im Modul `NCIZNEP` (siehe Natural *Installation*-Dokumentation). Die Einstellungen beider Parameter müssen identisch sein.

Wert	Erklärung
(siehe unten)	Eine beliebige, gültige CICS-Transaktionskennung.
NMSG	Dies ist die Standard-Einstellung.

Diese Transaktionskennung darf mit keiner zum Aufrufen von Natural benutzten Transaktionskennung identisch sein und muss in CICS definiert sein.

### **PREFIX - Gemeinsames Präfix für Programme und Dateien**

`PREFIX=value` definiert ein gemeinsames Modulpräfix für die Natural CICS-Komponenten, z.B.:

- Natural-CICS-Systemverzeichnis *prefixCB*,
- CICS 3270 Bridge XFAINTU Exit *prefixXFA*,
- VSAM Roll Files *prefixRn*, wobei  $n = 1 - 9$  ist,
- Systemsteuerungsdatensätze im CICS-Hauptzwischenpeicher, in denen von der Natural CICS-Schnittstelle Informationen über alle permanenten GETMAIN-Speicher als lokale Pools und gemeinsam genutzte Threads zwischengespeichert werden.

Die Schlüssel der TS-Steuerungsdatensätze haben die Form *prefixXCR*, wobei *X* ein nicht druckbares Zeichen ist.

Es empfiehlt sich generell, dieses gemeinsame Präfix für alle Programme zu verwenden, die einen Bezug zur Natural CICS-Schnittstelle haben, z.B. *prefixDRV* für das Natural-CICS-Schnittstellen-Modul, *prefixNEP* für das Natural CICS-Schnittstellen-Knotenfehlerprogramm.

Um die Zwischenspeichersteuerungssätze von der automatischen Löschung auszunehmen, dürfen Sie in der jeweiligen TSMODEL-Ressourcendefinition kein Ablauf-Intervall angeben.

Wert	Erklärung
XXXXX	<i>prefix</i> ist ein Name mit 1 bis 5 Zeichen, der die Namenskonventionen für Lademodule und Zwischenspeicherwarteschlangen erfüllt.

Es ist keine Standard-Einstellung vorhanden.

### **PRMDEST - Name des Bestimmungsorts für Natural CICS-Profilparametereingaben**

`PRMDEST=value` gibt den Namen des CICS-Bestimmungsorts an, der dynamische Natural-Profilparameter enthält.

Wert	Erklärung
<i>name</i>	Ein beliebiger, gültiger Name für den Bestimmungsort.
NPRM	Dies ist die Standard-Einstellung.

Während der Systeminitialisierung ruft das Natural CICS Interface dynamische Natural-Profilparameter ab und speichert sie in seiner Umgebung. Beim Start der Session werden potenzielle andere Profilparameter (die per Terminal-Eingabe oder durch einen Front-End-Aufrufer eingegeben wurden) am Ende der Parameter-Zeichenkette verkettet, die vom Bestimmungsort PRMDEST abgerufen wurden, d.h., explizit dynamische Profilparameter können benutzt werden, um diejenigen Natural CICS Interface-System-Profilparameter zu überschreiben, die vom Bestimmungsort PRMDEST gelesen wurden.

Für den optionalen Natural CICS Interface-Profilparameter-Eingabe-Bestimmungsort muss ein Eintrag in der CICS-Bestimmungsort-Tabelle definiert werden, normalerweise ein zusätzlicher Partitionsbestimmungsort.

### PSTRNID - Steuerung der \*INIT-PROGRAM-Variableneinstellung

Wenn ein Natural-Task durch ein Front-Ende-Programm aktiviert wird, bestimmt der Parameter PSTRNID, wie die Natural-Systemvariable \*INIT-PROGRAM gesetzt wird.

Mögliche Werte für PSTRNID=*value*:

Wert	Erklärung
ON	*INIT-PROGRAM wird auf die tatsächliche Transaktionskennung gesetzt, die für die pseudo-konversationelle Natural CICS-Task-Verarbeitung benutzt wird, welche nicht zwangsläufig die Transaktionskennung des Task ist, der ursprünglich die Natural-Sesion gestartet hat.  Dies ist die Standard-Einstellung.
OFF	*INIT-PROGRAM wird auf die Transaktionskennung des Task gesetzt, der ursprünglich die Natural-Sesion gestartet hat.

### RCVASYN - Asynchrone Session wiederherstellen

RCVASYN=*value* definiert, wie das Natural CICS Interface asynchrone Sessions behandelt.

Wert	Erklärung
ON	Dies ist die Standard-Einstellung.  Das Natural CICS Interface erzwingt einige Natural-Profilparameter-Einstellungen für Nicht-Terminal-Sessions, um eine unerwartete Eingabe oder abnormale Beendigungen aufgrund von Fehlern NT06, NT11 oder anderen Ein-/Ausgabebefehlen zu verhindern.  RCVASYN=ON erzwingt folgende Parametereinstellungen:

Wert	Erklärung
	<ul style="list-style-type: none"> <li>■ CM=OFF , MENU=OFF , PC=OFF</li> <li>■ TTYPE=ASYL, wenn die SENDER-Einstellung leer, nicht angegeben oder eine CICS-Übergangsdaten-Warteschlange ist, oder wenn CONSOLE angegeben ist (siehe <i>Asynchronous Natural Sessions under CICS</i> in der <i>TP Monitor Interfaces</i>-Dokumentation).</li> <li>■ SENDER='msgdest', wenn die SENDER-Angabe leer oder nicht angegeben ist.</li> <li>■ OUTDEST='sender', wenn die OUTDEST-Angabe leer oder nicht angegeben ist.</li> <li>■ INTENS=1 , EJ=OFF, wenn die SENDER-Angabe CONSOLE oder eine CICS-Übergangsdaten-Warteschlange ist, die nicht für Druckersteuerzeichen eingerichtet ist.</li> </ul>
OFF	Das Natural CICS Interface unternimmt nichts Spezielles bei Nicht-Terminal-Sessions. Es liegt in der Verantwortung des Benutzers, angemessene Natural-Profilparameter für eine asynchrone Natural-Session zu setzen, siehe <i>Asynchronous Natural Processing</i> .

### RESENDC - Prüfen auf erneutes Bildschirm-Senden nach Subroutinen-Aufruf

Natural optimiert den 3270-Ausgabedatenstrom standardmäßig. Die von Natural benutzte Bildschirm-Abbildungstechnik ermöglicht es Natural, sich immer an die zuletzt gesendete Maske zu „erinnern“. Somit sendet Natural, wenn eine neue Maske gesendet wird, tatsächlich nur „Aktualisierungen“ der alten Maske. Bei dieser Logik ist es möglich, dass ein Bildschirm-Abbild durch 3GL-Programme zerstört werden kann, die Bildschirm-Ein- und Ausgaben selbst durchführen.

Mögliche Werte für RESENDC=*value*

Wert	Erklärung
ON	Das Natural CICS Interface prüft, ob irgendwelche 3GL-Programme Bildschirm-Ein- und Ausgaben durchgeführt haben. Falls ja, veranlasst das Natural CICS Interface, dass Natural bei der nächsten Bildschirm-Ein- und Ausgabe einen Vollbildschirm sendet.  Dies ist die Standard-Einstellung.
OFF	Das Natural CICS Interface veranlasst, dass Natural nur Aktualisierungen sendet.

### RESENDS - Erneutes Bildschirm-Senden nach Fortführung einer pseudo-konversationellen Session

Natural optimiert den 3270-Ausgabedatenstrom standardmäßig. Die von Natural benutzte Bildschirm-Abbildungstechnik ermöglicht es Natural, sich immer an die zuletzt gesendete Maske zu „erinnern“. Somit sendet Natural, wenn eine neue Maske gesendet wird, tatsächlich auch nur „Aktualisierungen“. Bei dieser Logik ist es möglich, dass ein Bildschirm-Abbild durch 3GL-Programme zerstört werden kann, z.B. durch Nachrichtenumschaltung (CICS CMSG-Transaktion) während einer pseudo-konversationellen Bildschirm-Ein- und Ausgabe.

Mögliche Werte für RESENDS=*value*:

Wert	Erklärung
ON	Das Natural CICS Interface erkennt während der Natural-Session auch Bildschirm-Ein- und Ausgaben von Außen und veranlasst, dass Natural den zuletzt ausgegebenen Bildschirm erneut sendet.  Dies ist die Standard-Einstellung.
OFF	Beim Senden einer neuen Maske sendet Natural nur „Aktualisierungen“.

### RJEDEST - Name des Natural CICS-Jobstart-Bestimmungsorts

RJEDEST=*value* betrifft nur z/OS-Betriebssysteme.

Wert	Erklärung
(siehe unten)	Name des Bestimmungsorts.
NRJE	Dies ist die Standard-Einstellung.

RJEDEST gibt den *Bestimmungsortnamen* für den zusätzlichen CICS-Partitions-Bestimmungsort an, der von der Utility NATRJE zum Starten von Jobs über die JES-interne Reader Facility benutzt wird.

 **Vorsicht:** In der CICS DCT und Start-JCL muss ein geeigneter CICS-Bestimmungsort definiert sein, siehe auch den entsprechenden Schritt in *Installing Natural CICS Interface on z/OS* in der Natural *Installation*-Dokumentation.

Der Funktionscode L oder B (*parm3* des NATRJE CALL-Statement) muss für den letzten NATRJE-Aufruf gesetzt sein.

Wenn L angegeben wird und *nrje* ein zusätzlicher Partitions-Bestimmungsort ist, wird der Bestimmungsort geschlossen, was wiederum den Start des internen Reader auslöst.

Wenn B angegeben wird und *nrje* ein indirekter Bestimmungsort ist, wird der Bestimmungsort nicht geschlossen. In diesem Fall muss eine vorangehende /\*EOF-Karte gestartet werden, um den Start des internen Reader auszulösen.

Weitere Informationen zur Natural Utility NATRJE siehe Natural *Utilities*-Dokumentation.

### SLCALL - Standard-Verbindungsaufruf (Standard Linkage Call)

Das Natural-Statement CALL ruft unter Verwendung von CICS-Konventionen, d.h. über ein EXEC CICS LINK, ein dynamisches Nicht-Natural-Programm auf. Ein dynamisches Nicht-Natural-Programm kann auch mit Standard-Programmverbindungskonventionen (z.B. BALR/BASR/BASSM 14, 15) aufgerufen werden, wenn ein entsprechender Anzeiger in dem Natural-Programm gesetzt ist, bevor das CALL-Statement ausgeführt wird; siehe auch Terminalkommandos %P=S, %P=SC, %P=L und %P=LS.

 **Vorsicht:** Die Terminalkommandos %P=S, %P=SC, %P=L und %P=LS umgehen den SLCALL-Automatismus der automatischen Benutzung einer bestimmten Verbindungskonvention.

SLCALL=*value* gestattet Ihnen die automatische Benutzung einer bestimmten Verbindungskonvention.

Wert	Erklärung
ON	Das Natural CICS Interface bestimmt, ob das aufzurufende Modul ein gültiges CICS-Kommandoebenen-Programm ist, indem es am Ladepunkt des Moduls nach der Zeichenkette DFH sucht. Wird DFH gefunden, dann wird das Programm über ein EXEC CICS LINK aufgerufen. Wird DFH nicht gefunden, dann wird das Modul gemäß den Standard-Verbindungskonventionen behandelt und über ein BALR/BASSM 14, 15 aufgerufen.
OFF	Die Verbindungskonvention wird nicht benutzt. Dies ist die Standard-Einstellung.

### SLNOHLD - Lade-Option für externe, über Standard-Verbindungskonventionen aufzurufende Programme

SLNOHLD=*value* definiert, wie das Natural CICS Interface externe, über Standard-Verbindungskonventionen aufzurufende Nicht-LE-Programme (d.h., dynamische Nicht-CICS-Programme und RCA-Programme) in einer Nicht- CICSplex-Umgebung behandelt.

Wert	Erklärung
ON	Dies ist die Standard-Einstellung.  Das Natural CICS Interface lädt alle externen Nicht-LE-Programme, die über Standard-Verbindungskonventionen (einschließlich RCA-Programme) über ein EXEC CICS LOAD ohne die HOLD-Option aufgerufen werden sollen, und gestattet somit, dass diese Programme mit NEWCOPY neu kopiert werden, während die Session in einer pseudo-konversationellen Bildschirm-Ein-/Ausgabe unterbrochen ist bzw. wartet.  SLNOHLD=ON entspricht der Verarbeitung, die das Natural CICS Interface gewöhnlich für LE-Programme und für Nicht-LE-Programme in einer CICSplex-Umgebung trotzdem durchführt.
OFF	Dies ist die Einstellung, mit der Natural seit jeher gearbeitet hat.  Das Natural CICS Interface lädt alle externen Nicht-LE-Programme, die über Standard-Verbindungskonventionen (einschließlich RCA-Programme) über ein EXEC CICS LOAD HOLD aufgerufen werden sollen, d.h., ein solches Programm wird in Abhängigkeit von der Einstellung des Profilparameters DELETE im Speicher behalten, RCA-Programme bis zum Session-Ende.

### SNDLAST - LAST-Option-Verwendung für EXEC CICS SEND-Kommandos

SNDLAST=*value* ist nützlich bei SNA-Terminals (LUTYPE2) mit Klammer-Protokoll, um bei pseudo-konversationellen Bildschirm-Ein-/Ausgaben die „Ende-Klammer“ zu erzwingen.

Wert	Erklärung
ON	Die Option LAST wird bei EXEC CICS SEND-Kommandos benutzt, bevor der Task im pseudo-konversationellen Modus beendet wird.  Dies ist die Standard-Einstellung.
OFF	Die Option LAST wird nicht benutzt.

## STORVIO - Abfangen von Speicherverletzungen

STORVIO=*value* sorgt für das Abfangen von Speicherverletzungen bei externen Programmaufrufen mit der Aufrufoption %P=C(C).

Wert	Erklärung
OFF	Das Abfangen von Speicherverletzungen ist deaktiviert.  Dies ist die Standard-Einstellung.
( <i>mm, nn</i> )	Das Abfangen von Speicherverletzungen wird durch Angabe eines beliebigen STORVIO-Subparameters aktiviert.  Der erste Subparameter dient zur Angabe eines Toleranzwertes im Bereich von 0 bis 16777215: Die Speichergröße für das zusätzliche %C(C) GETMAIN wird um diesen Wert erhöht, um so zu versuchen, reale CICS-Speicherverletzungen zu verhindern.  Der zweite Subparameter gibt an, welche Reaktion bei Feststellung einer Speicherverletzung erfolgen soll.  Mögliche Werte:
0	Es wird nur eine Speicherverletzungsmeldung NCI0250 ausgegeben und es erfolgt keine weitere besondere Interaktion.  Dies ist der Standardwert für Subparameter.
1 - 32767	Zusätzlich zur Meldung NCI0250 wird eine NAT0920-Bedingung ausgegeben, wobei der angegebene Wert als Ursachencode übergeben wird. Da in CICS der NAT0920-Ursachencode normalerweise den EIBRESP-Wert einer gescheiterten EXEC CICS LOAD- oder LINK-Anforderung enthält, wird empfohlen, <i>keinen</i> Wert im Bereich eines gültigen CICS EIBRESP-Wertes anzugeben, d.h., die Werte 1 bis 255 besser CICS zu überlassen.
32768 oder höher	Zusätzlich zur Meldung NCI0250 wird eine abnormale Beendigung S0C3 erzwungen, die eine NAT0954-Bedingung erzeugt.

## TERMVAR - Terminal-Kennungs-Variable für Natural-Arbeitsdateien

TERMVAR=*value* gestattet es einem Natural-Benutzer, exklusive Natural-Arbeitsdateien unter CICS zu haben, ohne die Terminal-Kennung kennen zu müssen.

Wert	Erklärung
xxxx	Die Variable xxxx ist eine Zeichenkette mit vier Zeichen. Siehe Erläuterung weiter unten.
&TID	Dies ist die Standard-Einstellung.

Da Terminal-Kennungen in CICS eindeutig sein müssen, enthalten exklusive Arbeitsdateien im CICS-Zwischenspeicher normalerweise die CICS-Terminal-Kennung. Der Parameter TERMVAR gestattet es Ihnen, eine Variable zu definieren. Wird diese Variable im Namen einer Arbeitsdatei gefunden, wird sie durch die tatsächliche Terminal-Kennung ersetzt. Zeichenketten mit nicht-alphanumerischen Zeichen müssen in Hochkommas (') eingeschlossen werden.

Beachten Sie bitte, dass die gepackte CICS-Task-Nummer für Nicht-Terminal-Sessions als *logische* Terminal-Kennung benutzt wird.

 **Vorsicht:** Die Variablen-Zeichenkette darf nicht die Teilzeichenfolge '\*\*' enthalten, weil Natural diese Teilzeichenfolge durch die Nummer der Arbeitsdatei ersetzt, was es unmöglich macht, die Terminal-Kennung zu ersetzen.

## TIOBSZ - Größe des Natural-Terminal-Ein-/Ausgabe-Puffers

TIOBSZ=*value* gibt die Größe des Natural-Terminal-Ein-/Ausgabe-Puffers an.

Wert	Erklärung
8 - 60	Größe des Natural-Terminal-Ein-/Ausgabe-Puffers in KB.
16	Dies ist die Standard-Einstellung.

## TRANCHK - Eingabemaske auf Transaktionskennung prüfen

Falls die Verbindung mit einer CICS-Session verloren geht oder getrennt wird (z.B. unter VM or wenn ein Session Manager installiert ist), ohne dass die Session beendet worden ist, kann ein anderer Benutzer beim Aufrufen von CICS in diese offene Session gelangen. Üblicherweise ist die erste Maßnahme eines Benutzers in einer CICS-Umgebung die Eingabe einer Transaktionskennung.

Mögliche Werte für TRANCHK=*value*:

Wert	Erklärung
ON	Das Natural CICS Interface prüft, ob die ersten 4 Bytes der vom Benutzer eingegebenen Transaktionskennung mit der Natural-Transaktionskennung übereinstimmen. Falls ja, unterstellt das Natural CICS Interface einen „Neustart“, nachdem eine Verbindung verloren gegangen oder getrennt worden ist. Alle Ressourcen der alten Session werden freigegeben und es wird eine neue Session gestartet.
OFF	Vom Benutzer eingegebene Daten werden nicht auf die Natural-Transaktionskennung geprüft. Dies ist die Standard-Einstellung.

### TTYCNLSL - Konsolenkommunikation

Dieser Parameter dient der Kompatibilität zu früheren Versionen des Natural CICS Interface. `TTYCNLSL=value` steuert Session- und Geräte-Eigenschaften für Natural-Sessions, die durch ein Konsolengerät unter Verwendung von z.B. dem Kommando `MODIFY` gestartet worden sind.

Wert	Erklärung
ON	Die Systemvariable <code>*DEVICE</code> wird auf <code>TTY</code> gesetzt: Die Kommunikation mit der Konsole erfolgt in einem 3270-Datenstrom, der TTY-Steueranweisungen enthält.  Der Profilparameter <code>PSEUDO</code> wird ausgewertet und erlaubt oder erlaubt nicht, dass die Session im pseudo-konversationellen Modus läuft.  Dies ist die Standard-Einstellung.
OFF	Die Systemvariable <code>*DEVICE</code> wird auf <code>BATCH</code> gesetzt und erzwingt Batch-/Kommandozeilen-Modus: Jede Zeile wird durch ein <code>EXEC CICS WRITE OPERATOR</code> -Kommando separat an die Konsole ausgegeben.  Der Profilparameter <code>PSEUDO</code> wird ignoriert: Die Session läuft im konversationellen Modus, um anzuzeigen, dass eine Session ansteht.

### UCTRAN - Unterstützung von Kleinschreibung/gemischter Schreibweise in Natural

`UCTRAN=value` aktiviert bzw. deaktiviert die Unterstützung von Kleinschreibung/gemischter Schreibweise durch das Natural CICS Interface.

Wert	Erklärung
ON	Gleich wie <code>UCTRAN=(ON, ON)</code> . Die NCI-Unterstützung von Kleinschreibung/gemischter Schreibweise ist vollständig aktiviert.  Dies ist die Standard-Einstellung.
OFF	Gleich wie <code>UCTRAN=(OFF, ON)</code> . Die NCI-Unterstützung von Kleinschreibung/gemischter Schreibweise ist für pseudo-konversationelle Bildschirm-Ein-/Ausgabe deaktiviert.

Der erste Subparameter steuert die NCI-Unterstützung von gemischter Schreibweise nach einer pseudo-konversationellen Bildschirm-Ein-/Ausgabe, während der zweite Subparameter die NCI-

Unterstützung von gemischter Schreibweise nach einer konversationellen Bildschirm-Ein-/Ausgabe unterstützt. Letztere schließt auch das NTC-Hochladen ein.

### Erster Subparameter (pseudo-konversationelle Bildschirm-Ein-/Ausgaben)

Um die Unterstützung von Kleinschreibung/gemischter Schreibweise für pseudo-konversationelle Natural-Sessions zu erreichen, ist es nötig, dass die Terminal-Eingabe nicht schon in Großschreibung umgesetzt wird, bevor der Natural-Nukleus die Kontrolle übernimmt. Darum schaltet das Natural CICS Interface den Modus von Terminals, die mit `UCTRAN(ON)` definiert sind, standardmäßig für die gesamte Dauer der Natural-Session auf gemischte Schreibweise (`UCTRAN(TRANID)`).

Da aus Sicherheitsgründen jegliche Änderung an CICS-Definitionen/Steuerblöcken unerwünscht sein kann, kann man verhindern, dass das Natural CICS Interface die Umsetzung in Großschreibung bei einem Terminal ändert, indem man diesen NTCICSP-Parameter `UCTRAN` auf `OFF` setzt. Wenn das der Fall ist, muss der Benutzer definieren, dass sein Terminal im Kleinschreibungsmodus (CICS `TYPETERM`-Parameter `UCTRAN(TRANID/OFF)`) läuft, um die Unterstützung von Kleinschreibung/gemischter Schreibweise in Natural nutzen zu können.

Da alle von der aktuellen Natural-Version unterstützten CICS-Versionen die Umschaltung der Groß-/Kleinschreibung auf Transaktionsebene durch den `UCTRAN`-Parameter in einem `PROFILE` der Transaktion bieten, sollte dieser NTCICSP-Parameter auf `OFF` gesetzt werden, wodurch die Unterstützung von Kleinschreibung/gemischter Schreibweise CICS überlassen wird.



**Anmerkung:** In CICS bestimmt die Kombination der `UCTRAN`-Parameter sowohl in der `TYPETERM`- als auch in der `PROFILE`-Definition, wie CICS die Terminal-Eingabe einer pseudo-konversationellen Transaktion behandelt (weitere Informationen siehe *CICS Resource Definition Manual* usw.). Deshalb ist es immer ratsam, dass hauptsächlich das mit der Transaktion verbundene `PROFILE` den erforderlichen Status der Umsetzung in Großbuchstaben definiert, um so sicherzustellen, dass irgendwelche `TYPETERM`-Modus-Änderungen hinsichtlich der Umsetzung in Großbuchstaben keine Auswirkung auf eine Anwendung haben.

### Zweiter Subparameter (konversationelle Bildschirm-Ein-/Ausgabe)

Die Unterstützung von Kleinschreibung/gemischter Schreibweise für konversationelle Bildschirm-Ein-/Ausgaben bedeutet, dass das Natural CICS Interface bei den CICS-Terminal-Eingabeanforderungen die `ASIS`-Option (`CONVERSE/RECEIVE ASIS`, d.h. in der vorliegenden Form) verwendet. Wenn der zweite Subparameter auf `OFF` gesetzt ist, erledigt das Natural CICS Interface die konversationellen CICS-Terminal-Eingabeanforderungen ohne die `ASIS`-Option.

## UNITID - Herstellung eindeutiger Terminal-Kennungen

UNITID=*value* hilft, die Terminal-Kennungen für Natural-Zwecke über mehr als eine CICS-Region eindeutig zu machen.

Wert	Erklärung
ON	Das Natural CICS Interface hängt eine CICS-Systemkennung (lokale SYSID, falls kein MRO, sonst TOR SYSID) an die 4 Byte lange CICS-Terminal-Kennung und erzeugt so eine 8 Byte lange, logische Terminal-Kennung.
OFF	Das Natural CICS Interface benutzt die CICS-Terminal-Kennung in der vorliegenden Form. Dies ist die Standard-Einstellung.

Dieser Parameter kann dann von Interesse sein, wenn Ressourcen von mehreren CICS-Regionen gemeinsam, insbesondere bei Nicht-CICSplex, als SIP Server oder Roll Server genutzt werden: Wenn in mehreren CICS-Umgebungen dieselben Terminal-Kennungen benutzt werden, hilft dieser Parameter dabei, für Natural eindeutige Terminal-Kennungen bereit zu stellen. Innerhalb des Natural CICS Interface sind die Natural-Terminal-Kennungen 8 Byte breite Felder, und eine Kombination aus 8-Byte-Terminal-Kennung und 8-Byte-CICS-Benutzerkennung wird als Schlüssel für SIP und den Roll Server genommen.

Das Ergebnis dieses Parameters wird vom Natural CICS Interface für den Session-Schlüssel und den Roll Server-Schlüssel und von Natural für die Systemvariable \*INIT-ID benutzt.



### Anmerkungen:

1. Ein Terminalkennungs-Exit (NCITIDEX) wird möglicherweise diese logische Terminalkennung nachträglich behandeln. Beschreibung von NCITIDEX siehe *TP Monitor Interfaces*-Dokumentation.
2. Außerdem besteht die Möglichkeit, dass ein Benutzerkennungs-Exit (NCIUIDEX und NATUEX1) die Systemvariable \*INIT-ID nachträglich behandelt. Beschreibung von NCIUIDEX siehe *TP Monitor Interfaces*-Dokumentation, NATUEX1 in der *Operations*-Dokumentation.
3. Dieser Parameter betrifft außerdem NAF-Drucker (*Natural Advanced Facilities*), d.h., die Drucker müssen im NAF Spool und Report Management System NATSPOOL entsprechend definiert werden oder es sollte ein Benutzerkennungs-Exit verwendet werden, um die Systemvariable \*INIT-ID für Drucker nachzubehandeln.
4. Für Nicht-Terminal-Sessions richtet das Natural CICS Interface immer eine 8 Byte lange, logische „Terminal-Kennung“ ein, die aus der gepackten CICS-Task-Nummer und der CICS-Systemkennung besteht, d.h., UNITID=ON wird für asynchrone Tasks erzwungen, wobei die CICS-Task-Nummer als Terminal-Kennung genommen wird.

## USERID - Behandlung der CICS-Benutzerkennung

USERID=*value* definiert, wie Natural unter CICS eine CICS-Benutzerkennung für eine Natural-Session behandeln soll.

Der erste Subparameter ist für terminalgebundene CICS-Sessions, der zweite Subparameter für nicht-terminalgebundene, d.h., asynchrone, DPLEd usw. CICS-Sessions. Der dritte Subparameter ist für Programm-zu-Programm-Sessions, d.h., DTP, APPC.

Wert	Erklärung
ANY	Ein beliebiger, von EXEC CICS ASSIGN USERID (..) zurückgegebener, nicht leerer Wert wird als gültig betrachtet.  Dies ist die Standard-Einstellung.
ON	Ein von EXEC CICS ASSIGN USERID (..) zurückgegebener, nicht leerer Wert wird als gültig betrachtet, wenn er nicht mit der CICS-Standard-Benutzerkennung identisch ist, und (nur bei terminalgebundenen Sessions) wenn sich der Benutzer bei CICS angemeldet hat.
OFF	Der von EXEC CICS ASSIGN USERID (..) zurückgegebene Wert wird ignoriert.

### Weiterverarbeitung

Wenn eine CICS-Benutzerkennung ungültig ist oder ignoriert wird, wird die editierte (entpackte) CICS-Tasks-Nummer für Nicht-Terminal-Sessions, d.h. asynchrone, oder DPLEd usw. genommen. Für terminalgebundene CICS-Sessions wird die 3 Byte lange Operator-Kennung genommen, wenn diese nicht leer ist, sonst wird die CICS-Terminal-Kennung genommen. Für DTP Sessions wird die Pseudo-Terminal-Kennung genommen.



#### Anmerkungen:

1. CICS-Terminal-Kennungen sind innerhalb einer CICS-Region eindeutig, wohingegen CICS-Benutzerkennungen und Operator-Kennungen nicht zwangsläufig eindeutig sind. CICS-Terminal-Kennungen können jedoch Duplikate in anderen CICS-Regionen haben, was doppelte Benutzerkennung in Adabas zur Folge hat.
2. Der Natural-Benutzerkennungs-Exit NATUEX1 (siehe *Operations*-Dokumentation) oder das Natural CICS-Benutzerkennungs-Exit-Interface NCIUIDEX (siehe *TP Monitor Interfaces*-Dokumentation) können für die kundenspezifische Anpassung des Inhalts der Systemvariablen \*INIT-USER verwendet werden.

## Beispiel für ein NTCICSP-Makro

---

```
NTCICSP PREFIX=NCI92,      *  
DIRNAME=NCI92,           *  
BACKRPL=COMA,            *  
CALLRPL=COMA
```

# 40

## CLEAR - Verarbeitung der CLEAR-Taste im NEXT-Modus

---

Dieser Natural-Profilparameter veranlasst Natural, immer dann, wenn die CLEAR-Taste während der Programmausführung im NEXT-Modus gedrückt wird, ein spezifisches Natural-Terminalkommando auszuführen.

<b>Mögliche Werte</b>	Beliebiges Zeichen	Die Standard-Aktion kann durch Angabe eines Zeichens aufgehoben werden, das ein gültiges Natural-Terminalkommando bildet, wenn es an das Terminalkommando-Steuerzeichen (gemäß Angabe im Parameter <b>CF</b> ) angehängt wird.
<b>Standard-Einstellung</b>	%	Standardmäßig reagiert Natural beim Drücken der CLEAR-Taste so, als ob der Benutzer das Terminalkommando %% eingegeben hätte.
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	



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

### Beispiel:

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

Natural führt das Terminalkommando %R aus, wenn die CLEAR-Taste im NEXT-Modus gedrückt wird.



# 41 CM - Kommandomodus

---

Dieser Natural-Profilparameter kann benutzt werden, um den Natural-Kommandomodus (NEXT und MORE) zu unterdrücken.

<b>Mögliche Werte</b>	ON	NEXT und MORE stehen für die Kommandoeingabe zur Verfügung.
	OFF	Die Natural-Session wird beendet, wenn NEXT angetroffen wird; die MORE-Zeile ist schreibgeschützt (keine Eingabe möglich).
<b>Standard-Einstellung</b>	ON	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	



# 42

## CMPO - Kompilierungsoptionen

---

- CMPO-Parameter-Syntax ..... 148
- NTCMPO-Macro-Syntax ..... 149
- Schlüsselwort-Subparameter ..... 150
- Beispiel für CMPO-Parameter ..... 150
- Beispiel für NTCMPO-Makro ..... 150

Dieser Natural-Profilparameter kann zu Beginn einer Session benutzt werden, um beim Session-Start dieselben Kompilierungsoptionen anzugeben oder zu überschreiben, die statisch mit dem Makro `NTCMPO` im Natural-Parametermodul oder während einer aktiven Session mit dem Systemkommando `COMPOPT` angegeben werden können.

<b>Mögliche Werte</b>	Siehe <a href="#">CMPO-Parameter-Syntax</a> .	
<b>Standard-Einstellung</b>	Identisch mit den entsprechenden Optionen des Systemkommandos <code>COMPOPT</code> . Siehe <a href="#">Schlüsselwort-Subparameter</a> bzw. <code>COMPOPT</code> -Optionen.	
<b>Dynamische Angabe</b>	ja	Dieser Parameter kann nur dynamisch angegeben werden. Im Natural-Parametermodul wird stattdessen das Makro <code>NTCMPO</code> benutzt.
<b>Angabe innerhalb der Session</b>	ja	Mit Systemkommando <code>COMPOPT</code> .

Dieses Dokument behandelt folgende Themen:

## CMPO-Parameter-Syntax

---

Der Parameter `CMPO` hat folgende Syntax:

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

Siehe [Schlüsselwort-Subparameter](#).

Die Option `DB2PKYU` gilt nur, wenn die in Ihrer Umgebung installierte Version von Natural for Db2 diese Option unterstützt.

 **Anmerkungen:**

1. Die Schlüsselwort-Subparameter sind funktional identisch mit den Kompilierungsoptionen, die innerhalb der Session mit dem Systemkommando `COMPOPT` angegeben werden können.
2. Jeder Subparameter kann den Wert `ON` oder `OFF` annehmen (`GFID` kann außerdem den Wert `VID` annehmen).
3. Weitere Informationen, z.B. Standardwerte und Funktionsbeschreibungen siehe Systemkommando `COMPOPT` in der *Systemkommandos*-Dokumentation.

## NTCMPO-Macro-Syntax

Das Makro NTCMPO im Natural-Parametermodul hat folgende Syntax:

```

NTCMPO  CHKRULE=value,          *
        CPAGE=value,            *
        DB2ARRY=value,         *
        DB2BIN=value,          *
        DB2PKYU=value,        *
        DB2TSTI=value,        *
        DBSHORT=value,       *
        ECHECK=value,         *
        GDASC=value,          *
        GFID=value,           *
        KCHECK=value,         *
        LOWSRCE=value,       *
        *
        MASKCME=value,        *
        MAXPREC=value,        *
        MEMOPT=value,         *
        PCHECK=value,         *
        PSIGNF=value,        *
        THSEP=value,         *
        TQMARK=value,        *
        TSENABL=value

```

Siehe [Schlüsselwort-Subparameter](#).



### Anmerkungen:

1. Die Option DB2PKYU ist nur vorhanden, wenn die in Ihrer Umgebung installierte Version von Natural for Db2 diese Option unterstützt.
2. Die möglichen Schlüsselwort-Subparameter sind funktional identisch mit den Optionen, die innerhalb einer Session mit dem Systemkommando COMPOPT angegeben werden können.
3. Jeder Subparameter kann den Wert ON oder OFF annehmen (GFID kann außerdem den Wert VID annehmen).
4. Im Makro NTCMPO können die Schlüsselwort-Subparameter in beliebiger Reihenfolge angegeben werden.
5. Weitere Informationen, z.B. Standardwerte und Funktionsbeschreibungen, erhalten Sie, wenn Sie den im Syntax-Diagramm aufgeführten Links folgen.

## Schlüsselwort-Subparameter

---

CHKRULE | CPAGE | DB2ARRY | DB2BIN | DB2PKYU | DB2TSTI | DBSHORT | ECHECK | GDASC | GFID |  
KCHECK | LOWSRCE | MASKCME | MAXPREC | MEMOPT | PCHECK | PSIGNF | THSEP | TQMARK | TSENABL

Eine vollständige Beschreibung der Compiler-Optionen finden Sie beim Systemkommando `COMPOPT` in der *Systemkommandos*-Dokumentation. Die dort aufgeführten Standardwerte gelten ebenfalls für die entsprechenden Schlüsselwort-Subparameter von `CMPO` und `NTCMPO`.

## Beispiel für CMPO-Parameter

---

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

## Beispiel für NTCMPO-Makro

---

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

\*

# 43 CMPR - Standard-Algorithmus zur

## Komprimierungsoptimierung

---

Mit diesem Natural-Profilparameter kann der Natural-Administrator den allgemeinen Standard-Algorithmus zur Komprimierungsoptimierung angeben, um Hauptspeicher bei Sessions einzusparen, die zurzeit mit der Verarbeitung beschäftigt sind, und um die Performance von Natural zu verbessern.

Darüber hinaus kann der Natural-Administrator die Art der Komprimierungsoptimierung speziell für einzelne Puffer-Typen angeben, indem er den Parameter `CMPR` des Makros `NTBUFID` im `NATCONFIG`-Modul benutzt. Die Einstellung dieses Parameters hebt die allgemeine Standard-Einstellung des Profilparameters `CMPR` auf. Weitere Informationen siehe *Customization of Buffer Characteristics* in der *Operations*-Dokumentation.

<b>Mögliche Werte</b>	OPT0	Komprimierung ohne Optimierung.
	OPT1	Komprimierung mit Optimierung identischer Zeichen aus dem Puffer, die am unteren und am oberen Ende verwendet werden.
	OPT2	Komprimierung mit Optimierung durch Anordnen in Elementen („Tiles“) mit identischen Zeichen. Die Elementgröße beträgt 128 Bytes.
	(OPT2, <i>nnn</i> )	Komprimierung mit Optimierung durch Anordnen in Elementen („Tiles“) mit identischen Zeichen. Die Elementgröße beträgt ein Vielfaches von 128 Bytes.  <i>nnn</i> bestimmt die Elementgröße durch Multiplizieren mit 128. Mögliche Werte: 1 - 255.  Beispiel: (OPT2, 5) ergibt eine Elementgröße von 640 Bytes.
<b>Standard-Einstellung</b>	OPT2	Synonym für (OPT2, 1).
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	



# 44 COMP - Parameter für Natural Complete/SMARTS

## Interface

---

▪ COMP Parameter Syntax .....	154
▪ NTCOMP-Makro-Syntax .....	154
▪ Schlüsselwort-Subparameter .....	155
▪ Beispiel für NTCOMP-Makro .....	159
▪ Beispiel für COMP-Parameter .....	159

Dieser Natural-Profilparameter dient zum Angeben der Parameter für das Natural Com-plete/SMARTS Interface. Er entspricht dem Makro `NTCOMP` im Natural-Parametermodul.

<b>Mögliche Werte</b>	Siehe <i>COMP Parameter Syntax</i> .	
<b>Standard-Einstellung</b>	Siehe <i>Schlüsselwort-Subparameter</i> .	
<b>Dynamische Angabe</b>	ja	Der Parameter COMP kann nur dynamisch angegeben werden. Im Natural-Parametermodul wird das Makro <code>NTCOMP</code> verwendet.
<b>Angabe innerhalb der Session</b>	nein	



**Anmerkung:** Informationen zum Natural Com-plete/SMARTS Interface siehe *Natural under Com-plete/SMARTS* in der *TP Monitor Interfaces*-Dokumentation.

## COMP Parameter Syntax

Der Profilparameter COMP wird wie folgt angegeben:

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

## NTCOMP-Makro-Syntax

Das Makro NTCOMP wird wie folgt angegeben:

```
NTCOMP EXIT=value, *
        HCDTID=value, *
        INITID=value, *
        LC=value, *
        LE370=value, *
        MSGHDR=value, *
        NTHSIZE=value, *
        SERVER=value, *
        SPIEA=value, *
        THABOVE=value, *
        TTYxxx=value, *
        UCTRAN=value, *
        U2PRINT=value
```

## Schlüsselwort-Subparameter

EXIT | HCDTID | INITID | LC | LE370 | MSGHDR | NTHSIZE | SERVER | SPIEA | THABOVE | TTY<sub>xxx</sub> | UCTRAN | U2PRINT

### EXIT - User Exit Module Name

EXIT=*value* definiert den Namen eines User Exit-Moduls, das vor der Initialisierung von Natural während der Initialisierung einer Session aufgerufen werden kann.

Wert:	Erklärung:
1 - 8 Zeichen oder ' '	Name des User Exit.
' ' (leer)	Es wird kein User Exit verwendet. Dies ist der Standardwert.

### HCDTID – Initialisierung des Hardcopy-Bestimmungsorts

HCDTID=*value* steuert die Initialisierung des Hardcopy-Bestimmungsorts (Destination).

Wert:	Erklärung:
ON	Der Hardcopy-Bestimmungsort wird mit der Terminal-Kennung initialisiert.
OFF	Der Hardcopy-Bestimmungsort entspricht dem logischen Namen des Terminals. Dies ist der Standardwert.

### INITID – Inhalt von \*INIT-ID

INITID=*value* steuert den Inhalt der Systemvariablen \*INIT-ID.

Wert:	Erklärung:
TIBNAM	*INIT-ID enthält den logischen Einheitsnamen des Benutzerterminals.
TID	*INIT-ID enthält die Zeichenkette <i>lbnnnnnn</i> , wobei <i>l</i> die Stack-Ebene ist, auf der die Session läuft, <i>b</i> leer ist und <i>nnnnnn</i> die rechtbündig ausgerichtete TID-Nummer ohne führende Nullen ist. Dies ist der Standardwert (Natural-Terminal-Kennung).
CPATCH	*INIT-ID enthält dieselbe Zeichenkette wie bei INITID=TID, jedoch ist <i>b</i> kein Leerzeichen, sondern das Complete-Einfügungszeichen (Patch Character).

## LC – Kleinschreibungsmodus einschalten

Mit `LC=value` kann das Terminal zwischen Kleinschreibungs- und Großschreibungsmodus umgeschaltet werden.

Wert:	Erklärung:
ON	Kleinschreibungsmodus. Dies ist der Standardwert.
OFF	Großschreibungsmodus.

## LE370 – Nutzung der LE/370-Umgebung

`LE370=value` gibt die Nutzung der LE/370-Umgebung als vorinitialisierte Umgebung (CEEPIPI Interface) unter Complete/SMARTS an.

Wert:	Erklärung:
ON	Alle 3GL-Aufrufe werden in der vorinitialisierten LE/370-Enklave abgewickelt.
OFF	Dies ist der Standardwert.

## MSGHDR – Aktivierung der Meldungsüberschrift

`MSGHDR=value` aktiviert bzw. deaktiviert eine Meldungsüberschrift für Natural-Fehler- oder Beendigungsmeldungen unter Verwendung der Message Switching Facility von Complete für asynchrone Natural-Transaktionen

Wert:	Erklärung:
ON	Die Meldungsüberschrift ist aktiviert. Dies ist der Standardwert.
OFF	Die Meldungsüberschrift ist deaktiviert.

## NTHSIZE – Natural-Thread-Größe

`NTHSIZE=value` gibt die Größe des Speicherbereichs an, der für Natural-Puffer, Datenbereiche und Threads genutzt wird.

Wert:	Erklärung:
256 - 2097151	Größe in KB, wobei das eigentliche obere Limit durch die Größe des Complete-Threads bestimmt ist.
1024	Dies ist der Standardwert.



**Anmerkung:** Dieser Speicherbereich wird innerhalb des physischen Complete-Thread zugewiesen. Der verbleibende Bereich (Complete Region-Größe RG für die Natural-Transaktion minus NTHSIZE-Wert) steht für das dynamische Laden von Nicht-Natural-Subroutinen, die Vergrößerung von Natural Thread Buffers oder Natural Work Pools usw. zur Verfügung.

## SERVER - Name des Natural-Servers

SERVER=*value* gibt den Namen des Natural-Servers an, der während des Complete-Startvorgangs initialisiert wird.

Wert:	Erklärung:
1 - 8 Zeichen	Name des Natural Server.
NCFNAT92	Dies ist der Standardwert (versionsabhängig).



### Anmerkungen:

1. Der angegebene Server dient dazu, über Natural-Sitzungen hinweg gemeinsam genutzte Speicher und Tabellen, z.B. lokale Buffer Pools, zu verwalten. Der Server muss im Complete Startup definiert werden.
2. Sie können das mitgelieferte Server-Modul NCFNAT92 unter einem anderen Namen kopieren und verschiedene Natural Complete Interfaces mit verschiedenen Servern verlinken und laufen lassen, d.h., mit verschiedenen Sätzen lokaler Buffer Pools im selben Complete.

## SPIEA – Aktivierung der ABEXIT Exits

SPIEA=*value* aktiviert bzw. deaktiviert die ABEXIT-Exits.

Wert:	Erklärung:
ON	Aktiviert den ABEXIT Exit. Dies ist der Standardwert.
OFF	Deaktiviert den ABEXIT Exit. Diese Einstellung sollte nur zu Testzwecken benutzt werden.

## THABOVE – Lage des Natural Thread

THABOVE=*value* bestimmt die Lage des Natural Thread (siehe Parameter [NTHSIZE](#)).

Wert:	Erklärung:
ON	Der Natural Thread wird in der Com-plete Thread-Erweiterung oberhalb der 16-MB-Grenze zugeordnet.  Dies ist der Standardwert (Com-plete Thread-Erweiterung benutzen).
OFF	Der Natural Thread wird im physischen Com-plete Thread unterhalb der 16-MB-Grenze zugeordnet.

## TTYxxx - TTY-Geräte-Steuerzeichen

TTYxxx=*value* bestimmt die Fernschreiber-Geräte-Steuerzeichen. Folgende hexadezimale Werte können eingestellt werden:

Wert:	Erklärung:
TTYCR=0D	Fernschreiber-Wagenrücklauf
TTYLF=15	Fernschreiber-Zeilenvorschub
TTYIC=00	Fernschreiber-Leerzeichen
TTYNIC=00	Anzahl der Fernschreiber-Leerzeichen
TTYBS=16	Fernschreiber-Rückschritt
TTYAL=07	Fernschreiber-Alarm



**Anmerkung:** Es gibt keinen Standardwert.

## UCTRAN – Umsetzung von Klein- in Großschreibung bei Com-plete/SMARTS-Fehlermeldungen

UCTRAN=*value* steuert die Umsetzung von Klein- in Großschreibung für die Com-plete/SMARTS-Fehlermeldungen.

Wert:	Erklärung:
ON	Umsetzung in Großbuchstaben aktiviert.
OFF	Umsetzung in Großbuchstaben deaktiviert.  Dies ist der Standardwert.

## U2PRINT – Dynamische Druckerzuordnung

U2PRINT=*value* steuert die dynamische Druckerzuordnungsfunktion von Com-plete bei Hardcopy-Anforderungen.

Wert:	Erklärung:
ON	Natural ruft bei Hardcopy-Anforderungen die Com-plete U2PRINT Routine für die Angabe eines Zieldruckers (Destination) auf.
OFF	Die dynamische Hardcopy-Druckerzuordnung ist deaktiviert. Natural verwendet den Vorgabewert aus dem Profilparameter HCDEST.  Dies ist der Standardwert.

## Beispiel für NTCOMP-Makro

```
NTCOMP LE370=ON,INITID=TIBNAM,NTHSIZE=2000
```

## Beispiel für COMP-Parameter

```
NTCOMP LE370=ON, *
      INITID=TIBNAM, *
      NTHSIZE=2000
```



# 45

## CP - Name der Standard-Codepage

Dieser Natural-Profilparameter bestimmt die Standard-Codepage für Natural-Daten und Natural-Sources.

<b>Mögliche Werte</b>	1 - 64 Zeichen	<p>Der Name der gewünschten Codepage.</p> <p><b>Anmerkung:</b></p> <ol style="list-style-type: none"> <li>1. Es ist jede beliebige Zeichenkette möglich; diese muss jedoch vordefiniert werden, und zwar durch einen der Codepage-Parameter CCSID, CCSN, IANA oder ALIAS des Makros NTCPAGE im Source-Modul NATCONFIG.</li> <li>2. UTF-32 ist nicht erlaubt.</li> <li>3. Informationen zur Multi-Byte-Codepage-Unterstützung siehe <i>Unterstützung von Multi-Byte-Codepages</i> in der <i>Unicode- und Codepage-Unterstützung-Dokumentation</i>.</li> </ol>									
	ON	<p>Stellen Sie die Standard-Codepage für Großrechner wie folgt ein:</p> <p>Die Codepage für z/OS ist abhängig von der Einstellung des Natural-Profilparameters <b>ULANG</b>:</p> <table border="1"> <thead> <tr> <th>ULANG-Parameter-Einstellung:</th> <th>Verwendete Codepage:</th> </tr> </thead> <tbody> <tr> <td>ULANG=1 (Englisch)</td> <td>IBM01140</td> </tr> <tr> <td>ULANG=2 (Deutsch)</td> <td>IBM01141</td> </tr> <tr> <td>ULANG=3 (Französisch)</td> <td>IBM01147</td> </tr> <tr> <td>ULANG=4 (Spanisch)</td> <td>IBM01145</td> </tr> </tbody> </table> <p>Für andere Sprachen wird IBM01140 als Standard-Codepage verwendet.</p> <p><b>Anmerkung:</b></p> <ol style="list-style-type: none"> <li>1. Die sprachcode-abhängige Anpassung des Profilparameters CP gilt nur für den während der Session aktiven Profilparameter <b>ULANG</b>.</li> </ol>	ULANG-Parameter-Einstellung:	Verwendete Codepage:	ULANG=1 (Englisch)	IBM01140	ULANG=2 (Deutsch)	IBM01141	ULANG=3 (Französisch)	IBM01147	ULANG=4 (Spanisch)
ULANG-Parameter-Einstellung:	Verwendete Codepage:										
ULANG=1 (Englisch)	IBM01140										
ULANG=2 (Deutsch)	IBM01141										
ULANG=3 (Französisch)	IBM01147										
ULANG=4 (Spanisch)	IBM01145										

		2. Jegliche spätere Sprachcode-Änderung in Natural Security oder mit dem Terminalkommando %L= hat keine Auswirkung auf die Anfangsdefinition der Standard-Codepage.
	OFF	Keine Codepage-Unterstützung.
	' ' (leer)	Wie bei ON.
	AUTO	Falls vorhanden, wird der Codepage-Name vom Benutzerterminal genommen. Dies gilt nur für die folgenden Online-Umgebungen: TSO, CICS, Com-plete.  <b>Anmerkung:</b>  1. Informationen zur Multi-Byte-Codepage-Unterstützung siehe <i>Unterstützung von Multi-Byte-Codepages</i> in der <i>Unicode- und Codepage-Unterstützung-Dokumentation</i> .  2. CP=AUTO wird in einer Natural Single Point of Development-Umgebung nicht unterstützt.
<b>Standard-Einstellung</b>	OFF	Keine Codepage-Unterstützung.
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	

**Anmerkungen:**

1. Die Standard-Codepage gilt immer dann, wenn für eine codepage-abhängige Operation, z.B. Datenumwandlung nach und aus Unicode (beispielsweise mittels einer statement-spezifischen ENCODED-Option oder durch einen anderen Profilparameter) keine Codepage angegeben wird.
2. Für die aktuelle Natural-Session wird davon ausgegangen, dass alle Codepage-Daten, z.B. Natural-Sources, Inhalte von Feldern im Format A, in diesem Standard-Codepage-Format gespeichert werden. Siehe auch *Natural-Profilparameter und Parameter-Makros* in der *Unicode- und Codepage-Unterstützung-Dokumentation*.
3. Wenn der Profilparameter CP nicht auf OFF, sondern auf einen anderen Wert gesetzt ist, ändert sich der Wert des Profilparameters CFICU auf ON.
4. Wenn der Profilparameter CP nicht auf OFF, sondern auf einen anderen Wert gesetzt ist, werden die mit den Profilparametern TAB, UTAB1, UTAB2 und SCTAB gesetzten Werte beim Start einer Natural-Session ignoriert. Siehe auch *Umsetzungstabellen (Translation Tables)* in der *Unicode- und Codepage-Unterstützung-Dokumentation*.
5. Wenn der Profilparameter CP auf eine Multi-Byte-Codepage (MBCS) gesetzt wird, dann werden die logischen Shift-In- und Shift-Out-Zeichen mit der Codepage geliefert, und deshalb werden die Angaben im Profilparameter SOSI ignoriert.

**Tipps:**

- Um herauszufinden, welche Standard-Codepage sich bei der Auswertung des CP-Parameters ergeben hat, können Sie sich den Inhalt der Systemvariablen \*CODEPAGE anschauen (siehe Systemkommando CPINFO) oder die Funktion *Unicode-Eigenschaften - Funktion: Unicode Properties* der SYSCP-Utility benutzen.
- Um herauszufinden, welche Standard-Codepage zum Kodieren eines Natural-Source-Objekts verwendet wird, können Sie das Systemkommando LIST DIRECTORY oder die SYSCP-Utility benutzen. Die SYSCP-Utility können Sie außerdem benutzen, um die Codepage für ein Source-Objekt zu ändern.



# 46

## CPCVERR - 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 Zielooperand 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.



### Anmerkungen:

1. Dieser Parameter wird nicht bei der Umsetzung von Natural-Sourcecodes berücksichtigt, wenn diese in den Editierbereich geladen oder katalogisiert werden.
2. 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.

<b>Mögliche Werte</b>	ON	Ein Natural-Fehler NAT3413 wird ausgegeben, wenn während der ICU-Umsetzung mindestens ein Codepoint nicht korrekt umgesetzt werden konnte. Bei Ausgabe-Statements erfolgt keine Fehlermeldung.	
	OFF	Es wird kein Fehler ausgegeben, wenn ein oder mehrere Codepoints während der ICU-Umsetzung nicht korrekt umgesetzt werden konnten.	
<b>Standard-Einstellung</b>	ON		
<b>Dynamische Angabe</b>	ja		
<b>Angabe innerhalb der Session</b>	ja	Gültiges Statement:	SET GLOBALS
		Gültiges Kommando:	GLOBALS

Siehe auch:

- *Natural-Profilparameter und Parameter-Makros in der Unicode- und Codepage-Unterstützung-Dokumentation.*
- *Codepage-Unterstützung bei Editoren, Systemkommandos und Utilities in der Unicode- und Codepage-Unterstützung-Dokumentation.*
- *Verwendung eines Fehlertransaktionsprogramms im Leitfaden zur Programmierung.*

# 47

## CPOBJIN - Codepage der Batch-Eingabedatei

Dieser Natural-Profilparameter dient zur Angabe der Codepage für die Batch-Eingabedatei CMOBJIN (siehe *Natural in Batch Mode* in der *Operations*-Dokumentation).

<b>Mögliche Werte</b>	1 - 64 Zeichen	Name der gewünschten Codepage.  <b>Anmerkung:</b> Es kann jede beliebige Zeichenkette angegeben werden. Diese muss jedoch in einem der Codepage-Parameter CCSID, CCSN, IANA oder ALIAS des Makros NTCPAGE im Source-Modul NATCONFIG vordefiniert werden.
	' ' (leer)	Verwendet wird die aus der Auswertung des Profilparameters CP resultierende Codepage.
<b>Standard-Einstellung</b>	' ' (leer)	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	



### Anmerkungen:

1. Bei ausgeschalteter Natural-Codepage-Unterstützung (z.B. bei CP=OFF) wird jeder Wert, der zu diesem Parameter angegeben wird, ignoriert.
2. Siehe auch *Natural-Profilparameter und Parameter-Makros* in der *Unicode- und Codepage-Unterstützung*-Dokumentation.



# 48

## CPPRINT - Codepage der Batch-Ausgabedatei

Dieser Natural-Profilparameter dient zur Angabe der Codepage für die Batch-Ausgabedatei CPPRINT (siehe *Natural in Batch Mode* in der *Operations-Dokumentation*).

<b>Mögliche Werte</b>	1 - 64 Zeichen	Name der gewünschten Codepage.  <b>Anmerkung:</b> Es kann jede beliebige Zeichenkette angegeben werden. Diese muss jedoch in einem der Codepage-Parameter CCSID, CCSN, IANA oder ALIAS des Makros NTCPAGE im Source-Modul NATCONFIG vordefiniert werden.
	' ' (leer)	Verwendet wird die aus der Auswertung des Profilparameters CP resultierende Codepage.
<b>Standard-Einstellung</b>	' ' (leer)	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	



### Anmerkungen:

1. Bei ausgeschalteter Natural-Codepage-Unterstützung (z.B. bei CP=OFF) wird jeder Wert, der zu diesem Parameter angegeben wird, ignoriert.
2. Siehe auch *Natural-Profilparameter und Parameter-Makros* in der *Unicode- und Codepage-Unterstützung-Dokumentation*.



# 49 CPSYNIN - Codepage der Batch-Eingabedatei für

## Kommandos

---

Dieser Natural-Profilparameter dient zur Angabe der Codepage der Batch-Eingabedatei für Kommandos `CMSYNIN` (siehe *Natural in Batch Mode* in der *Operations-Dokumentation*).

<b>Mögliche Werte</b>	1 - 64 Zeichen	Name der gewünschten Codepage.  <b>Anmerkung:</b>  1. Es kann jede beliebige Zeichenkette angegeben werden. Diese muss jedoch in einem der Codepage-Parameter <code>CCSID</code> , <code>CCSN</code> , <code>IANA</code> oder <code>ALIAS</code> des Makros <code>NTCPAGE</code> im Source-Modul <code>NATCONFIG</code> vordefiniert werden.  2. UTF-32 ist nicht erlaubt.
	' ' (leer)	Verwendet wird die aus der Auswertung des Profilparameters <code>CP</code> resultierende Codepage.
<b>Standard-Einstellung</b>	' ' (leer)	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	



### Anmerkungen:

1. Bei ausgeschalteter Natural-Codepage-Unterstützung (z.B. bei `CP=OFF`) wird jeder Wert, der zu diesem Parameter angegeben wird, ignoriert.
2. Siehe auch *Natural-Profilparameter und Parameter-Makros* in der *Unicode- und Codepage-Unterstützung-Dokumentation*.



# 50

## CSIZE - Größe des Con-nect/Con-form-Pufferbereichs

---

Dieser Natural-Profilparameter bestimmt die Größe des Con-nect/Con-form-Puffer-Bereichs.

<b>Mögliche Werte</b>	1 - 512	Puffergröße in KB.
	0	Wenn CSIZE=0 angegeben wird oder wenn der angeforderte Speicherplatz nicht verfügbar ist, kann Con-nect bzw. Con-form nicht benutzt werden.
<b>Standard-Einstellung</b>	0	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	



### Anmerkungen:

1. Dieser Natural-Profilparameter ist nur gültig, wenn Con-nect/Con-form installiert ist.
2. Alternativ können Sie den äquivalenten Natural-Profilparameter [DS](#) oder das Parametermakro [NTDS](#) benutzen.
3. Weitere Informationen siehe Con-nect/Con-form *Installation*-Dokumentation.



# 51 CSTATIC - Statisch verlinkte Module

---

- CSTATIC-Parameter-Syntax ..... 176
- NTCSTAT-Macro-Syntax ..... 177
- Beispiel für CSTATIC-Parameter ..... 177
- Beispiele für NTCSTAT-Makros ..... 177

Dieser Natural-Profilparameter kann benutzt werden, um eine Liste mit Namen von Modulen zu definieren, die mit dem Natural-Parametermodul statisch verlinkt werden sollen. Er entspricht dem Macro `NTCSTAT` im Natural-Parametermodul.

<b>Mögliche Werte</b>	Siehe <a href="#">CSTATIC-Parameter-Syntax</a> .	
<b>Standard-Einstellung</b>	keine	
<b>Dynamische Angabe</b>	nein	
<b>Angabe innerhalb der Session</b>	nein	



#### Anmerkungen:

1. Jedes angegebene und mit dem Natural-Parametermodul verlinkte Modul kann über ein `CALL`-Statement aus einem Natural-Objekt aufgerufen werden.
2. Da die Länge eines Parameterwertes auf 256 Bytes begrenzt ist, ist auch die Anzahl der Modulnamen, die mit `CSTATIC` angegeben werden können, begrenzt. Wenn Sie mehr statisch verlinkte Module definieren wollen, können Sie als Alternative das Makro `NTCSTAT` benutzen.
3. Module, die statisch verlinkt wurden, können ersetzt werden, indem man sie während der Session-Initialisierung dynamisch lädt; siehe Profilparameter `RCA`. Module, die weder statisch noch dynamisch verlinkt wurden, werden geladen, wenn sie das erste Mal mit einem `CALL`-Statement aufgerufen werden.
4. Wenn Sie Module mit einem umgebungsunabhängigen Nukleus verlinken wollen, müssen Sie sie mit dem Parameter `CSTATIC` in zwei Parametermodulen angeben: Eines, das mit dem umgebungsunabhängigen Nukleus verlinkt ist, und eines, das mit dem umgebungsabhängigen Nukleus verlinkt ist. Dabei ist zu beachten, dass für alle anderen Parameterdefinitionen nur das mit dem umgebungsabhängigen Nukleus verlinkte Parametermodul benutzt wird.
5. Weitere Informationen siehe *Modules for Static Linking* in der *Installation*-Dokumentation.

Dieses Dokument behandelt folgende Themen:

## CSTATIC-Parameter-Syntax

Zu jedem Modulnamen (1 - 8 Zeichen) wird eine externe Referenz für den Linkage Editor erzeugt.

```
CSTATIC=module-name
```

Oder, falls die externe Referenz (*entry-name*) anders lautet als der Modulname, kann der Eintragname in Klammern nach dem Modulnamen angegeben werden:

```
CSTATIC=module-name(entry-name)
```

## NTCSTAT-Macro-Syntax

---

Mit dem Makro NTCSTAT kann jeweils ein Modul pro Makro-Aufruf angegeben werden. Zu jedem Modulnamen (1 - 8 Zeichen) wird eine externe Referenz für den Linkage Editor erzeugt.

```
NTCSTAT module-name
```

Oder, falls die externe Referenz (*entry-name*) anders lautet als der Modulname, kann der Eintragname, abgetrennt durch ein Komma, nach dem Modulnamen angegeben werden:

```
NTCSTAT module-name,entry-name
```

## Beispiel für CSTATIC-Parameter

---

```
CSTATIC=(MOD1,MOD7(ENTRY2),MOD12,MOD27($MAIN))
```

## Beispiele für NTCSTAT-Makros

---

```
NTCSTAT MOD1  
NTCSTAT MOD7,ENTRY2  
NTCSTAT MOD12  
NTCSTAT MOD27,$MAIN
```

---

# 52 CV - Kontrollvariable

Mit diesem Session-Parameter wird eine Kontrollvariable referenziert.

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



## Anmerkungen:

1. Eine Kontrollvariable wird mit Format C (siehe *Spezielle Formate im Leitfaden zur Programmierung*) definiert und dient dazu, Feldern dynamisch Attribute zuzuweisen und/oder den Status "modifiziert" eines Feldes in Verbindung mit einem INPUT- oder PROCESS PAGE-Statement zu prüfen; siehe auch *Logische Bedingungen, MODIFIED-Option im Leitfaden zur Programmierung*.
2. 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 ...`
3. Eine einzelne Attribut-Kontrollvariable kann auch zu mehreren Eingabefeldern zugeordnet 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.

4. 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.
5. Die Attribut-Kontrollvariable kann auf maximal drei Dimensionen erweitert werden, zum Beispiel, CONTR(\*), CONTR(\*, \*), CONTR(\*, \*, \*), je nach Rang des entsprechenden Array.

**Beispiel:**

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

# 53

## CVMIN - Status 'Modified' der Kontrollvariable bei Eingabe

---

Dieser Natural-Profilparameter bestimmt, ob der Kontrollvariablen der Status `MODIFIED` (geändert) zugeordnet soll, wenn der Wert des Feldes, an das die Kontrollvariable gebunden ist, durch einen *identischen* Wert überschrieben wird.

<b>Mögliche Werte</b>	ON	Wenn ein Feldwert mit demselben Wert überschrieben wird, dann wird der entsprechenden Kontrollvariablen der Status <code>MODIFIED</code> zugewiesen.
	OFF	Wenn ein Feldwert mit demselben Wert überschrieben wird, dann wird der entsprechenden Kontrollvariablen <i>nicht</i> der Status <code>MODIFIED</code> zugewiesen.
<b>Standard-Einstellung</b>	OFF	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	



**Anmerkung:** Wenn einer Attribut-Kontrollvariablen der Status `MODIFIED` zugewiesen wurde, dann bewertet die `MODIFIED`-Option dies als `TRUE`. Dies gilt unabhängig davon, ob die Eingabe manuell, vom Natural-Stack gelesen oder im Batch-Modus erfolgte.



# 54

## DATSIZE - Minimale Größe des lokalen Daten-Puffers

---

Mit diesem Natural-Profilparameter können Sie die minimale Größe des lokalen Daten-Puffers (DATSIZE) angeben.

<b>Mögliche Werte</b>	10 - 2097151	Minimale Größe des lokalen Daten-Puffers in KB.
<b>Standard-Einstellung</b>	32	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	



### Anmerkungen:

1. Alternativ können Sie den äquivalenten Natural-Profilparameter `DS` oder das Parametermakro `NTDS` benutzen.
2. Der `DATSIZE`-Puffer ist ein Puffer mit variabler Größe. Er wird dynamisch erweitert, wenn während der Session mehr Speicherplatz für lokale Datenbereiche benötigt wird. In einer Thread-Umgebung kann der `DATSIZE`-Puffer temporär außerhalb des Speicher-Thread zugewiesen werden, wenn er zu groß wird. Die Größe des `DATSIZE`-Puffer wird wieder auf die Minimalgröße zurückgesetzt, sobald die Anwendung den Speicherplatz nicht mehr benötigt.

### Funktion des `DATSIZE`-Puffer

Zur Ausführungszeit enthält der `DATSIZE`-Puffer die lokalen Daten, die vom zurzeit ausgeführten Natural-Hauptprogramm verwendet werden, und die lokalen Daten aller, von diesem Programm aufgerufenen, untergeordneten Objekte (Ausnahme: per `FETCH`-Statement aufgerufene Programme).

Wenn Sie Natural in einer Entwicklungsumgebung benutzen, entspricht die erforderliche minimale `DATSIZE` der Standard-Einstellung (d.h. 32 KB). Eine geringere `DATSIZE` ist nur möglich, wenn Sie Natural nur als Laufzeitumgebung benutzen und wenn dabei keine Natural-Utilities zur Verfügung stehen.

### Berechnung der benötigten DATSIZE

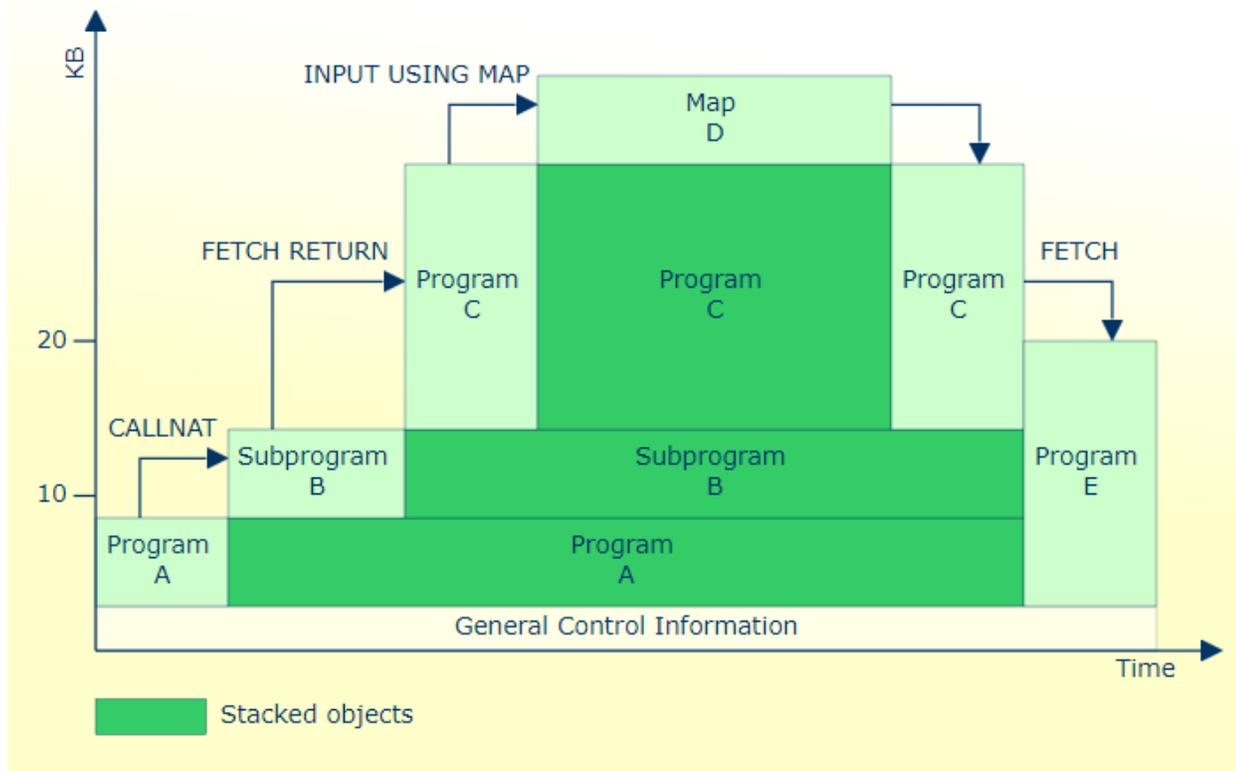
Die tatsächlich benötigte DATSIZE kann wie folgt berechnet werden (siehe Abbildung weiter unten):

Falls vom Natural-Hauptprogramm ein weiteres Objekt aufgerufen wird, werden die lokalen Daten dieses Objekts ebenfalls im DATSIZE-Puffer vorgehalten.

Werden von dem aufgerufenen Objekt weitere Objekte aufgerufen (mit einem der Statements CALLNAT, PERFORM, FETCH RETURN oder INPUT USING MAP oder per Aufruf einer Helpoutine/Help-map), werden ihre lokalen Daten ebenfalls im DATSIZE-Puffer vorgehalten; die lokalen Daten eines aufgerufenen Objekts werden solange im DATSIZE-Puffer vorgehalten, bis die Kontrolle vom aufgerufenen Objekt an das aufrufende Objekt zurückgegeben wird.

Wenn mit einem FETCH-Statement ein weiteres Hauptprogramm aufgerufen wird, dann werden die lokalen Daten aller zuvor aufgerufenen Objekte aus dem DATSIZE-Puffer gelöscht und die lokalen Daten des mit dem FETCH-Statement aufgerufenen Programms werden im DATSIZE-Puffer vorgehalten.

Darüber hinaus wird ein Betrag von ca. 128 Bytes mit allgemeinen Informationen zur Ausführung im DATSIZE-Puffer vorgehalten. Hinzu kommen ca. 128 Bytes mit Steuerungsinformationen zu jedem Objekt, dessen lokale Daten im im DATSIZE-Puffer vorgehalten werden; siehe Abbildung.



Das Systemkommando `LIST` bietet eine Option, mit der Sie sich Verzeichnisinformationen zu einem Objekt anzeigen lassen können. Dort finden Sie auch Angaben zum `DATSIZE`-Speicherbedarf des Objekts (ohne die Steuerungsinformationen).



# 55

## DB - Datenbanktypen und -optionen

---

▪ DB-Parameter-Syntax .....	188
▪ NTDB-Makro-Syntax .....	189
▪ Mögliche Datenbanktypen .....	190
▪ Mögliche Datenbank-Optionen .....	191
▪ Beispiele für DB-Parameter .....	191
▪ Beispiel für NTDB-Makro .....	192

Mit diesem Natural-Profilparameter können Sie Datenbanktypen und -optionen zu allen und zu spezifischen Datenbankkennungen (DBIDs) angeben. Er entspricht dem Makro **NTDB** im Natural-Parametermodul.

<b>Mögliche Werte</b>	Siehe <i>DB-Parameter-Syntax</i> .	
<b>Standard-Einstellung</b>	ADABAS, *	Der Standard-Datenbanktyp ist Adabas.
<b>Dynamische Angabe</b>	ja	Dieser Parameter kann nur dynamisch angegeben werden. Im Natural-Parametermodul muss stattdessen das entsprechende Makro <b>NTDB</b> verwendet werden.
<b>Angabe innerhalb der Session</b>	nein	



#### Anmerkungen:

1. Informationen zu den DBMS-Schnittstellen siehe *Datenbankmanagementsystem-Schnittstellen-Dokumentation*.
2. Unterstützte Versionen siehe *Datenbankmanagementsysteme* in den aktuellen *Freigabemitteilung (Release Notes)*.
3. Bei der Kompilierung wird die Funktionalität der Natural Data Manipulation Language (DML)-Statements soweit eingeschränkt, dass sie der Funktionalität entspricht, die mit dem angegebenen Datenbanktyp zur Verfügung steht.
4. Zur Laufzeit gibt der angegebene Datenbanktyp an, welche Natural DBMS-Schnittstelle zu einer Datenbankkennung aufgerufen wird.

Dieses Dokument behandelt folgende Themen:

## DB-Parameter-Syntax

Der DB-Parameter wird wie folgt angegeben:

### 1. Standard-Datenbank-Definition

Der Standard-Datenbanktyp und seine Standard-Optionen werden wie folgt angegeben. Sie gelten für alle Datenbankkennungen, die nicht ausdrücklich mit dem DB-Parameter bzw. dem Makro **NTDB** angegeben werden. Falls es keine Optionen gibt, können die Kommas und der Stern (\*) weggelassen werden.

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

## 2. Einzelne Datenbank-Definition

Eine einzelne Datenbankkennung wird wie folgt angegeben:

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

## 3. Mehrfache Datenbank-Definition

Mehrere Datenbankkennungen desselben Datenbanktyps mit denselben Optionen können zusammen (in Klammern) angegeben werden:

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

Dabei ist:

Syntax-Element	Wert	Erklärung
<i>database-type</i>	Siehe <a href="#">Mögliche Datenbanktypen</a> .	Datenbanktyp. Der Standardwert ist ADABAS. Dieser Subparameter ist beim Makro NTDB zwingend erforderlich.
<i>database-ID</i>	0 - 65535	Datenbankkennung. Die Datenbankkennung 255 darf nicht angegeben werden, weil sie für interne Zwecke reserviert ist. Sie können eine einzelne Datenbankkennung, eine Liste mit Datenbankkennungen (in Klammern zusammengefasst) oder, wenn Sie die Standardwerte für alle Datenbanken nicht ausdrücklich angeben wollen, einen Stern (*) angeben.
<i>options</i>		Siehe <a href="#">Mögliche Datenbank-Optionen</a> .

## NTDB-Makro-Syntax

Das Makro NTDB wird wie folgt angegeben:

### 1. Standard-Datenbank-Definition

Der Standard-Datenbanktyp und seine Standard-Optionen werden wie folgt angegeben. Sie gelten für alle Datenbankkennungen, die nicht ausdrücklich mit dem DB-Parameter bzw. dem Makro NTDB angegeben werden. Falls es keine Optionen gibt, können die Kommas und der Stern (\*) weggelassen werden.

---

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

## 2. Einzelne Datenbank-Definition

Eine einzelne Datenbankkennung wird wie folgt angegeben:

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

## 3. Mehrfache Datenbank-Definition

Mehrere Datenbankkennungen desselben Datenbanktyps mit denselben Optionen können zusammen (in Klammern) angegeben werden:

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

## Mögliche Datenbanktypen

---

Die Datenbanktypen, die mit dem Profilparameter `DB` bzw. dem Makro `NTDB` angegeben werden können, sind in der folgenden Tabelle aufgelistet.

Bei Adabas gibt ein Datenbanktyp mit Version die Funktionsebene der Datenbankmerkmale an, die von Natural genutzt werden soll.

Bei allen Datenbanktypen außer Adabas muss das zugehörige Datenbankbedienungsmodul (Database Handler) installiert sein. Weitere Informationen siehe Dokumentation des Datenbankbedienungsmodul, das in Ihrer Umgebung benötigt wird.

Datenbanktyp	Adabas-Merkmale oder zu verwendendes Datenbankbedienungsmodul (Database Handler)
ADABAS	Adabas Version 8.2-Merkmale
oder	ADABAS ist der Standardwert.
ADAV82	
ADAV7	Adabas Version 7-Merkmale
ADAV8	Adabas Version 8-Merkmale
VSAM	Natural for VSAM
DLI	Natural for DL/I
Db2	Natural for Db2
PROCESS	Entire System Server
TRS	Adabas Text Retrieval
INCORE	Natural ISPF

## Mögliche Datenbank-Optionen

Die folgenden Optionen können sowohl beim Profilparameter DB als auch beim Makro NTDB angegeben werden:

Option	Erklärung
ACODE	Die Natural-Anwendung muss Adabas mitteilen, ob Codepage- oder Unicode-Unterstützung gewünscht wird, wenn die verwendete Adabas-Datenbankkennung für Zeichenkodierung und Datenkonvertierung freigegeben ist. Deshalb muss mit dem OP-Aufruf die ACODE-Einstellung, die anwendungsspezifische Codepage für alle A-Felder und/oder die WCODE=4095 (UTF-16)-Einstellung für alle W-Felder mitgegeben werden. Siehe auch <i>Unicode and Code Page Support</i> .
WCODE	
ENTIRE	Die Datenbank soll über Entire DB bedient werden.
OPEN	Diese Option gilt nur für Adabas-Datenbanken, bei denen Adabas das Absetzen einer Open-Anforderung erfordert. Wenn für eine solche Datenbank OPEN angegeben wird, dann wird immer eine Open-Anforderung abgesetzt (selbst dann, wenn ETID leer ist).
READ	Die Datenbank soll schreibgeschützt sein.

Die folgenden Optionen können nur beim Profilparameter DB angegeben werden:

Option	Erklärung
NOENTIRE	Setzt die ENTIRE-Option zurück.
NOOPEN	Setzt die OPEN-Option zurück.
NOREAD	Setzt die READ-Option zurück.
OFF	Entfernt eine DB- oder NTDB-Definition für die angegebenen Datenbanken; siehe <a href="#">Beispiele für DB-Parameter</a> weiter unten.

## Beispiele für DB-Parameter

DB=(VSAM,(22,26,33)) Definiert die Datenbanken 22, 26 und 33 als VSAM-Datenbanken.

DB=(\*,READ) Setzt alle Datenbanken, für die die Standard-Datenbankdefinition gilt, auf schreibgeschützt.

DB=(,(8,9),NOREAD) Entfernt die Schreibschutz-Option bei den Datenbanken 8 und 9.

DB=( ,17,OFF)

Setzt die Standard-Datenbankdefinition der Datenbank 17 auf die Standard-Einstellungen zurück.

## Beispiel für NTDB-Makro

---

NTDB DLI ,7 Definiert Datenbank 7 als DL/I-Datenbank.

# 56 DB2 - Parameter für

## SQL-Datenbank-Verwaltungssystem-Schnittstellen

---

▪ DB2-Parameter-Syntax .....	194
▪ NTDB2-Makro-Syntax .....	194
▪ Schlüsselwort-Subparameter .....	195
▪ Beispiel für DB2-Parameter .....	208
▪ Beispiel für NTDB2-Makro .....	208

Mit diesem Natural-Profilparameter können Sie die Parameter für die Datenbank-Verwaltungssystem-Schnittstelle (Natural DBMS Interface) Natural for Db2 angeben. Er entspricht dem Makro [NTDB2](#) im Natural-Parametermodul.

<b>Mögliche Werte</b>	Siehe <a href="#">DB2-Parameter-Syntax</a> .	
<b>Standard-Einstellung</b>	Siehe <a href="#">Schlüsselwort-Subparameter</a> .	
<b>Dynamische Angabe</b>	ja	Der Parameter DB2 kann nur dynamisch angegeben werden. Im Natural-Parametermodul muss stattdessen das entsprechende Makro <a href="#">NTDB2</a> verwendet werden.
<b>Angabe innerhalb der Session</b>	nein	

Dieses Dokument behandelt folgende Themen:

## DB2-Parameter-Syntax

Der Profilparameter DB2 wird wie folgt angegeben:

```
DB2=(keyword-subparameter=value,keyword-subparameter=value,...)
```

Siehe [Schlüsselwort-Subparameter](#).

## NTDB2-Makro-Syntax

Das Makro NTDB2 wird wie folgt angegeben:

NTDB2	BTIGN= <i>value</i> ,	*
	CONVERS= <i>value</i> ,	*
	CONVRS2= <i>value</i> ,	*
	DB2COLL= <i>value</i> ,	*
	DB2GROV= <i>value</i> ,	*
	DB2PLAN= <i>value</i> ,	*
	DB2SSID= <i>value</i> ,	*
	DB2XID= <i>value</i> ,	*
	DDFSERV= <i>value</i> ,	*
	DELIMID= <i>value</i> ,	*
	EBPFSRV= <i>value</i> ,	*
	EBPMAX= <i>value</i> ,	*
	EBPPRAL= <i>value</i> ,	*
	EBPSEC= <i>value</i> ,	*
	ETIGN= <i>value</i> ,	*
	FSERV= <i>value</i> ,	*
	MAXLOOP= <i>value</i> ,	*
	MF= <i>value</i> ,	*

NNPSF=va1ue,	*
PSCIGN=va1ue,	*
REFRESH=va1ue,	*
RETRYPO=va1ue,	*
RWRDONL=va1ue,	*
SMFSRV=va1ue,	*
STATDYN=va1ue	

Siehe [Schlüsselwort-Subparameter](#).

## Schlüsselwort-Subparameter

Es gibt zwei Gruppen von Schlüsselwort-Subparametern:

### Allgemeine Schlüsselwort-Subparameter

BTIGN | CONVERS | CONVR2 | DDFSERV | DELIMID | EBPFSRV | EBPPRAL | EBPSEC | EBPMAX | ETIGN  
| FSERV | MAXLOOP | MF | NNPSF | PSCIGN | REFRESH | RETRYPO | RWRDONL | SMFSRV | STATDYN

### Spezielle Schlüsselwort-Subparameter

Die folgenden Schlüsselwort-Subparameter gehören zusammen. Sie gelten nur bei Db2 für z/OS in Umgebungen, in denen CAF und RRSAF verwendet wird.

DB2COLL | DB2GROV | DB2PLAN | DB2SSID | DB2XID



#### Anmerkungen:

1. Diese Schlüsselwort-Subparameter stellen die Db2-Verbindungs- und Ressourcen-Funktionalität zur Verfügung, die in früheren Versionen durch das Programm NATPLAN verfügbar gemacht wurden. Diese Parameter gelten nur bei Db2 für z/OS und für Umgebungen, in denen entweder die Db2 Call Attachment Facility (CAF) oder die Db2 Resource Recovery Services Attachment Facility (RRSAF) verwendet wird.
2. Eine Ausnahme bildet dabei die Natural for Db2 Stored Procedure-Umgebung, in der Db2 bereits die Db2-Ressourcen basierend auf dem Stored Procedure Creation-Parameter COLLID bereitstellt. Somit werden die hier erwähnten Schlüsselwort-Subparameter in einer Natural for Db2 Stored Procedure-Umgebung nicht benutzt. Bevor in einer CAF oder RRSAF-Umgebung der allererste Db2 SQL-Zugriff durch Natural erfolgt, stellt Natural die Verbindung zu dem gewünschten Db2-Subsystem (DB2SSID) her, und es erfolgt die Zuordnung des gewünschten Plans (DB2PLAN).
3. Falls die Anwendung schon vor der ersten SQL-Abfrage durch NATPLAN mit Db2 verbunden wurde, werden das Macro NTDB2 bzw. der Parameter DB2 ignoriert, und es wird die schon bestehende Verbindung benutzt.

## BTIGN - BACKOUT TRANSACTION-Fehler ignorieren

BTIGN=*value* ermöglicht es, den Fehler zu ignorieren, der dadurch verursacht wurde, dass ein BACKOUT TRANSACTION-Statement für die Rückgängigmachung der aktuellen Transaktion zu spät abgesetzt wurde, weil zuvor schon ein impliziter Syncpoint vom TP-Monitor abgesetzt worden war.

Wert:	Erklärung:
ON	Der Fehler nach einem zu spät abgesetzten BACKOUT TRANSACTION wird ignoriert. Dies ist der Standardwert.
OFF	Der Fehler nach einem zu spät abgesetzten BACKOUT TRANSACTION wird <i>nicht</i> ignoriert.



**Anmerkung:** Dieser Subparameter ist nur in CICS- und IMS TM-Umgebungen relevant.

## CONVERS - Konversationaler Modus unter CICS

CONVERS=*name* ermöglicht den konversationellen Modus in CICS-Umgebungen, in denen kein Natural File Server verwendet wird.

Wert:	Erklärung:
ON	Konversationaler Modus ist erlaubt. Dies ist der Standardwert.
OFF	Konversationaler Modus ist <i>not</i> erlaubt.



**Anmerkungen:**

1. Wenn dieser Subparameter auf OFF gesetzt ist und kein Natural File Server verwendet wird, können Sie keine Datenbankschleifen über Terminal-Ein-/Ausgaben hinweg fortsetzen; in diesem Fall können folgende Schlüssel auftreten.
2. Bei Natural for Db2 gelten die Db2 SQLCODES -501, 504, 507, 514 oder 518.
3. Falls Sie in einer CICS-Umgebung die Funktion *SQL Services (NDB)* (Beschreibung siehe *Database Management System Interfaces-Dokumentation*) ohne Natural for Db2 File Server verwenden, müssen Sie CONVERS=ON angeben, sonst kann der oben erwähnte Fehler auftreten.

## CONVRS2 – Konversationaler Modus 2 unter CICS

CONVRS2=*value* dient dazu, in CICS-Umgebungen den konversationellen Modus 2 zuzulassen bzw. nicht zuzulassen.

Wert:	Erklärung:
ON	Der konversationelle Modus 2 ist gestattet.
OFF	Der konversationelle Modus 2 ist <i>nicht</i> gestattet. Dies ist der Standardwert.



### Anmerkungen:

1. Dieser Subparameter dient dazu, den konversationellen Modus 2 in CICS-Umgebungen zu kontrollieren. Konversationaler Modus 2 bedeutet, dass Aktualisierungstransaktionen über Terminal Ein-/Ausgaben beibehalten werden, bis ein explizites COMMIT oder explizites ROLLBACK-Statement abgesetzt worden ist (Vorsicht: Db2- und CICS-Ressourcen werden über Terminal-Ein-/Ausgaben beibehalten!). Das bedeutet, dass CONVRS2=ON die gleiche Wirkung hat wie der Natural-Profilparameter PSEUDO=OFF, aber mit der Ausnahme, dass der konversationelle Modus nach einem Db2-Datenbank-Aktualisierungs-Statement (UPDATE, DELETE, INSERT) begonnen und nach einem COMMIT oder ROLLBACK wieder beendet wird, während PSEUDO=OFF konversationellen Modus für die ganze Natural-Session bewirkt.
2. Siehe auch CALLNAT-Subprogramm NDBCONV (in der *Database Management System Interfaces*-Dokumentation), mit dem der konversationelle Modus 2 dynamisch gesetzt und zurückgesetzt werden kann.

## DB2COLL – Name der Db2 Collection

DB2COLL=*value* dient zur Angabe des Collection-Namens von Db2-Paketen, die von der Anwendung in einer Umgebung benutzt werden, in das RRSF Interface eingesetzt wird.

Wert:	Erklärung:
<i>value</i>	Ein gültiger, 18 Zeichen langer Db2 Collection-Name.
' ' (leer)	Es ist kein Name angegeben. Dies ist der Standardwert.



### Anmerkungen:

1. Dieser Subparameter wird nur durch Natural für z/OS ausgewertet.
2. Dieser Parameter wird vom RRSF Interface nur dann ausgewertet, wenn das DB2PLAN-Zeichen ein Fragezeichen (?) als erstes Zeichen enthält.
3. Siehe auch [Spezielle Schlüsselwort-Subparameter](#).

## DB2GROV – Priorität gegenüber Db2-Gruppe

DB2GROV=*value* gibt an, ob die durch DB2SSID gekennzeichnete Verbindung zum Db2-System mit dem einzelnen Db2-Subsystem oder mit der Db2 Sharing Group hergestellt werden soll, falls eine Sharing Group und ein einzelnes Db2 mit identischer DB2SSID existiert.

Wert:	Erklärung:
' ' (leer)	Die Verbindung wird zu der durch DB2SSID gekennzeichneten Db2 Sharing Group hergestellt. Dies ist der Standardwert.
NOGROUP	Die Verbindung wird zu dem durch DB2SSID gekennzeichneten Subsystem hergestellt.



### Anmerkungen:

1. Dieser Subparameter wird nur durch Natural für z/OS ausgewertet.
2. Siehe auch [Spezielle Schlüsselwort-Subparameter](#).

## DB2PLAN – Name des Db2-Plans

DB2PLAN=*value* dient zur Angabe des Namens des von der Anwendung benutzten Plans.

Wert:	Erklärung:
<i>value</i>	Ein gültiger, 8 Zeichen langer Db2-Planname. Wenn das erste Zeichen ein Fragezeichen (?) ist und wenn die Anwendung das RRSAP Interface benutzt, dann verwendet die Anwendung die Pakete, die durch den Collection-Namen gekennzeichnet sind, der mit dem Subparameter DB2COLL angegeben wurde.
' ' (leer)	Es ist kein Name angegeben. Dies ist der Standardwert.



### Anmerkungen:

1. Dieser Subparameter wird nur durch Natural für z/OS ausgewertet.
2. Siehe auch [Spezielle Schlüsselwort-Subparameter](#).

## DB2SSID – Db2-Subsystemkennung

DB2SSID=*value* gibt den Namen der Db2 Sharing Group bzw. den Namen des Db2-Subsystems an, zu dem die Verbindung hergestellt werden soll.

Wert:	Erklärung:
<i>value</i>	Ein gültiger, 4 Zeichen langer Name einer Db2 Sharing Group oder eines Db2-Subsystems.
'' (leer)	Es ist kein Name angegeben. Dies ist der Standardwert.

**Anmerkungen:**

1. Dieser Subparameter wird nur durch Natural für z/OS ausgewertet.
2. Siehe auch [Spezielle Schlüsselwort-Subparameter](#).

**DB2XID – Globale Db2-Transaktionskennung**

DB2XID=*value* gibt an, ob das RRSAF Interface eine globale Transaktionskennung verwenden soll oder nicht.

Wert:	Erklärung:
ON	RRSAF erzeugt eine globale Transaktionskennung. Dies ist der Standardwert.
OFF	RRSAF erzeugt keine globale Transaktionskennung.

**Anmerkungen:**

1. Dieser Subparameter wird nur durch Natural für z/OS ausgewertet.
2. Siehe auch [Spezielle Schlüsselwort-Subparameter](#).

**DDFSERV – Alternativer DD-Name für den Natural File Server**

DDFSERV=*ddname* gibt Folgendes an: Entweder einen DD-Namen für den Natural File Server (VSAM) oder den Namen des gemeinsam genutzten Speicherobjekts, das als File Server (SMFSRV=ON) benutzt wird.

Wert:	Erklärung:
<i>ddname</i>	Jeder gültige 8 Zeichen lange DD-Name.
CMFSERV	Dies ist der Standard-Name.

## DELIMID – Escape-Zeichen für Begrenzungszeichen

DELIMID=*value* gibt das Escape-Zeichen an, das für die Generierung von SQL-Begrenzungszeichen für die Spaltennamen und Tabellennamen in SQL-Statements verwendet werden soll.

Wert:	Erklärung:
DQ	Doppeltes Hochkomma (")
SQ	Einzelnes Hochkomma (')
OFF	Begrenzungszeichnungen sind nicht aktiv. Dies ist der Standardwert.



### Anmerkungen:

1. Ein Begrenzungszeichen besteht aus einer Folge von einem oder mehreren, in Escape-Zeichen eingeschlossenen Zeichen. Sie müssen ein Begrenzungszeichen angeben, wenn Sie reservierte SQL-Wörter für Spaltennamen und Tabellennamen benutzen; siehe [Beispiel für DELIMID](#).
2. Um die Generierung von Begrenzungszeichen zu ermöglichen, muss DELIMID in doppelte ( ' ' ) oder einzelne ( ' ) Hochkommata gesetzt werden.
3. Das für DELIMID angegebene Escape-Zeichen und der SQL STRING DELIMITER schließen sich gegenseitig aus. Das impliziert, dass die zum Einschließen von alphanumerischen Zeichenketten in SQL-Statements verwendete Markierung (doppeltes oder einzelnes Hochkomma) unterschiedlich zu dem Wert sein muss, der bei DELIMID angegeben ist. Wenn Sie Begrenzungszeichen ermöglichen, müssen Sie außerdem sicherstellen, dass der bei DELIMID angegebene Wert mit dem Begrenzungszeichen für SQL-Zeichenketten in Ihrer Db2-Installation übereinstimmt.
4. Um festzustellen, welche Begrenzungszeichen in der SQL-Zeichenkette generiert werden, siehe auch Subparameter [RWRDONL](#).

### Beispiel für DELIMID:

In dem folgenden Beispiel ist als Escape-Zeichen für das Begrenzungszeichen ein doppeltes Hochkomma ( ' ' ) verwendet worden:

Natural-Statement:

```
SELECT FUNCTION INTO #FUNCTION FROM XYZ-T1000
```

Generierte SQL-Zeichenkette:

```
SELECT "FUNCTION" FROM XYZ.T1000
```

### EBPFSSRV - Editor Buffer Pool für den Natural File Server

EBPFSSRV=*value* gibt an, ob der Natural File Server den Software AG Editor buffer pool als Speichermedium benutzt.

Wert:	Erklärung:
ON	The Software AG Buffer Pool soll als Speichermedium für den Natural File Server benutzt werden.  ON <i>muss</i> angegeben werden, wenn der Natural File Server in einer Parallel Sysplex-Umgebung verwendet werden soll. In diesem Fall muss Ihre Natural-Session den Editor-Hilfs-Buffer-Pool benutzen (siehe auch <i>Support of a z/OS Parallel Sysplex Environment in Installing Software AG Editor</i> ).
OFF	Eine VSAM-Datei soll als Speichermedium für den Natural File Server benutzt werden oder ein gemeinsam genutztes Speicherobjekt oberhalb der Grenze wird als File Server (SMFSSRV=ON) benutzt.  Dies ist der Standardwert.

### EBPMAX – Maximale Zuordnung im Editor Buffer Pool

EBPMAX=*value* gibt die maximale Anzahl an Blöcken an, die für jeden Benutzer des Natural File Server zugeordnet werden soll, wenn der Software AG Editor buffer pool als Speichermedium benutzt wird.

Wert:	Erklärung:
0 - 32676	Maximale Anzahl an Blöcken, die zugeordnet werden soll.
100	Dies ist der Standardwert.



#### Anmerkungen:

1. Dieser Subparameter definiert die Obergrenze für die Zuordnung von Buffer Pool-Blöcken zu einem einzelnen Benutzer.
2. Wenn der Subparameter **EBPFSSRV** auf OFF gesetzt ist, wird EBPMAX zur Laufzeit nicht berücksichtigt.

## EBPPRAL – Primäre Zuordnung im Editor Buffer Pool

EBPPRAL=*value* gibt die Anzahl an Blöcken an, die primär für jeden Benutzer des Natural File Server zugeordnet werden soll, wenn der Software AG Editor buffer pool als Speichermedium benutzt soll.

Wert:	Erklärung:
0 - 32676	Anzahl an Blöcken, die primär zugeordnet werden soll.
20	Dies ist der Standardwert.



### Anmerkungen:

1. Wenn der Subparameter **EBPF SRV** auf **OFF** gesetzt ist, wird **EBPPRAL** zur Laufzeit nicht berücksichtigt.

## EBPSEC – Sekundäre Zuordnung im Editor Buffer Pool

EBPSEC=*value* der Subparameter gibt die Anzahl an Blöcken an, die sekundär für jeden Benutzer des Natural File Server zugeordnet werden soll, wenn der Software AG Editor buffer pool als Speichermedium benutzt soll.

Wert:	Erklärung:
0 - 32676	Anzahl an Blöcken, die primär zugeordnet werden soll.
10	Dies ist der Standardwert.



### Anmerkungen:

1. Die sekundäre Zuordnung wird verwendet, wenn die primäre Zuordnung aufgebraucht ist.
2. Wenn der Subparameter **EBPF SRV** auf **OFF** gesetzt ist, wird **EBPSEC** zur Laufzeit nicht berücksichtigt.

## ETIGN - END TRANSACTION-Fehler ignorieren

ETIGN=*value* wird benutzt, um **END TRANSACTION** Statements in einer nachrichtenorientierten IMS-Region (MPP oder nachrichtenorientiertes BPM) zu behandeln.

Wert:	Erklärung:
ON	Der END TRANSACTION-Fehler wird ignoriert, und die Verarbeitung wird fortgesetzt.  Dies ist der Standardwert.
OFF	Der END TRANSACTION-Fehler wird <i>nicht</i> ignoriert.

**Anmerkungen:**

1. Dieser Subparameter ist nur in IMS MPP- und nachrichtenorientierten BPM-Umgebungen relevant.
2. In einer solchen Region kann ein END TRANSACTION Statement nicht durch das Natural IMS TM Interface ausgeführt werden und wird deshalb ohne Benachrichtigung ignoriert. In solchen Situationen kann der Subparameter ETIGN benutzt werden, um ersatzweise eine Fehlermeldung auszugeben.

**FSERV - Natural File Server aktivieren**

FSERV=*value* gibt an, ob der Natural File Server benutzt werden soll und ob er im Falle eines Initialisierungsfehlers gesperrt werden kann.

Wert:	Erklärung:
ON	Der Natural File Server soll benutzt werden.
OFF	Der Natural File Server soll nicht benutzt werden.  Dies ist der Standardwert.
DIS	Der Natural File Server soll benutzt werden, aber er soll gesperrt werden, wenn er nicht initialisiert werden kann.

**Anmerkungen:**

1. Wenn FSERV auf ON gesetzt ist und wenn der Natural File Server nicht betriebsbereit ist, wird die Initialisierung von Natural for Db2 mit einer entsprechenden Natural-Fehlermeldung beendet. Das Natural for Db2 Interface wird deaktiviert, und jeder SQL-Aufruf wird mit einer entsprechenden Fehlermeldung zurückgewiesen.

## MAXLOOP – Maximale Anzahl verschachtelter Programmschleifen

MAXLOOP=*nn* gibt die maximal mögliche Anzahl an verschachtelten SQL-Datenbankzugriff-Statements an.

Wert:	Erklärung:
1 - 99	Maximal mögliche Anzahl an Datenbankzugriff-Statements.
10	Dies ist der Standardwert.

## MF - Multi-Fetch-Zeilenzahl

MF=*value* gibt die Anzahl der Zeilen an, die von Db2 in einer FETCH-Operation abgefragt werden sollen. Der Subparameter MF kann benutzt werden, um Multi-Fetch-Operationen durch Db2 auf globaler Basis zu ermöglichen. Es sind keine Änderungen am Anwendungsprogramm erforderlich.

Bei der statischen Generierung bestimmt der Subparameter MF außerdem, ob ein generiertes DECLARE CURSOR-Statement die Klausel WITH ROWSET POSITIONING enthält:

- Wenn MF auf Null (0) gesetzt ist, enthalten DECLARE CURSOR-Statements in dem statisch generierten Programm keine WITHROWSET POSITIONING-Klausel. Dadurch wird, unabhängig von der Einstellung des MF-Subparameters in der ausführenden Natural-Session, das Multi-Fetching während der Ausführung des statischen Programms verhindert.
- Programme, die vor der Einführung des MF-Subparameters statisch generiert werden, sind nicht in der Lage, Multi-Fetch-Operationen auszuführen, stattdessen wird Single-Fetching verwendet. Wenn MF auf einen Wert größer als Null (0) gesetzt ist, enthalten DECLARE CURSOR-Statements in dem statisch generierten Programm die WITHROWSET POSITIONING-Klausel. Bei diesen Programmen wird Multi-Fetching verwendet, wenn der MF-Subparameter in der ausführenden Natural-Session einen Wert größer als Null (0) hat. Falls die ausführende Natural-Session mit MF=0 läuft, verwenden diese Programme kein Multi-Fetching.

Wenn ein Natural for Db2-Programm in einem FIND-, READ- oder SELECT-Statement bereits Multi-Fetch-Syntax verwendet, dann wird dieses Statement wie im Programm angegeben ausgeführt und nicht durch den Subparameter MF beeinflusst. Unabhängig davon, ob eines dieser Statements schon eine Multi-Fetch-Angabe hat, wird das Statement im Multi-Fetch-Puffer einen Zwischenpeicherplatz für die Anzahl der im Subparameter MF angegebenen Zeilen benutzen.

FIND-, READ- und SELECT-Statements, die mit einem Positioned UPDATE oder Positioned DELETE in Verbindung stehen, verwenden keine Multi-Fetch-Operationen — auch dann nicht, wenn der Subparameter MF auf einen Wert größer als Null (0) gesetzt ist.

Wenn der Subparameter MF auf einen Wert größer als Null (0) gesetzt ist und bei einer oder mehreren Zeilen eine Warnbedingung auftritt (z.B. aufgrund einer Zeilenwertabschneidung), kann Db2 einen SQLCODE +354 zurückgeben. In diesem Fall setzen Sie PSCIGN=ON, um den Rückgabecode als positiven SQLCODE zu ignorieren, oder erhöhen Sie die Größe der empfangenden Felder

im Programm so, dass die Warnbedingung nicht mehr auftritt. In einem Nicht-Multi-Fetch-Modus hat die Warnbedingung keinen SQLCODE größer als Null (0) zur Folge.

Wert:	Erklärung:
1 -32767	Anzahl der Zeilen, die von Db2 in einer FETCH-Operation abgerufen werden sollen.
0	Dies ist der Standardwert.



**Anmerkung:** Dieser Subparameter wird nur von Natural for Db2 verwendet.

### NNPSF – Positives Vorzeichen numerischer Natural-Variablen auf F setzen

NNPSF=*value* wird benutzt, um das Zeichen für das Vorzeichen positiver Natural-Variablen mit Format N zu ändern, wenn diese vom SQL-Datenbanksystem gefüllt werden. Normalerweise haben diese Variablen das C als Zeichen für das positive Vorzeichen. Wenn der Subparameter NNPSF auf ON gesetzt wird, dann wird F als Zeichen für das positive Vorzeichen verwendet.

Wert:	Erklärung:
ON	Positive Zahlen, die vom SQL-Datenbanksystem in Natural-Variable eingestellt werden, erhalten das F als Vorzeichen.
OFF	Positive Zahlen, die vom SQL-Datenbanksystem in Natural-Variable eingestellt werden, bleiben unverändert.  Dies ist der Standardwert.

### PSCIGN - Positive SQLCODEs als SQLCODE 0 behandeln

PSCIGN=*value* hat Einfluss darauf, wie positive SQLCODEs, die vom SQL-Datenbanksystem zurückgeliefert werden, behandelt werden.

Wert:	Erklärung:
ON	Positive SQLCODEs werden als Null behandelt.
OFF	Positive SQLCODEs verursachen eine Fehlermeldung (NAT3700).  Dies ist der Standardwert.



**Anmerkungen:**

1. Wenn der Subparameter PSCIGN auf OFF gesetzt ist, wird die Fehlermeldung NAT3700 ausgegeben.
2. Wenn der Subparameter PSCIGN auf ON gesetzt ist, werden positive SQLCODEs so behandelt, als ob sie Null wären; das bedeutet, es wird keine Fehlermeldung (NAT3700) ausgegeben.

## REFRESH - Automatische Aktualisierung der Db2 Server- und Package Set-Einstellungen

REFRESH=*value* dient dazu, den Db2 Server und Package Set automatisch auf die Werte zu setzen, die bei der Ausführung der letzten Transaktion gültig waren.

Wert:	Erklärung:
ON	Jedes Mal, bevor eine Datenbanktransaktion beginnt, und wenn ein Server oder ein Package Set angegeben worden ist, erfolgt eine automatische Aktualisierung.
OFF	Es erfolgt keine automatische Aktualisierung. Dies ist der Standardwert.



### Anmerkungen:

1. Server und Package Set werden unter Verwendung der Db2 SQL-Statements `CONNECT TO server-name` und `SET CURRENT PACKAGESET = 'package-name'` aktualisiert.

## RETRYPO – Anzahl erneuter Positionierungsversuche

RETRYPO=*value* begrenzt die Anzahl der erneuten Versuche, die von Natural for Db2 unternommen werden, um einen dynamisch verschiebbaren Cursor in einer pseudo-konversationellen Umgebung (IMS MPP oder CICS) erneut zu positionieren.

Wert:	Erklärung:
1 - 2147483648	Anzahl der von Natural for Db2 unternommenen erneuten Versuche.
0	Wenn RETRYPO auf 0 gesetzt ist, erfolgen keine Wiederholungsversuche.
10	Dies ist der Standardwert.



### Anmerkungen:

1. Dieser Subparameter gilt nur für dynamische, blätterfähige Cursors.
2. In pseudo-konversationellen Umgebungen werden die Cursors bei einer Terminal-Ein-/Ausgabe geschlossen. Bei dynamischen blätterfähigen Cursors werden die aktuellen absoluten Positionsnummern und die aktuellen Schlüsselspaltenwerte gespeichert. Nach erfolgter Terminal-Ein-/Ausgabe wird der dynamische, blätterfähige Cursor wieder geöffnet und absolut auf die Position der gespeicherten absoluten Position platziert. Die Inhalte der Schlüsselspalten werden mit den gespeicherten Werten verglichen. Wenn sie übereinstimmen, wird die Verarbeitung mit der nächsten angeforderten Datenbankoperation fortgesetzt.
3. Falls die Inhalte der Schlüsselspalten nicht mit den gespeicherten Werten übereinstimmen, werden die nächsten Spalten abgerufen und mit den gespeicherten Werten verglichen, bis entweder die Werte übereinstimmen oder keine Spalte gefunden wird oder die maximal mögliche Anzahl der Wiederholversuche (RETRYPO) erreicht ist. In letzterem Fall wird die Fehlermeldung NAT3703 ausgegeben. Wenn eine Tabellenzeile abgerufen wird, deren Schlüsselspalten

mit den gespeicherten Werten übereinstimmen, wird die Verarbeitung mit der nächsten Datenbankanweisung fortgesetzt.

4. `RETRYPO` begrenzt die Wiederholversuche in beiden Richtungen (*nächster* bzw. *vorhergehender*).

### RWRDONL - Begrenzungskennzeichen nur für reservierte Wörter generieren

`RWRDONL=value` legt fest, welche Kennzeichen als Begrenzungskennzeichen in einer SQL-Zeichenkette generiert werden.

Wert:	Erklärung:
ON	Nur Kennzeichen, die reservierte Wörter sind, werden als Begrenzungskennzeichen generiert. Die Liste der reservierten Wörter ist eine Zusammenfassung der Einträge aus den Listen für reservierte Wörter für Db2 für z/OS, Db2 für LINUX, OS/2, Windows und UNIX sowie aus ISO/ANSI SQL99.  Dies ist der Standardwert.
OFF	Alle Kennzeichen werden als Begrenzungskennzeichen generiert.



**Anmerkung:** `RWRDONL` wird nur dann wirksam, wenn die Einstellung des Subparameters `DELIMID` Begrenzungskennzeichen zulässt.

### SMFSRV - Shared Memory Objects File Server (FSSM) benutzen

`SMFSRV=value` gibt an, ob ein Shared Memory Objects File Server (FSSM) benutzt wird. Weitere Informationen siehe *File Server – Shared Memory Object* im Abschnitt *Natural for Db2* in der *Database Management System Interfaces-Dokumentation*

Wert:	Erklärung:
ON	Das im Parameter <code>DDFSERV</code> angegebene, gemeinsam genutzte Speicherobjekt oberhalb der Grenze wird für den FSSM benutzt.
OFF	Das gemeinsam genutzte Speicherobjekt wird nicht benutzt.  Als Speichermedium für den File Server wird entweder eine VSAM-Datei oder der Software AG Editor Buffer Pool benutzt.  Dies ist der Standardwert.

## STATDYN – Umschaltung von statisch auf dynamisch zulassen

STATDYN=*value* wird verwendet, um die Ausführung von statisch generierten SQL-Statements zuzulassen, wenn die statische Ausführung einen Fehler zurückliefert.

Wert:	Erklärung:
NEVER	Die dynamische Ausführung wird niemals gestattet.  Dies ist der Standardwert.
ALWAYS	Die dynamische Ausführung ist immer nach einem Fehler gestattet.
SPECIAL	Die dynamische Ausführung ist nur nach speziellen Fehlern gestattet.  Spezielle Fehler sind: <ul style="list-style-type: none"> <li>■ NAT3706: Lademodul nicht gefunden.</li> <li>■ SQL -805: DBRM (Database Request Module) im Plan nicht vorhanden.</li> <li>■ SQL -818: Zeitstempel stimmen nicht überein.</li> </ul>

## Beispiel für DB2-Parameter

---

```
DB2=(FSERV=DIS,DELIMID=DQ,RWRDONL=ON,STATDYN=ALWAYS)
```

## Beispiel für NTDB2-Makro

---

```
NTDB2 FSERV=ON, *
      DELIMID=DQ, *
      RWRDONL=ON
```

# 57

## DB2SIZE - Natural-Pufferbereich für Db2

Dieser Natural-Profilparameter dient zum Einstellen der maximalen Größe des Pufferbereichs, der von einigen Natural-Datenbankschnittstellenprodukten benötigt wird.

<b>Mögliche Werte</b>	0 - 32767	Maximale Größe des Pufferbereichs in KB.  <b>Anmerkung:</b>  1. Wenn der angeforderte Speicher nicht verfügbar ist, kann das Natural for Db2 Interface nicht benutzt werden.  2. Setzen Sie DB2SIZE auf 0, wenn Natural <i>nicht</i> für Db2 verwendet werden soll.  3. Wenn Natural for Db2 verwendet werden soll, muss DB2SIZE mindestens auf 40 KB gesetzt werden.
<b>Standard-Einstellung</b>	0	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	



### Anmerkungen:

1. Siehe auch die folgenden Abschnitte in der *Installation*-Dokumentation:
2. *Natural Parameter Modifications for Natural for Db2*



# 58

## DBCLOSE - Datenbank-Schließung am Ende einer Session

Dieser Natural-Profilparameter bestimmt, ob Natural alle Datenbanken, auf die es während einer Session zugegriffen hat, am Ende dieser Session schließt.

<b>Mögliche Werte</b>	ON	Natural schließt alle Datenbanken.
	OFF	Natural schließt nur die Datenbanken, die mit einem expliziten Open-Kommando geöffnet worden waren.  Ein explizites Open-Kommando wird in den folgenden Fällen abgesetzt: <ul style="list-style-type: none"> <li>■ Der Profilparameter <a href="#">ETID</a> ist nicht auf ' ' (leer) gesetzt.</li> <li>■ Der Profilparameter <a href="#">DBOPEN</a> ist auf =ON gesetzt.</li> <li>■ Die Datenbanköffnung wird durch die OPEN-Option des Makros <a href="#">NTDB</a> oder des Profilparameters <a href="#">DB</a> erzwungen.</li> </ul>
	ETDB	Natural schließt nur die Datenbanken, die mit dem Profilparameter <a href="#">ETDB</a> angegeben wurden.
<b>Standard-Einstellung</b>	OFF	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	

Weitere Parameter, die bei der Transaktionsverarbeitung berücksichtigt werden: [ADAMODE](#) | [DBOPEN](#) | [ENDBT](#) | [ET](#) | [ETDB](#) | [ETEOP](#) | [ETIO](#) | [ETSYNC](#)



# 59

## DBGAT - Debug Attach Server für NaturalONE

---

■ DBGAT-Parameter-Syntax .....	214
■ NTDBGAT-Makro-Syntax .....	215
■ Beispiel für DBGAT-Parameter .....	215
■ Beispiel für NTDBGAT-Makro .....	215

Dieser Natural-Profilparameter ermöglicht im Zusammenhang mit NaturalONE das Debugging einer externen Natural-Anwendung von einem z/OS-Host-Rechner aus. Er entspricht dem Makro `NTDBGAT` im Natural-Parametermodul.

<b>Mögliche Werte</b>	Siehe <i>DBGAT-Parameter-Syntax</i> .	
<b>Standard-Einstellung</b>	keine	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	
<b>Anwendungsprogrammierschnittstelle</b>	nein	

Ausführliche Informationen zum Debugging von externen Natural-Anwendungen siehe *NaturalONE*-Dokumentation.

Siehe auch *Natural für Attached-Debugging vorbereiten* in der *Natural Debugger*-Dokumentation.

Dieses Dokument behandelt folgende Themen:

## DBGAT-Parameter-Syntax

---

Der Profilparameter `DBGAT` wird wie folgt angegeben:

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

Dabei ist:

Schlüsselwort-Subparameter	Wert	Erklärung
ACTIVE	ON	ON bedeutet, dass der Debug Attach-Mechanismus aktiv ist. Die Natural-Laufzeitumgebung ist bereit für das Debugging.
	OFF	
HOST	1 - 64 Zeichen	Name des Debug Attach Server, zu dem die Verbindung hergestellt werden soll.
PORT	0 - 65535	Nummer des Port, an dem der Debug Attach Server empfängt.
CLID	1 - 64 Zeichen	Client-Kennung des NaturalONE-Projekts, das ausgetestet werden soll.

## NTDBGAT-Makro-Syntax

---

Das Makro NTDBGAT wird wie folgt angegeben:

```
NTDBGAT ACTIVE=value, *  
        HOST=value, *  
        PORT=value, *  
        CLID=value
```

## Beispiel für DBGAT-Parameter

---

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

## Beispiel für NTDBGAT-Makro

---

```
NTDBGAT ACTIVE=ON, *  
        HOST=MYHOST, *  
        PORT=50882
```



# 60

## DBGERR - Automatischer Debugger-Start bei

### Laufzeitfehler

---

Dieser Natural-Profilparameter ermöglicht den automatischen Start des Natural Debugger, wenn zur Laufzeit ein Natural-Fehler auftritt.

<b>Mögliche Werte</b>	ON	Der Debugger wird automatisch gestartet und generiert einen Bildschirm, in dem Sie weitere Informationen zu dem vorliegenden Fehler erhalten.  <b>Anmerkung:</b>  1. Die Laufzeitumgebung gibt beim Auftreten eines Natural-Fehlers die Kontrolle an den Debugger ab, und zwar unabhängig davon, ob der Debugger bereits aktiv ist oder nicht. Dadurch wird der manuelle Bedienungsaufwand vermieden, das Natural-Systemkommando <code>TEST ON</code> in einem solchen Fall benutzen zu müssen.  2. Weitere Informationen siehe <i>Debugger starten</i> in der Natural <i>Debugger-Dokumentation</i> .
	OFF	Der Debugger wird nicht automatisch gestartet.
<b>Standard-Einstellung</b>	OFF	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	



# 61 DBID - Standard-Datenbankkennung für

## Natural-Systemdateien

---

Dieser Natural-Profilparameter kennzeichnet die Standard-Datenbank, in der sich die Natural-Systemdateien (FNAT, FUSER, FDIC, FSEC, FSPPOOL, FPROF, FREG) befinden.

<b>Mögliche Werte</b>	0 - 254 256 - 65535	Die Standard-Datenbankkennung für die Natural-Systemdateien. <b>Anmerkung:</b> Die Datenbankkennung 255 ist für die interne Verwendung reserviert.
<b>Standard-Einstellung</b>	0	
<b>Dynamische Angabe</b>	ja	<b>Anmerkung:</b> Wenn Sie den DBID-Parameter dynamisch angeben, wird die Datenbankkennung für alle Natural-Systemdateien auf diese Einstellung gesetzt. Deshalb müssen Sie den DBID-Parameter <i>vor</i> den einzelnen systemdateispezifischen Profilparametern (FNAT, FUSER, FDIC, FSEC, FSPPOOL, FPROF und FREG) angeben, wenn sie außerdem einen dieser Profilparameter angeben wollen.
<b>Angabe innerhalb der Session</b>	nein	



### Anmerkungen:

1. Die mit dem DBID-Parameter angegebene Datenbankkennung gilt für alle Natural-Systemdateien, für die keine individuelle Datenbankkennung angegeben ist.
2. Datenbankkennungen für einzelne Natural-Systemdateien können mit dem Subparameter *database-ID* der Profilparameter FNAT, FUSER, FDIC, FSEC, FSPPOOL, FPROF und FREG angegeben werden.
3. Der Typ des Datenbanksystems wird durch die Angabe im Makro NTDB bestimmt.



# 62

## DBOPEN - Datenbank-Öffnung ohne ETID

---

Dieser Natural-Profilparameter steuert die Abwicklung der Datenbank-Öffnung.



**Anmerkung:** DBOPEN setzt die Einstellung `ETID=' '` (Leerzeichen) außer Kraft.

<b>Mögliche Werte</b>	ON	Ein Datenbank-OPEN-Kommando wird auch dann abgesetzt, wenn der Parameter ETID auf Leerzeichen gesetzt ist.
	OFF	Es wird kein Datenbank-Öffnungskommando abgesetzt, wenn der Parameter ETID auf Leerzeichen gesetzt ist.  <b>Ausnahme:</b> Ein Datenbank-OPEN-Kommando wird immer an die in <a href="#">ETDB</a> angegebene Datenbank gesendet, selbst wenn der Parameter ETID auf Leerzeichen und DBOPEN auf OFF gesetzt ist.
<b>Standard-Einstellung</b>	OFF	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	

Weitere Parameter, die bei der Transaktionsverarbeitung berücksichtigt werden: [ADAMODE](#) | [DBCLOSE](#) | [ENDBT](#) | [ET](#) | [ETDB](#) | [ETEOP](#) | [ETIO](#) | [ETSYNC](#)



# 63

## DBROLL - Datenbankaufrufe vor Session-Unterbrechung

---

Mit diesem Natural-Profilparameter geben Sie die Anzahl der Datenbankaufrufe an, nach der eine Natural-Session vorübergehend unterbrochen wird, d.h. ein möglicher Roll-out des Natural-Thread durchgeführt werden soll.

<b>Mögliche Werte</b>	0 - 32767	Anzahl der Datenbankaufrufe.
<b>Standard-Einstellung</b>	0	Keine Session-Unterbrechnung bei Datenbankaufrufen.
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	



### Anmerkungen:

1. Dieser Natural-Profilparameter gilt nur unter CICS und Com-plete.
2. Wenn die DBROLL-Anzahl (nicht Null) erreicht wird, setzt Natural eine bedingte CMROLL-Anforderung ab (siehe [Hinweis zu CMROLL](#) in der Beschreibung des Profilparameters [MAXROLL](#)); d.h., wenn andere Sessions auf einen Thread warten, dann wird die Session vorübergehend unterbrochen; was zu einem Roll-out des Natural-Thread führen kann.
3. Unter CICS: Wenn keine andere Session wartet, dann wird nur ein EXEC CICS SUSPEND ausgeführt, um die Kontrolle an andere Tasks mit höherer oder gleichrangiger Dispositionsriorität freizugeben.



# 64 DBUPD - Datenbankänderungen

Dieser Natural-Profilparameter gibt an, ob während der Natural-Session Datenbankänderungen erlaubt sein sollen.

<b>Mögliche Werte</b>	ON	Datenbankänderung erlaubt.
	OFF	<p>Datenbankänderung nicht erlaubt.</p> <p>Wenn ein Programm, das ein datenbankveränderndes Statement (UPDATE, STORE, DELETE, INSERT oder MERGE) enthält, kompiliert wird (CHECK, CATALOG oder STOW), wird die Natural-Fehlermeldung NAT0105 (Datenbankänderung nicht erlaubt) ausgegeben.</p> <p>Wenn ein Programm, das ein datenbankveränderndes Statement (UPDATE, STORE oder DELETE) enthält, ausgeführt wird, erfolgt keine Datenbankänderung. Stattdessen wird bei der nächsten Bildschirm-Ein-/Ausgabe eine Warnmeldung NAT1010 ausgegeben.</p> <p>Des Weiteren stellt eine Datenbankschleife, die ein UPDATE oder DELETE Statement enthält, keine Datensätze in den Hold-Status (kein „Lesen mit Hold“).</p>
<b>Standard-Einstellung</b>	ON	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	
<b>Anwendungsprogrammierschnittstelle</b>	USR1005N	<p>Siehe <i>SYSEXT Utility</i> in der Dokumentation <i>Debugger und Dienstprogramme</i>.</p> <p>* Empfohlen.</p>
	USR1042N *	



# 65 DC - Dezimalstellenzeichen

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

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



**Anmerkungen:**

1. Innerhalb einer Natural-Session kann der Profilparameter DC vom Session-Parameter DC überschrieben werden.
2. Unter Natural Security: Die Einstellung dieses Parameters kann von der Session Parameters-Option des Library-Profiles außer Kraft gesetzt werden.

# 66

## DD - Tagesdifferenz einstellen

---

Dieser Natural-Profilparameter dient dazu, das aktuelle Maschinendatum (wie anhand der internen Maschinenzeit ausgelesen) dadurch anzupassen, dass eine Anzahl von Tagen addiert oder subtrahiert wird. Dadurch wird es möglich, eine Anwendung, die an einem bestimmten Tag laufen sollte, aber aus irgendeinem Grund an diesem Tag nicht laufen konnte, erneut mit dem Datum dieses Tages laufen zu lassen.

Der Profilparameter DD wird wie folgt angegeben:

DD=+nn

oder

DD=-nn

Dabei ist *nn* die Anzahl der Tage.

<b>Mögliche Werte</b>	-32767 bis +32767	Das Maschinendatum wird angepasst. Die Angabe des Plus-Zeichens (+) ist optional.
	0	Es erfolgt keine Anpassung.
<b>Standard-Einstellung</b>	0	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	
<b>Anwendungsprogrammierschnittstelle</b>	USR1005N	Siehe <i>SYSEXT Utility</i> in der Dokumentation <i>Debugger und Dienstprogramme</i> .

Siehe auch the Profilparameter [TD](#) und [YD](#).



# 67

## DELETE - Löschung dynamisch geladener Programme

Dieser Natural-Profilparameter bestimmt, ob dynamisch geladene Programme bei Beendigung des Natural-Programms, in dem sie geladen worden sind, zu löschen sind.

Mit dem Profilparameter `CDYNAM` können Sie die Anzahl der Nicht-Natural-Programme, die gleichzeitig geladen werden können, einschränken.

<b>Mögliche Werte</b>	ON	Dynamisch geladene Programme werden bei Beendigung des Natural-Programms, in dem sie geladen worden sind, gelöscht.
	OFF	Dynamisch geladene Programme werden bei Beendigung des Natural-Programms, in dem sie geladen worden sind, nicht gelöscht. Sie werden solange beibehalten, bis das aktuelle Level 1-Programm beendet wird.
<b>Standard-Einstellung</b>	ON	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	
<b>Anwendungsprogrammierschnittstelle</b>	USR1005N	Siehe <i>SYSEXT Utility</i> in der Dokumentation <i>Debugger und Dienstprogramme</i> .

Es gelten folgende plattformspezifische Anforderungen:

<b>Plattform:</b>	<b>Kommentar:</b>
Unter CICS	In einer CICS-Umgebung gilt dieser Parameter nur, wenn das Nicht-Natural-Programm über Standard-Linkage-Konventionen (SET CONTROL 'P=S') aufgerufen wird.



# 68 DF - Datumsformat

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

<b>Mögliche Werte</b>	S	8-Byte-Darstellung mit 2-stelliger Jahreskomponente und Delimitern ( <i>yy-mm-dd</i> ).  Bei <code>DF=S</code> 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 <a href="#">Anmerkung</a> .
	L	10-Byte-Darstellung mit 4-stelliger Jahreskomponente und Delimitern ( <i>yyyy-mm-dd</i> ). Siehe <a href="#">Anmerkung</a> .
<b>Standard-Einstellung</b>	S	
<b>Gültige Statements</b>	FORMAT	Parameter kann dynamisch mit dem <code>FORMAT</code> -Statement angegeben werden.
	INPUT DISPLAY WRITE PRINT	Parameter kann auf Statement-Ebene und auf Element-Ebene angegeben werden.
	MOVE COMPRESS STACK RUN FETCH	Parameter kann auf Element-Ebene angegeben werden.
<b>Gültiges Kommando</b>	Keines	



## Anmerkungen:

1. Der `DF`-Parameter wird zur Kompilierungszeit ausgewertet.

2. Die Reihenfolge der Tages-, Monats- und Jahreskomponenten sowie die Delimiterzeichen werden durch den **DTFORM**-Profilparameter bestimmt.
3. 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.
4. 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.
5. 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.
6. Siehe auch den Abschnitt *Datumsformat für alphanumerische Darstellung – der DF-Parameter im Leitfaden zur Programmierung*.

# 69

## DFOUT - 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.

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



### Anmerkungen:

1. In einer Natural-Session kann der Profilparameter `DFOUT` durch den Session-Parameter `DFOUT` überschrieben werden.
2. 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.

3. Bei beiden DFOUT-Einstellungen wird die Reihenfolge der Tages-, Monats- und Jahreskomponenten in den Datumswerten durch den Profilparameter `DTFORM` bestimmt.
4. Siehe auch *Datumsinformationen verarbeiten* im *Leitfaden zur Programmierung*.

# 70

## DFSTACK - Datumsformat für Natural-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 Natural-Stack abgelegt werden.

<b>Mögliche Werte</b>	S	Datumsvariablen werden mit 2-stelliger Jahreskomponente und mit durch den Profilparameter <b>DTFORM</b> bestimmten Delimiterzeichen auf dem Natural-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 Natural-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 Natural-Stack abgelegt.  Beispiel: <i>yyyymmdd</i>	
<b>Standard-Einstellung</b>	S		
<b>Dynamische Angabe</b>	ja		
<b>Angabe innerhalb der Session</b>	ja	Gültige Statements:	SET GLOBALS
		Gültiges Kommando:	GLOBALS
<b>Programmierschnittstelle (API)</b>	USR1005N	Siehe <i>SYSEXT Utility</i> in der Dokumentation <i>Debugger und Dienstprogramme</i> .	



**Anmerkungen:**

1. In einer Natural-Session kann der Profilparameter `DFSTACK` durch den Session-Parameter `DFSTACK` überschrieben werden.
2. Der `DFSTACK`-Parameter gilt nicht für `STACK-`, `RUN-` und `FETCH-`Statements, für die der Session-Parameter `DF` gesetzt ist.
3. Siehe auch *Datumsinformationen verarbeiten im Leitfaden zur Programmierung*.

# 71

## DFTITLE - 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).

<b>Mögliche Werte</b>	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>	
<b>Standard-Einstellung</b>	S		
<b>Dynamische Angabe</b>	ja		
<b>Angabe innerhalb der Session</b>	ja	Gültige Statements:	SET GLOBALS
		Gültiges Kommando:	GLOBALS
<b>Programmierschnittstelle (API)</b>	USR1005N	Siehe <i>SYSEXT Utility</i> in der Dokumentation <i>Debugger und Dienstprogramme</i> .	



**Anmerkungen:**

1. In einer Natural-Session kann der Profilparameter `DFTITLE` durch den Session-Parameter `DFTITLE` überschrieben werden.
2. 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).
3. Die Reihenfolge der Tages-, Monats- und Jahreskomponenten sowie die Delimiterzeichen werden durch den Profilparameter `DTFORM` bestimmt.
4. Siehe auch *Datumsinformationen verarbeiten* und *Datumsformat für Standard-Seitenüberschriften – Der DFTITLE-Parameter im Leitfaden zur Programmierung*.

# 72 DL - Ausgabelänge

---

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

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



# 73

## DLISIZE - Pufferbereichsgröße für DL/I

Dieser Natural-Profilparameter bestimmt die maximale Größe des für Natural for DL/I benötigten Pufferbereichs.

<b>Mögliche Werte</b>	26 - 512	Puffergröße in KB.  <b>Anmerkung:</b>  1. Die tatsächlich benötigte Größe ist abhängig von den Angaben im Makro NDLPARM (siehe <i>Natural for DL/I</i> -Dokumentation).  2. Wenn Sie die Standard-Einstellungen in NDLPARM verwenden, genügt DLISIZE=26.
	0	Wenn Sie während der Natural-Session keine DL/I-Unterstützung benötigen, wird empfohlen, Natural mit DLISIZE=0 aufzurufen, um den durch die Handhabung ungenutzter Pufferspeicher verursachten Aufwand zu vermeiden.
<b>Standard-Einstellung</b>	0	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	



### Anmerkungen:

1. Dieser Natural-Profilparameter gilt nur bei Natural for DL/I.
2. Wenn die mit dem Parameter DLISIZE angegebene Größe nicht genügt, wird bei der Initialisierung von Natural for DL/I eine entsprechende Meldung ausgegeben, aus der Sie die anzugebende Größe entnehmen können.
3. Wenn Natural for DL/I installiert ist, werden die entsprechenden Pufferspeicher bei der Initialisierung der Natural-Session angefordert.

4. Falls der angeforderte Speicher nicht verfügbar ist, kann Natural for DL/I nicht benutzt werden.

# 74

## DO - Anzeige-Reihenfolge von Ausgabedaten

Dieser Natural-Profil- und Session-Parameter gibt an, wie Felder für die Anzeige auf Terminals interpretiert werden, die bidirektionale Daten unterstützen.

<b>Mögliche Werte</b>	L	Gibt an, dass die Daten von der Anwendung in logischer Anzeige-Reihenfolge sind. Die Feldzeichen werden entsprechend ihrer Zeicheneigenschaften dargestellt (von links nach rechts oder von rechts nach links).	
	V	Gibt an, dass die Daten von der Anwendung in visueller Anzeige-Reihenfolge sind. <b>Anmerkung:</b> 1. Alle Daten werden, bevor sie zum Terminal gesendet werden, invertiert. 2. Diese Option ist erforderlich bei alten Anwendungen, die für Terminals geschrieben wurden, welche die invertierte (von rechts nach links verlaufende) Anzeigerichtung unterstützen (veranlasst durch den Profilparameter PM=I oder das Termiankommando %VON).	
<b>Standard-Einstellung</b>	L		
<b>Dynamische Angabe</b>	ja		
<b>Angabe innerhalb der Session</b>	ja	Gültiges Statement:	SET GLOBALS
		Gültiges Kommando:	GLOBALS



### Anmerkungen:

1. Das Ein-/Ausgabegerät muss in der Lage sein, in Abhängigkeit von den Zeicheneigenschaften die korrekte Anzeige-Reihenfolge zu erzeugen. Dies ist zum Beispiel der Fall, wenn eine Anwendung in einem Browser unter dem Natural Web I/O Interface läuft. Bei anderen Terminaltypen hat dieser Parameter keinerlei Auswirkung.
2. Ausführliche Informationen zur Verwendung der Einstellung PM=I siehe *Bidirectional Language Support* in der *Unicode and Code Page Support*-Dokumentation.



# 75

## DS - Größe der Pufferspeicher festlegen

---

■ DS-Parameter-Syntax .....	248
■ NTDS-Makro-Syntax .....	249
■ Tabelle der Puffer-Größen .....	249
■ Beispiel für DS-Parameter .....	252
■ Beispiele für NTDS-Makros .....	252

Mit diesem Natural-Profilparameter können Sie die Standard-Einstellungen für die Anfangsgröße verschiedener Natural-Pufferspeicher (Buffers) angeben. Er entspricht dem Makro `NTDS` im Natural-Parametermodul.

<b>Mögliche Werte</b>	Siehe <i>DS-Parameter-Syntax</i> .	
<b>Standard-Einstellung</b>	Siehe <i>Tabelle der Puffer-Größen</i> .	
<b>Dynamische Angabe</b>	ja	Dieser Parameter kann nur dynamisch angegeben werden. Im Natural-Parametermodul muss stattdessen das entsprechende Makro <code>NTDS</code> verwendet werden.
<b>Angabe innerhalb der Session</b>	nein	



**Anmerkungen:**

1. In früheren Versionen von Natural wurden einzelne Natural-Profilparameter (z.B. `SSIZE`) benutzt, um die Größe der Puffer anzugeben. Der Profilparameter `DS` ist eine Art universeller Parameter, mit dem Sie alle Speichergrößen angeben können. Er entspricht dem Makro `NTDS` im Natural-Parametermodul.
2. Sie können auch weiterhin nur die Einzelparameter benutzen oder die Einzelparameter parallel zum Parameter `DS` benutzen. Bei der Parameterauswertung werden die einzelnen Puffergrößen-Parameter intern in das neue `DS`-Parameterformat umgesetzt, zum Beispiel: `SSIZE=55` wird umgesetzt nach `DS=(SSIZE, 55)`.
3. Es gibt jedoch einige Puffer-Größen (z.B. `ESIZE`, `VSIZE`), die nicht mit dem Profilparameter `DS` angegeben werden können. Das hat verschiedene Gründe, z.B. die Größe ist Teil eines größeren Speichers oder die Größenangabe definiert das Gesamtmaximum einer Anzahl von Speichern.
4. Siehe auch *Natural Storage Management* und *General Rules for Parameter Usage* in der *Operations*-Dokumentation.

Dieses Dokument behandelt folgende Themen:

## DS-Parameter-Syntax

---

Der Profilparameter `DS` wird wie folgt angegeben:

```
DS=(name,size,name,size,...)
```

Dabei ist:

Syntax-Element	Wert	Erklärung
<i>name</i>	1 - 8 Zeichen.	Puffer-Name. Siehe <a href="#">Tabelle der Puffer-Größen</a> .
<i>size</i>	-	Puffer-Größe in KB. Grenzwerte siehe <a href="#">Tabelle der Puffer-Größen</a> .

 **Anmerkung:** Es können mehrere Paare mit Puffer-Namen/Puffer-Größen angegeben werden; siehe [Beispiel für DS-Parameter](#).

## NTDS-Makro-Syntax

Das Makro NTDS wird wie folgt angegeben:

```
NTDS name, size
NTDS name, size
...
```

Dabei ist:

Syntax-Element	Wert	Erklärung
<i>name</i>	1 - 8 Zeichen.	Puffer-Name. Siehe <a href="#">Tabelle der Puffer-Größen</a> .
<i>size</i>	-	Puffer-Größe in KB. Grenzwerte siehe <a href="#">Tabelle der Puffer-Größen</a> .

 **Anmerkung:** Für jedes Paar mit Puffer-Namen/Puffer-Größen muss ein eigenes NTDS-Makro angegeben werden; siehe [Beispiele für NTDS-Makros](#).

## Tabelle der Puffer-Größen

Buffer-Name	Beschreibung	Puffer-Größe (KB)	Standard-Einstellung	Verfügbar als Subparameter von DS und alternativ als einzelner Profilparameter
<a href="#">ASIZE</a>	Entire System Server-Hilfspuffer	0 oder 64 - 512	0	ja <b>Anmerkung:</b> Einzelheiten siehe Beschreibung des Profilparameters <a href="#">ASIZE</a> .
<a href="#">BSIZE</a>	Größe des EntireX Broker-Puffers	0 oder 1 - 64	0	ja
<a href="#">CSIZE</a>	Größe des Con-nect/Con-form-Pufferbereichs	0 oder 1 - 512	0	ja

Buffer-Name	Beschreibung	Puffer-Größe (KB)	Standard-Einstellung	Verfügbar als Subparameter von DS und alternativ als einzelner Profilparameter
DATSIZE	Größe des Puffers für lokale Daten	10 - 2097151	32	ja
DSIZE	Anfangsgröße des Debug-Puffers	0 oder 2 - 2097151	2	ja <b>Anmerkung:</b> Mit dem einzelnen Profilparameter <i>DSIZE</i> können Sie außerdem noch eine maximale Größe einstellen.
EDPSIZE	Größe des Software AG Editor-Hilfs-Buffer-Pool	0 oder 48 - 2097151	0	ja
EXCSIZE	Größe des Puffers für das Natural Expert C Interface	entfällt	entfällt	ja <b>Anmerkung:</b> Dieser Subparameter ist obsolet und wird nur noch aus Kompatibilitätsgründen beibehalten.
EXRSIZE	Puffergröße für Natural Expert-Regeltabellen	entfällt	entfällt	ja <b>Anmerkung:</b> Dieser Subparameter ist obsolet und wird nur noch aus Kompatibilitätsgründen beibehalten.
FLTUSER	Größe des Puffers für die Fast Locate-Tabelle und den Subroutinen-Zwischenspeicher.  Die in der Fast Locate-Tabelle enthaltenen Informationen verbessern die Performance, wenn Natural-Objekte wiederholt aufgerufen werden. Die Speicherung von Informationen im Subroutine-Zwischenspeicher bewirkt eine Verbesserung der Performance, wenn der Name des Moduls, in dem die Subroutine definiert ist, anhand des Subroutinen-Namens abgerufen wird, welcher mit dem PERFORM-Statement angegeben wird.	0 oder 4 - 2097151	8	Nur als Subparameter von DS verfügbar.  <b>Anmerkung:</b> Der Wert 0 erzwingt, dass weder eine Fast Locate-Tabelle noch ein Subroutinen-Zwischenspeicher zugeordnet wird, und bewirkt, dass mehr Datenbankaufrufe abgesetzt werden, um Informationen über die aufzurufenden Natural-Objekte abzurufen. Deshalb sollte der Wert 0 nur in Umgebungen verwendet werden, in denen Natural-Objekte desselben Namens in verschiedenen Steplib Libraries (siehe <i>Steplib-Libraries</i> in <i>Natural</i>

Buffer-Name	Beschreibung	Puffer-Größe (KB)	Standard-Einstellung	Verfügbar als Subparameter von DS und alternativ als einzelner Profilparameter
	Der FLTUSER-Puffer enthält nur Informationen, die sich auf Natural-Objekte beziehen, welche zur Anwendung des Benutzers gehören.			<p><i>benutzen</i>) existieren, und die Steplib Libraries sollten immer in der definierten Reihenfolge nach einem Objekt durchsucht werden..</p> <p>Um die Anfangsgröße des Puffers, die zum Speichern aller Einträge in der Fast Locate-Tabelle benötigt wird, abzuschätzen, kann die folgende Formel benutzt werden:</p> $6 \text{ KB} + (\text{Anzahl der Objekte} * 112 \text{ Bytes})$ <p>Was einer Einstellung von DS=(FLTUSER, 21) entspricht.</p> <p>Diese Einstellung gibt die Puffergröße beim Start der Session an. Wird mehr Platz benötigt, dann wird dieser automatisch während der Session zugeordnet.</p>
MONSIZE	Größe des SYSTP-Monitor-Puffers	0 oder 5 - 256	0	ja
MULFETCH	Größe des Multi-fetch-Puffers	0 oder 1 - 1024	64	<p>Nur als Subparameter von DS verfügbar.</p> <p><b>Anmerkung:</b> Ein für diesen Puffer angegebener Wert stellt nicht die Standard-Anfangsgröße, sondern die maximal Größe dar, die für Multi-fetch-Zwecke zugewiesen werden kann.</p>
NAFSIZE	Größe des Puffers für Natural Advanced Facilities	0 oder 1 - 64	0	ja
NSFSIZE	Größe des SAF Interface-Puffers	0 oder 8 - 64	0	<p>Nur als Subparameter von DS verfügbar.</p> <p>Informationen, wie das Natural-Parametermodul</p>

Buffer-Name	Beschreibung	Puffer-Größe (KB)	Standard-Einstellung	Verfügbar als Subparameter von DS und alternativ als einzelner Profilparameter
				angepasst werden muss, siehe <i>Activating Natural SAF Security</i> in der <i>Natural SAF Security</i> -Dokumentation.
RDCSIZE	Größe des Puffers für den Natural Data Collector	0 oder 2 - 128	0	ja
RJESIZE	Anfangsgröße des NATRJE-Puffers	0 oder 1 - 2097151	8	ja
RUNSIZE	Größe des Laufzeit-Puffers	10 - 64	16	ja
SSIZE	Größe des Editor-Arbeitsbereichs	0 oder 40 - 512	64	ja
TSIZE	Größe des Puffers für Adabas Text Retrieval	0 oder 1 - 2097151	0	ja
WSISIZE	Puffer für das Natural Workstation Interface	0 oder 10 - 256	0	ja
ZSIZE	Größe des Entire DB-Pufferbereichs	0 oder 1 - 64	0	ja

Weitere Informationen siehe Beschreibungen der einzelne Profilparameter zum Einstellen der Puffergröße.

## Beispiel für DS-Parameter

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

## Beispiele für NTDS-Makros

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

# 76

## DSC - Datenstromkomprimierung bei Terminals des Typs

### 3270

---

Mit diesem Parameter können Sie die von Natural durchgeführte automatische Optimierung *und* Komprimierung des Bildschirmdatenstroms für Terminals des Typs 3270 ausschalten.

<b>Mögliche Werte</b>	ON	Die Optimierung und Komprimierung des Bildschirmdatenstroms wird verwendet.
	OFF	Die Optimierung und Komprimierung des Bildschirmdatenstroms wird <i>nicht</i> verwendet.
<b>Standard-Einstellung</b>	ON	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	



#### Anmerkungen:

1. Dieser Natural-Profilparameter gilt nur bei Terminals des Typs 3270.
2. Bildschirm-Optimierung bedeutet, dass nur diejenigen Felder des Bildschirms zum Terminal gesendet werden, deren Inhalt sich geändert hat. Die Bildschirm-Komprimierung bedeutet eine (weitergehende) Verringerung der gesendeten Datenmenge; erreicht wird dies durch die Verwendung von Zählern für wiederkehrende Zeichen.
3. Die von Natural durchgeführte Optimierung bewirkt, dass Bildschirmdaten so komprimiert wie möglich gesendet werden. Falls es dadurch zu Konflikten mit der Bildschirmoptimierung eines TP Monitors oder durch Hardware-Einschränkungen kommen sollte, können Sie diesen Parameter benutzen, um die Natural-Bildschirmoptimierung auszuschalten; die Bildschirmdaten werden dann in nicht komprimierter Form gesendet; siehe zum Beispiel *Profile Parameter DSC=OFF Recommended* in der *Natural under CICS*-Dokumentation.
4. Dieser Parameter hat dieselbe Funktion wie das Terminalkommando %R0.

5. Bei Verwendung der Einstellungen `BX=L` oder `BX=R` des Session-Parameters `BX`, sollten Sie die Natural-Bildschirmoptimierung mit `DSC=OFF` oder `%R00FF` ausschalten.

# 77

## DSIZE - Größe des DBLOG-Puffers

---

Dieser Natural-Profilparameter gibt die Größe des Natural-DBLOG-Puffers an.

<b>Mögliche Werte</b>	Siehe <i>DSIZE-Parameter-Syntax</i>	
<b>Standard-Einstellung</b>	2,256	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	



### Anmerkungen:

1. Alternativ können Sie den äquivalenten Natural-Profilparameter *DS* oder das Parametermakro *NTDS* benutzen.
2. Der Natural-DBLOG-Pufferbereich wird von der *DBLOG-Utility* genutzt, die in der *Utilities-Dokumentation* beschrieben ist.

### DSIZE-Parameter-Syntax

Der Parameter *DSIZE* wird wie folgt angegeben:

```
DSIZE=initial-size
```



**Anmerkung:** *DSIZE=0* schaltet die *DBLOG-Utility* aus.

Oder:

DSIZE=(*initial-size,maximum-size*)

Dabei ist:

Syntax-Element	Wert	Erklärung
<i>initial-size</i>	2 - 2097151	Anfangsgröße des DBLOG-Puffers in KB.  <b>Anmerkung:</b> Falls die Anfangsgröße nicht reicht, erhöht Natural (falls nötig, öfters) automatisch die Puffergröße bis zur angegebenen maximalen Größe (siehe unten).
<i>maximum-size</i>	2 - 2097151	Maximale Größe des DBLOG-Puffers in KB.  <b>Anmerkung:</b> Wenn dieser Wert nicht größer als die Anfangsgröße (siehe oben) ist, dann wird der DBLOG-Puffer nicht vergrößert.

**Beispiele:**

DSIZE=100

DSIZE=(,2500)

DSIZE=(50,800)

# 78 DTFORM - Datumsformat

Dieser Natural-Profilparameter gibt das Standard-Format an, in dem Datumsangaben automatisch durch Natural als Teil des Standard-Titels bei Natural-Reports, als Datumskonstante und Datumseingaben geliefert werden sollen.

Mögliche Werte	Wert	Gebiet	Datumsformat
	E	Europa	DD/MM/YYYY
	G	Deutschland	DD.MM.YYYY
	I	International	YYYY-MM-DD
	U	USA	MM/DD/YYYY
<b>Standard-Einstellung</b>	I		
<b>Dynamische Angabe</b>	ja		
<b>Angabe innerhalb der Session</b>	nein		
<b>Anwendungsprogrammierschnittstelle</b>	USR1005N	Siehe <i>SYSEXT Utility</i> in der Dokumentation <i>Debugger und Dienstprogramme</i> .	



## Anmerkungen:

1. Falls nicht `DTFORM=U` angegeben wird, wird als erster Tag einer Woche der Montag angenommen; bei Angabe von `DTFORM=U` wird Sonntag als erster Tag einer Woche angenommen.
2. Bei Datumskonstanten besteht die Jahreskomponente (YYYY) aus vier Ziffern. Bei Reports, Datumseingaben, der Natural-Systemfunktion `VAL` werden nur die letzten beiden Ziffern der Jahreskomponente verwendet. Dies gilt auch, wenn das Datum in ein alphanumerisches Feld verschoben wird.
3. Das Ausgabeformat des Datums im Titel einer Standard-Report-Seite wird außerdem durch den Profilparameter `DFTITLE` angegebn.
4. Siehe auch *Datumsinformationen verarbeiten* und *Standard-Editiermaske für Datum — der DTFORM-Parameter im Leitfaden zur Programmierung*.



# 79

## DU - Dump-Erstellung

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.

Mögliche Werte		
	ON	Bei einem Programmabbruch wird ein Memory-Dump generiert (TP-Monitor-Dump-Datei oder SYSUDUMP 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. <b>Anmerkung:</b> <ol style="list-style-type: none"><li>1. Im Batch-Betrieb sind von Natural ergriffene Folgemaßnahmen von der Einstellung des Profilparameters CC abhängig.</li><li>2. Im Online-Betrieb reagiert Natural mit der Fehlermeldung NAT0950, NAT0953, NAT0954, NAT0955 oder NAT0956.</li><li>3. Weitere Informationen zum ABEND erhalten Sie durch das Systemkommando DUMP.</li></ol>
	SNAP	Bei einem Programmabbruch im Verlauf der Natural-Session wird sofort ein Dump erzeugt. <b>Anmerkung:</b> Die Natural-Session wird anschließend fortgesetzt (wie bei DU=OFF), nachdem der DUMP 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. <b>Anmerkung:</b> Falls Natural für LE-Unterstützung eingestellt ist, beendet Natural die Natural-Session sofort ohne DUMP und übergibt die Kontrolle an die LE-Fehlerbehandlung. Deshalb wird dringend empfohlen, die LE-Laufzeitoption

		TERMTHDACT(UA IMM) 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.  <b>Anmerkung:</b> DU=ABEND steht beim Natural Session-Parameter DU nicht zur Verfügung.	
<b>Standard-Einstellung</b>	OFF		
<b>Dynamische Spezifikation</b>	ja		
<b>Spezifikation in Session</b>	ja	Gültige Statements:	SET GLOBALS
		Gültiges Kommando:	GLOBALS



**Anmerkungen:**

1. In einer Natural-Session kann der Profilparameter DU durch den Session-Parameter DU überschrieben werden.
2. Das Setzen des Profilparameters DU kann aufgrund der I/O-Verarbeitung der Dump-Datei die System-Verarbeitungszeit erheblich beeinträchtigen.
3. Seien Sie vorsichtig, wenn Sie diesen Parameter benutzen, weil alle für den Benutzer gerade aktiven Programme und Subroutinen im Natural Buffer Pool zurückgehalten werden.
4. Die Einstellung DU=ON, DU=SNAP bzw. DU=FORCE kann sich aufgrund einer möglichen Puffer-Fragmentierung merklich negativ auf die System-Verarbeitungszeit auswirken.
5. Der Profilparameter DUE kann benutzt werden, um für spezifische Fehler einen Speicherauszug zu erhalten.
6. Unter Natural Security: Die Einstellung dieses Parameters kann durch die *Session Parameters*-Option des Library-Profiles überschrieben werden.

# 80

## DUE - Dump-Erstellung, fehlerspezifisch

---

Mit diesem Natural-Profilparameter können Sie Natural-Fehlernummern angeben, bei deren Auftreten ein Speicher-Dump erstellt werden soll.

<b>Mögliche Werte</b>	1 - 9999	Eine einzelne Fehlernummer oder mehrere Fehlernummern, durch ein Komma getrennt und in Klammern, bei deren Auftreten ein Speicher-Dump erstellt werden soll.  <b>Anmerkung:</b> Wenn der Parameter DUE mehrmals angegeben wird, werden alle Fehlernummern in einer Tabelle gespeichert.
	OFF	Die Tabelle wird gelöscht, und alle zuvor angegebenen Fehlernummern werden entfernt.
<b>Standard-Einstellung</b>	OFF	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	ja	Terminalkommando %DUE



### Anmerkungen:

1. Die fehlerspezifische Dump-Erstellung kann hilfreich sein, wenn man eine Analyse einer bestimmten Fehlersituation durch Mitarbeiter der Software AG anfordern möchte.
2. Wenn ein Fehler auftritt, der mit DUE angegeben wurde, wird eine Programmprüfung erzwungen.
3. Wenn der Profil-/Session-Parameter DU auf OFF gesetzt ist, wird DU=ON gesetzt.

**Beispiele:**

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

# 81

## DY - Dynamische Attribute

---

- DY-Parameter-Syntax ..... 265
- Beispiele ..... 266

Mit diesem Session-Parameter werden Anzeigefeldern dynamisch Feldattribute zugewiesen

<b>Mögliche Werte</b>	Siehe <i>DY-Parameter-Syntax</i> .	
<b>Standard-Einstellung</b>	Keine	
<b>Gültige Statements</b>	DISPLAY INPUT PRINT WRITE	Parameter kann auf Statement- und/oder Element-Ebene angegeben werden.
<b>Gültiges Kommando</b>	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.

Folgende Themen werden behandelt:

## DY-Parameter-Syntax

```
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> UNDERLINED
```

wird ausgegeben als:

```
THIS i 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)



# 82

## DYNPARM - Verwendung dynamischer Parameter

---

■ DYNPARM-Parameter-Syntax .....	270
■ NTDYNP-Makro-Syntax .....	271
■ Beispiel .....	271

Mit diesem Natural-Profilparameter können Sie die Verwendung dynamischer Profilparameter außerhalb von **PROFILE**- und **SYS**-Profilparameterketten einschränken. Der Parameter entspricht dem Makro **NTDYNP** im Natural-Parametermodul.

<b>Mögliche Werte</b>	ON oder OFF, mit oder ohne Parameterliste. Siehe <i><b>DYNPARAM-Parameter-Syntax</b></i> .	
<b>Standard-Einstellung</b>	ON	Alle Profilparameter können dynamisch angegeben werden.
<b>Dynamische Angabe</b>	ja	Außerhalb von <b>PROFILE</b> - oder „ <b>SYS</b> “-Parameterketten kann der Parameter <b>DYNPARAM</b> nur einmal benutzt werden, und nur dann, wenn das Makro <b>NTDYNP</b> nicht im Natural-Parametermodul angegeben ist.
<b>Angabe innerhalb der Session</b>	nein	
<b>Anwendungsprogrammierschnittstelle</b>	USR1005N	Siehe <i>SYSEXT Utility</i> in der Dokumentation <i>Debugger und Dienstprogramme</i> .



**Anmerkungen:**

1. Die mit dem Parameter **DYNPARAM** (oder dem Makro **NTDYNP**) definierten Einschränkungen gelten nicht innerhalb von **PROFILE**- oder **SYS**-Profilparameterketten. Wird **DYNPARAM** innerhalb von **PROFILE**- oder **SYS**-Profilparameterketten verwendet, dann ersetzt er alle zuvor mit **DYNPARAM** oder **NTDYNP** definierten Einschränkungen.
2. **DYNPARAM** kann in einer Kette nur einmal verwendet werden und sollte am Ende der Kette stehen.

Folgende Themen werden behandelt:

## DYNPARAM-Parameter-Syntax

Der Parameter **DYNPARAM** wird wie folgt angegeben:

```
DYNPARAM=ON
```

Alle Profilparameter können dynamisch angegeben werden.

```
DYNPARAM=(ON,parameter-name,parameter-name,...)
```

Es können nur solche Parameter dynamisch angegeben werden, deren *parameter-name* angegeben ist. Andere Parameter verursachen die Ausgabe einer Fehlermeldung NAT7008.

Oder:

```
DYNPARAM=OFF
```

Es können keine Profilparameter dynamisch angegeben werden.

```
DYNPARAM=(OFF,parameter-name,parameter-name,...)
```

Alle Profilparameter können dynamisch angegeben werden, aber nicht die, deren *parameter-name* angegeben ist. Diese Parameter verursachen die Ausgabe einer Fehlermeldung NAT7008.

## NTDYNP-Makro-Syntax

Das Makro NTDYNP wird wie folgt angegeben:

```
NTDYNP ON,parameter-name,parameter-name,parameter-name
NTDYNP parameter-name,parameter-name,...
```

Es können nur solche Parameter dynamisch angegeben werden, deren *parameter-name* angegeben ist. Andere Parameter verursachen die Ausgabe einer Fehlermeldung NAT7008.

Oder:

```
NTDYNP OFF,parameter-name,parameter-name,parameter-name
NTDYNP parameter-name,parameter-name,...
```

Alle Profilparameter können dynamisch angegeben werden, aber nicht die, deren *parameter-name* angegeben ist. Diese Parameter verursachen die Ausgabe einer Fehlermeldung NAT7008.

## Beispiel

Dieses Beispiel zeigt, wie die dynamischen Parameter **FNAT** und **FSEC** eingeschränkt werden können. Im Natural-Parametermodul wird dazu die folgende Parametereinschränkung definiert:

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

Zusätzlich könnten fast alle Parameterprofile wie folgt aussehen:

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

Falls es einigen besonderen Benutzern gestattet sein soll, alle Parameter einschließlich FNAT und FSEC zu benutzen, könnten ihre Parameterprofile wie folgt aussehen:

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

Dadurch werden normale Benutzer gezwungen, den Parameter **PROFILE** als ersten dynamischen Parameter anzugeben. Danach sind alle Parameter einschließlich FNAT und FSEC zulässig. Natürlich muss der Zugang zur Utility SYSPARM eingeschränkt werden.

# 83

## ECHO - Ausdrucken von Batch-Eingabedaten

### ein-/ausschalten

---

Mit diesem Profilparameter können Sie das Ausdrucken von Eingabedaten aus dem Dataset CMSYNIN oder CMOBJIN bei INPUT-Statements, die Natural während der Batch-Verarbeitung zur Verfügung gestellt werden, ein- und ausschalten.

<b>Mögliche Werte</b>	ON	Natural druckt die während der Batch-Verarbeitung zur Verfügung gestellten Eingabedaten in die Ausgabedatei CMPRINT.
	OFF	Natural druckt die während der Batch-Verarbeitung zur Verfügung gestellten Eingabedaten nicht.
<b>Standard-Einstellung</b>	ON	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	



#### Anmerkungen:

1. Dieser Natural-Profilparameter gilt nur im Batch-Modus.
2. Man kann auch das Ausdrucken einer *einzelnen Eingabedatenzeile* unterdrücken, indem man dieser Zeile eine Zeile voranstellt, die das Terminalkommando %\* zur Unterdrückung des Ausdrucks der nachfolgenden Eingabedatenzeile enthält.



# 84

## EDBP - Software AG Editor Buffer Pool-Definitionen

---

▪ EDBP-Parameter-Syntax .....	276
▪ NTEDBP-Makro-Syntax .....	277
▪ Schlüsselwort-Subparameter .....	277
▪ Beispiel für EDBP-Parameter .....	283
▪ Beispiel für NTEDBP-Makro .....	283

Dieser Natural-Profilparameter steuert die Initialisierung und den Betrieb des Software AG Editor Buffer Pool und seiner Arbeitsdatei. Er entspricht dem Makro `NTEDBP` im Natural-Parametermodul.

<b>Mögliche Werte</b>	Siehe <a href="#">EDBP-Parameter-Syntax</a> .	
<b>Standard-Einstellung</b>	Siehe <a href="#">Schlüsselwort-Subparameter</a> .	
<b>Dynamische Angabe</b>	ja	Dieser Parameter kann nur dynamisch angegeben werden. Im Natural-Parametermodul muss stattdessen das Makro <code>NTEDBP</code> verwendet werden.
<b>Angabe innerhalb der Session</b>	ja	Benutzen Sie die Editor Buffer Pool Services Utility <code>SYSED</code> T.



#### Anmerkungen:

1. Der Editor Buffer Pool wird für eine Session mit dem Profilparameter `BPI` mit `TYPE=EDIT` oder mit dem Profilparameter `EDPSIZE` (Editor-Hilfs-Buffer-Pool) definiert.
2. Gemeinsam genutzter Editor Buffer Pool: Wenn der Editor Buffer Pool von mehreren Natural-Sessions gemeinsam genutzt wird, werden alle Schlüsselwort-Subparameter (außer `DDNAME`, `DSNAME` und `FMODE`) nur von der Session berücksichtigt, die als erste den Editor Buffer Pool während eines Editor Buffer Pool-Kaltstarts initialisiert. Bei einem Editor Buffer Pool-Warmstart werden die Subparameter für den Editor Buffer Pool (außer `DDNAME`, `DSNAME` und `FMODE`) aus der Buffer Pool-Arbeitsdatei gelesen. Mit dem Subparameter `COLD=ON` kann während der Initialisierung des Editor Buffer Pool ein Buffer Pool-Kaltstart erzwungen werden.
3. Editor-Hilfs-Buffer-Pool: Wird ein Editor-Hilfs-Buffer-Pool verwendet (siehe Profilparameter `EDPSIZE`), dann gelten nur die folgenden Subparameter: `FTOUT`, `LRECL` und `MAXLF`.
4. Weitere Informationen zum Editor Buffer Pool siehe *Editor Buffer Pool* in der *Operations*-Dokumentation.
5. Informationen zur Buffer Pool Performance siehe *SYSED*T Utility - *Editor Buffer Pool Administration* in der *Utilities*-Dokumentation.

## EDBP-Parameter-Syntax

Der Profilparameter `EDBP` wird wie folgt angegeben:

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

Siehe [Schlüsselwort-Subparameter](#).

## NTEDBP-Makro-Syntax

Das Makro NTEDBP wird wie folgt angegeben:

```

NTEDBP CHECK=value, *
      COLD=value, *
      CTOUT=value, *
      DDNAME=value, *
      DSNAME=value, *
      DTOUT=value, *
      FMODE=value, *
      FTOUT=value, *
      IMSG=value, *
      ITOUT=value, *
      LRECL=value, *
      LTOUT=value, *
      MAXLF=value, *
      PWORK=value, *
      RECNUM=value, *
      RWORK=value, *
      UTOUT=value

```

Siehe [Schlüsselwort-Subparameter](#).

## Schlüsselwort-Subparameter

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

### CHECK - Software AG Editor beim Session-Start prüfen und initialisieren

CHECK=*value* gibt an, ob der Software AG Editor schon beim Start der Natural-Session anstatt bei seinem ersten Aufruf initialisiert wird.

Wert	Erklärung
ON	<p>Der Software AG Editor wird beim Start der Natural-Session initialisiert, wenn die folgenden Bedingungen erfüllt sind:</p> <ol style="list-style-type: none"> <li>1. Das Software AG Editor-Modul NAT EDT ist mit Natural verlinkt.</li> <li>2. Der Profilparameter S SIZE ist auf einen Wert &gt; 0 gesetzt.</li> </ol> <p>Während der Initialisierung des Software AG Editor wird der S SIZE-Puffer zugeordnet, und es wird auf den Software AG Editor Buffer Pool und (wenn EDPSIZE=0) auf seine Arbeitsdatei zugegriffen. Diese werden dabei auf Verfügbarkeit geprüft.</p>

Wert	Erklärung
OFF	Initialisierung und Püfung erfolgen beim erstmaligen Aufrufen des Software AG Editor. Dies ist der Standardwert.

### COLD - Buffer Pool-Kaltstart

COLD=*value* gibt an, ob ein Buffer Pool-Kaltstart durchgeführt wird.

Wert	Erklärung
ON	Buffer Pool-Kaltstart wird durchgeführt.
OFF	Es wird kein Buffer Pool-Kaltstart durchgeführt. Dies ist der Standardwert.



**Anmerkung:** Ein Kaltstart bedeutet, dass der Buffer Pool-Arbeitsspeicher gelöscht und während der Buffer Pool-Initialisierung neu initialisiert wird. Sämtliche Editor-Wiederherstellungsinformationen und alle in der Arbeitsdatei gespeicherten Buffer Pool-Parameter gehen dabei verloren.

### CTOUT - Zeitüberschreitung bei geänderten Buffer Pool-Blöcken

CTOUT=*value* gibt den Zeitüberschreitungswert für geänderte Buffer Pool-Blöcke an.

Wert	Erklärung
1 - 32767	Zeitüberschreitungswert (in Sekunden).
120	Dies ist der Standardwert.



**Anmerkung:** Ein geänderter Buffer Pool-Block wird in die Arbeitsdatei geschrieben, nachdem der angegebene Zeitraum überschritten wurde und kein ungeänderter oder freier Block verfügbar ist.

### DDNAME - Logischer Name der Arbeitsdatei der JCL-Definition

DDNAME=*value* gibt den Namen der logischen Arbeitsdatei der JCL-Definition an.

Wert	Erklärung
1 - 8 Bytes.	Name der logischen Arbeitsdatei.
CMEDIT	Dies ist der Standardwert.



**Anmerkungen:**

1. Unter CICS: Für die Editor-Arbeitsdatei muss ein entsprechender Dateisteuertableneintrag definiert werden.
2. Unter Com-plete: Der angegebene logische Name der Arbeitsdatei ist der Name der SD-Datei.

### DSNAME - Arbeitsdatei-Dataset-Name

DSNAME=*value* gibt den Arbeitsdatei-Dataset-Namen für Batch und TSO an.

Wert	Erklärung
1 - 44 Bytes.	Name des Arbeitsdatei-Dataset.
	Es gibt keinen Standardwert.



#### Anmerkungen:

1. Dieser Subparameter gilt nur unter z/OS.
2. Wird kein DD JCL-Statement geliefert und kein ALLOC-Statement (nur unter TSO) für die Editor-Arbeitsdatei abgesetzt, dann wird DSNAME dynamisch zugewiesen.

### DTOUT - Zeitüberschreitungsprüfwert für logische Dateien

DTOUT=*value* gibt den Zeitüberschreitungsprüfwert für logische Dateien an.

Wert	Erklärung
1 - 32767	Zeitüberschreitungsprüfwert (in Sekunden).
300	Dies ist der Standardwert.



**Anmerkung:** Logische Dateien werden jedesmal auf Zeitüberschreitung geprüft, wenn der angegebene Zeitraum überschritten worden ist.

### FMODE - Dateimodus der Arbeitsdatei

FMODE=*value* gibt den Dateimodus für die Arbeitsdatei an.

Wert	Erklärung
1 - 2 Zeichen.	Dateimodus.  Unter Com-plete/SMARTS: Der Wert SM gibt an, dass eine SMARTS-Arbeitsdatei benutzt wird. In diesem Fall bestimmt die SMARTS-Umgebungsvariable \$NAT_WORK_ROOT den Pfad.  Unter Com-plete/SMARTS: Wenn statt SM ein anderer Wert angegeben wird, wird eine Com-plete SD-Datei benutzt.  In einer SMARTS-Umgebung ohne Com-plete muss SM angegeben werden.

Wert	Erklärung
A1	Dies ist der Standardwert.

 **Anmerkung:** Dieser Subparameter gilt nur bei Com-plete/SMARTS.

### FTOUT - Zeitüberschreitungswert für logische Dateien

FTOUT=*value* gibt den Zeitüberschreitungswert für logische Dateien an.

Wert	Erklärung
60 - 16777215	Zeitüberschreitungswert (in Sekunden).
86400	Dies ist der Standardwert.

 **Anmerkung:** Eine logische Datei wird gelöscht, wenn der angegebene Zeitraum überschritten worden ist und kein Zugriff erfolgt ist.

### IMSG - Buffer Pool-Initialisierungs- und Beendigungsmeldung

IMSG=*value* gibt an, ob eine Buffer Pool-Initialisierungs- und Beendigungsmeldung an der Operator-Konsole ausgegeben werden soll.

Wert	Erklärung
ON	Es wird eine Buffer Pool-Initialisierungs- und Beendigungsmeldung an der Operator-Konsole ausgegeben.
OFF	Es wird keine Buffer Pool-Initialisierungs- und Beendigungsmeldung an der Operator-Konsole ausgegeben.  Dies ist der Standardwert.

### ITOUT - Zeitüberschreitungswert für Buffer Pool-Initialisierung

ITOUT=*value* gibt den Zeitüberschreitungswert für Buffer Pool-Initialisierung an (nur bei Buffer Pools mit mehreren Benutzern).

Wert	Erklärung
1 - 32767	Zeitüberschreitungswert für Buffer Pool-Initialisierung (in Sekunden).
300	Dies ist der Standardwert.

 **Anmerkung:** Der Buffer Pool wird durch den ersten Benutzer, der auf ihn zugreift, initialisiert. Andere Benutzer müssen warten, bis der erste Benutzer die Initialisierung beendet. Wird die Initialisierung nicht nach dem angegebenen Zeitraum beendet (z.B. wegen eines Abbruchs durch den Benutzer), erhalten alle Benutzer eine Fehlermeldung.

## LRECL - Arbeitsdatei-Datensatzlänge

LRECL=*value* gibt die Buffer Pool-Blockgröße und die Arbeitsdatei-Datensatzlänge an.

Wert	Erklärung
800 - 16384	Länge (in Bytes).
4096	Dies ist der Standardwert.



### Anmerkungen:

1. Dieser Parameter wird nur unter Complete und bei Editor-Hilfs-Buffer-Pools berücksichtigt.
2. In anderen Umgebungen wird die Arbeitsdatei-Datensatzlänge bestimmt, wenn die Editor-Arbeitsdatei angelegt wird.

## LTOUT - Zeitüberschreitungswert für gesperrte Buffer Pool-Blöcke

LTOUT=*value* gibt den Zeitüberschreitungswert für gesperrte Buffer Pool-Blöcke an.

Mögliche Werte:

Wert	Erklärung
1 - 32767	Zeitüberschreitungswert (in Sekunden).
20	Dies ist der Standardwert.



**Anmerkung:** Ein Buffer Pool-Block, der während eines Lesevorgangs aus der Arbeitsdatei gesperrt wurde, wird freigegeben, nachdem der angegebene Zeitraum überschritten worden ist.

## MAXLF - Maximale Anzahl logischer Dateien

MAXLF=*value* gibt die maximale Anzahl logischer Dateien an.

Wert	Erklärung
100 - 999999	Maximale Anzahl logischer Dateien.
1000	Dies ist der Standardwert.

**PWORK - Prozentsatz der als Arbeitsdatensätze verwendeten Arbeitsdatei-Datensätze**

PWORK=*value* gibt den Prozentsatz der während eines Buffer Pool-Kaltstarts als Arbeitsdatensätze verwendeten Arbeitsdatei-Datensätze an.

Wert	Erklärung
0 - 100	Verwendete Arbeitsdatensätze (in Prozent).
50	Dies ist der Standardwert.



**Anmerkung:** Die verbleibenden Datensätze werden als Wiederherstellungsdatensätze verwendet.

**RECNUM - Anzahl der Arbeitsdatei-Datensätze**

RECNUM=*value* gibt die Anzahl der Arbeitsdatei-Datensätze während eines Buffer Pool-Kaltstarts an.

Wert	Erklärung
100 - 65535	Anzahl der Arbeitsdatei-Datensätze.
200	Dies ist der Standardwert.



**Anmerkungen:**

1. Dieser Subparameter gilt nur unter Com-plete.
2. Diese Anzahl bestimmt die Größe der Arbeitsdatei.
3. Die Anzahl der Arbeitsdatei-Datensätze wird bestimmt, wenn die Editor-Arbeitsdatei angelegt wird.

**RWORK - Prozentsatz der für reguläre logische Dateien verwendete Arbeitsdatensätze**

RWORK=*value* gibt den Prozentsatz der Arbeitsdatensätze, die während eines Buffer Pool-Kaltstarts für reguläre logische Dateien verwendet werden.

Wert	Erklärung
51 - 100	Verwendete Arbeitsdatensätze (in Prozent).
90	Dies ist der Standardwert.



**Anmerkung:** Die verbleibenden Datensätze werden intern verwendet, um Blöcke aus dem Buffer Pool freizugeben.

## UTOUT - Zeitüberschreitungswert für unveränderte Buffer Pool-Blöcke

UTOUT=*value* gibt den Zeitüberschreitungswert für unveränderte Buffer Pool-Blöcke an.

Wert	Erklärung
1 - 32767	Zeitüberschreitungswert (in Sekunden).
20	Dies ist der Standardwert.



**Anmerkung:** Ein unveränderter Buffer Pool-Block wird in die Arbeitsdatei geschrieben, nachdem der angegebene Zeitraum überschritten worden ist und kein freier Block verfügbar ist.

## Beispiel für EDBP-Parameter

```
EDBP=(DDNAME=EDFILE1,IMSG=ON,CHECK=ON)
```

## Beispiel für NTEDBP-Makro

```
NTEDBP DDNAME=EDFILE1,IMSG=ON,CHECK=ON
```



# 85

## EDPSIZE - Größe des Software AG Editor-Hilfs-Buffer-Pool

---

Dieser Natural-Profilparameter bestimmt die Größe des Software AG Editor-Hilfs-Buffer-Pool.

<b>Mögliche Werte</b>	48 - 2097151	Größe des Editor-Hilfs-Buffer-Pool in KB.
	0	Es wird kein Editor-Hilfs-Buffer-Pool verwendet.
<b>Standard-Einstellung</b>	0	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	



### Anmerkungen:

1. Dieser Natural-Profilparameter muss benutzt werden, wenn der Software AG Editor in einer z/OS Parallel Sysplex-Umgebung läuft. Er ermöglicht es, dass der Software AG Editor ohne einen (lokalen oder globalen) Software AG Editor Buffer Pool betrieben werden kann.
2. Alternativ können Sie den äquivalenten Natural-Profilparameter [DS](#) oder das Parametermakro [NTDS](#) benutzen.
3. Für den Editor-Hilfs-Buffer-Pool wird keine Software AG Editor-Arbeitsdatei benötigt.
4. Wenn der Hilfs-Buffer-Pool verwendet wird, steht die Wiederherstellungsfunktion des Software AG Editor nicht zur Verfügung.
5. Wenn der Parameter EDPSIZE nicht auf Null gesetzt ist, wird ein Hilfs-Buffer-Pool zugewiesen und verwendet, auch wenn mit dem Profilparameter [BPI](#) bzw. mit dem Makro [NTBPI](#) ein (lokaler oder globaler) Software AG Editor Buffer Pool definiert ist.
6. Weitere Information zum Software AG Editor siehe *Operating the Software AG Editor* in der *Operations-Dokumentation*.



# 86

## EJ - 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.

<b>Mögliche Werte</b>	ON	Ein Seitenvorschub wird ausgeführt.		
	OFF	Es wird kein Seitenvorschub ausgeführt.  <b>Anmerkung:</b> Bei Testläufen, bei denen Seitenumbrüche keine Rolle spielen, empfiehlt es sich, EJ=OFF zu setzen, um Papier zu sparen.		
<b>Standard-Einstellung</b>	ON			
<b>Dynamische Angabe</b>	ja			
<b>Angabe innerhalb der Session</b>	ja	Gültige Statements:	SET GLOBALS	Parameter wird zur Laufzeit ausgewertet.
		Gültiges Kommando:	GLOBALS	Parameter kann mit dem GLOBALS-Systemkommando dynamisch angegeben werden.
<b>Programmierschnittstelle (API)</b>	USR1005N	Siehe <i>SYSEXT Utility</i> in der Dokumentation <i>Debugger und Dienstprogramme</i> .		



### Anmerkungen:

1. In einer Natural-Session kann der Profilparameter EJ durch den Session-Parameter EJ überschrieben werden.
2. Die EJ-Einstellung kann wiederum durch ein EJECT-Statement überschrieben werden.
3. Dieser Parameter gilt nur für den ersten auszugebenden Report (Report 0). Für weitere Reports ist das EJECT-Statement mit Report-Spezifikation (*rep*) zu verwenden.

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



**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); weitere Einzelheiten siehe *Asynchronous Natural Processing under CICS* in der *TP Monitor Interfaces*-Dokumentation.

# 87

## EM - Editiermaske

---

▪ EM-Parameter-Syntax .....	290
▪ Beispiele .....	291
▪ Leerzeichen in Editiermasken .....	291
▪ Standard-Editiermasken .....	292
▪ Editiermasken für numerische Felder .....	292
▪ Editiermasken für alphanumerische Felder .....	295
▪ Editiermasken für binäre Felder – Format B .....	297
▪ Hexadezimale Editiermasken .....	297
▪ Editiermasken für Datums- und Zeitfelder (Formate D und T) .....	299
▪ Editiermasken für logische Felder (Format L) .....	303

Mit dem Session-Parameter `EM` können Sie eine Editiermaske für ein Eingabe- und/oder Ausgabefeld definieren, welches in einem der Statements verwendet wird, die in der folgenden Tabelle unter *Gültige Statements* aufgeführt sind.

<b>Mögliche Werte</b>	Siehe <a href="#">EM-Parameter-Syntax</a> .	
<b>Standard-Einstellung</b>	Keine	
<b>Gültige Statements</b>	FORMAT	Parameter kann dynamisch mit dem <code>FORMAT</code> -Statement angegeben werden.  <b>Anmerkung:</b> Es darf jedoch nur <code>EM=OFF</code> (und keine konkrete Spezifikation einer Editiermaske) angegeben werden
	DEFINE DATA DISPLAY INPUT PRINT PROCESS PAGE/PROCESS PAGE UPDATE WRITE	Parameter kann auf Statement- und/oder Element-Ebene angegeben werden.
	MOVE EDITED	Parameter kann auf Element-Ebene angegeben werden.
<b>Gültiges Kommando</b>	Keines	



#### Anmerkungen:

1. Informationen zu Unicode-Editiermasken siehe Session-Parameter [EMU](#).
2. Der Session-Parameter `EM` kann auch bei Feldern mit Format `U` (Unicode) verwendet werden, siehe auch *Unicode- und Codepage-Unterstützung in der Natural-Programmiersprache, Session-Parameter, EMU, ICU, LCU, TCU im Vergleich zu EM, IC, LC, TC in der Unicode- und Codepage-Unterstützung-Dokumentation*.
3. Siehe auch *Editiermasken – der EM-Parameter im Leitfaden zur Programmierung*.

Die folgenden Themen werden nachfolgend erörtert:

## EM-Parameter-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 Datebdefinitionsmodul (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` und `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)
```

Sie können eine Abfolge derselben signifikanten Zeichen durch eine numerische Notation ersetzen, zum Beispiel: `x(8)` for `xxxxxxxx`. Die folgenden Beispiele veranschaulichen die verkürzte Notation, die für die signifikanten Zeichen von numerischen (`Z`, `9`), hexadezimalen (`H`), alphanumerischen (`X`) und Datums- (`N`, `L`) Editiermasken verwendet werden können:

```
EM=9(4)-9(5)      is equivalent to: EM=9999-99999
EM=H(10)          is equivalent to: EM=HHHHHHHHHHH
EM=X(6)..X(3)     is equivalent to: EM=XXXXXX..XXX
EM=YYYY-L(8)-DD-N(8) is equivalent to: EM=YYYY-LLLLLLLL-DD-NNNNNNNN
```

## Leerzeichen in Editiermasken

Unmittelbar hinter dem Gleichheitszeichen (=) des EM-Parameters sind keine Leerzeichen zulässig (zum Beispiel: `EM=<blank>XXX`).

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 <a href="#">DTFORM</a> gesetzt)
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 an die 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 Dezimaltrennzeichen (Punkt oder 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 Dezimaltrennzeichen (Komma oder Punkt) dar und ist signifikant. Nachfolgende Punkte werden als Literale behandelt.</p> <p><b>Anmerkung:</b> An dieser Stelle stellt der Punkt das zurzeit als Dezimaltrennzeichen festgelegte Zeichen dar. Falls mit dem Session- oder Profilparameter <b>DC</b> 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 Dezimaltrennzeichen darf kein Z stehen. Ein Nullwert kann unter Einbeziehung aller Zs in der Editiermaske als lauter Leerzeichen ausgegeben werden (siehe auch Session-Parameter <b>ZP</b> ).

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 Editiermaske 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)
<b>Editiermaske</b>					
EM=9.9	0.0	4.	7.	2.	0.
EM=99	00	54	87	62	30
EM=S99	+00	-54	+87	+62	+30
EM=+Z9	+0	-54	+87	+62	+30
EM=-9.99	0.03	-4.	7.	2.	0.
EM=N9	0	-4	7	2	0
EM=*9.99	0.03	4.	7.	2.	0.
EM=Z99	00	54	87	962	830
EM=*EURZZ9.9	EUR**0.0	EUR*54.	EUR*87.	EUR962.	EUR830.
EM=999+	000+	054-	087+	962+	830+
EM=999-	000	054-	087	962	830
IC=\$ EM=ZZZ.99	\$.03	\$54.	\$87.	\$962.	\$830.
<b>EM=H(6)</b>					

- 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 X, 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)          /* 'B-L-U', 3 bytes of field only.
WRITE #TEXT (EM=X-X-X-X-X)     /* 'B-L-U-E-', with truncated mask.
END
```

### 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 zur Eingabe-Editiermaske](#)
- [Hinweise zur 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 <a href="#">Hinweise für Eingabe-Editiermaske</a> ).
YY	Jahr, zweistellig (siehe Abschnitt <a href="#">Hinweise für Eingabe-Editiermaske</a> ).
Y	Jahr, einstellig. Darf nicht für Eingabefelder verwendet werden.
WW	Woche (siehe die Abschnitte <a href="#">Hinweise für Eingabe-Editiermaske</a> und <a href="#">Hinweise für Wochenanzeige in Ausgabe-Editiermaske</a> ).
ZW	Woche mit Nullwertunterdrückung (siehe die Abschnitte <a href="#">Hinweise für Eingabe-Editiermaske</a> und <a href="#">Hinweise für Wochenanzeige in Ausgabe-Editiermaske</a> ).
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 <a href="#">DTFORM</a> 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 zur 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 zur 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)
END
```

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
```

# 88

## EMFM - Editiermasken-Frei-Modus

---

Mit diesem Natural-Profilparameter können Sie den Editiermasken-Frei-Modus beim Session-Start ein- oder ausschalten.

<b>Mögliche Werte</b>	ON	Editiermasken-Freimodus aktiviert.
	OFF	Editiermasken-Freimodus deaktiviert.
<b>Standard-Einstellung</b>	OFF	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	Innerhalb einer laufenden Natural-Session können Sie diese Einstellung mit dem Terminalkommando %FM+ bzw. %FM- überschreiben.



### Anmerkungen:

1. Dier Editiermasken-Frei-Modus ermöglicht es Ihnen, Literale bei der Eingabe in ein Feld mit einer numerischen Editiermaske wegzulassen.
2. Weitere Informationen siehe *Numerischer Editiermasken-Freimodus* in der Beschreibung des INPUT-Statements in der *Statements*-Dokumentation.



# 89

## EMU - Unicode-Editiermaske

Mit dem Session-Parameter EMU können Sie eine Unicode-Editiermaske für ein Eingabe- und/oder Ausgabefeld definieren, welches in einem der Statements verwendet wird, die in der folgenden Tabelle unter *Gültige Statements* aufgeführt sind.

<b>Mögliche Werte</b>	Die Syntax des Session-Parameters EMU ist identisch mit der Syntax des Session-Parameters EM (siehe EM - <a href="#">Syntax</a> ).	
	<b>Anmerkung:</b> Siehe auch <i>Unicode-Editiermasken – EMU-Parameter</i> im <i>Leitfaden zur Programmierung</i> .	
<b>Standard-Einstellung</b>	Keine	
<b>Gültige Statements</b>	DEFINE DATA DISPLAY INPUT PRINT WRITE MOVE EDITED PROCESS PAGE	Parameter kann auf Element-Ebene angegeben werden.
<b>Gültiges Kommando</b>	Keins	



### Anmerkungen:

1. Editiermasken, die mit EMU definiert sind, werden im Unicode-Format gehalten, so dass der Inhalt unabhängig von der installierten System-Codepage ist.
2. Weitere Informationen und ein Beispiel finden Sie in *Unicode- und Codepage-Unterstützung in der Natural-Programmiersprache, Session-Parameter*, Abschnitt *EMU, ICU, LCU, TCU im Vergleich zu EM, IC, LC, TC* in der *Unicode- und Codepage-Unterstützung-Dokumentation*.



# 90

## ENDBT - BACKOUT TRANSACTION am Session-Ende

---

Dieser Natural-Profilparameter bestimmt, ob ein implizites `BACKOUT TRANSACTION`-Statement am Ende der Natural-Session abgesetzt werden soll oder nicht.

<b>Mögliche Werte</b>	ON	Natural setzt am Ende der Session ein implizites <code>BACKOUT TRANSACTION</code> -Statement ab.
	OFF	Natural setzt am Ende der Session kein implizites <code>BACKOUT TRANSACTION</code> -Statement ab.
	ETDB	Natural setzt am Ende der Session nur für die mit dem Profilparameter <a href="#">ETDB</a> angegebene Datenbank ein implizites <code>BACKOUT TRANSACTION</code> -Statement ab.
<b>Standard-Einstellung</b>	ON	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	

Weitere transaktionsverarbeitungsbezogene Parameter: [ADAMODE](#) | [DBCLOSE](#) | [DBOPEN](#) | [ET](#) | [ETDB](#) | [ETEOP](#) | [ETIO](#) | [ETSYNC](#)



# 91

## ENDMSG - Session-Ende-Meldung anzeigen

---

Mit diesem Natural-Profilparameter können Sie die Anzeige der Standard-Mitteilung NAT9995 unterdrücken, die am Ende der Natural-Session anzeigt, dass die Session normal beendet wurde.

<b>Mögliche Werte</b>	ON	Die Meldung NAT9995 wird am Ende der Session angezeigt.
	OFF	Die Meldung NAT9995 wird am Ende der Session nicht angezeigt.
<b>Standard-Einstellung</b>	ON	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	



**Anmerkung:** Falls mit dem Profilparameter **PROGRAM** ein Session-Back-End-Programm definiert ist, hat der Profilparameter keine Wirkung; der Meldungstext wird dann an das Back-End-Programm im Parameterbereich weitergegeben und wird nicht von Natural angezeigt.



# 92 ES - Leerzeilenunterdrückung

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

<b>Mögliche Werte</b>	ON	Eine Zeile aus einem DISPLAY- oder WRITE-Statement, die alle Leerwerte enthält, wird nicht ausgegeben.  <b>Anmerkung:</b> 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.
<b>Standard-Einstellung</b>	OFF	
<b>Spezifikation in Session</b>	ja	
<b>Gültige Statements</b>	FORMAT	Parameter kann dynamisch mit dem FORMAT-Statement angegeben werden.
	DISPLAY WRITE	Parameter may be specified at statement level and/or at element level.
<b>Gültiges Kommando</b>	Keines	



## Anmerkungen:

1. 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](#).
2. Siehe auch *Parameter zur Beeinflussung der Ausgabe von Feldern im Leitfaden zur Programmierung*.

**Beispiel:**

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

# 93

## ESCAPE - Terminalkommandos %% und %. ignorieren

---

Mit diesem Natural-Profilparameter können Sie die Benutzung der Terminalkommandos %% und %. unterbinden.

<b>Mögliche Werte</b>	ON	Ermöglicht die Benutzung der Terminalkommandos %% und %..
	OFF	Die Terminalkommandos %% und %. werden ignoriert; d.h., es ist nicht mehr möglich, das zurzeit aktive Natural-Programm bzw. die Natural-Session durch Eingabe von %% oder %. zu verlassen.
<b>Standard-Einstellung</b>	ON	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	



# 94

## ESIZE - Größe des User Buffer-Erweiterungsbereichs

Dieser Natural-Profilparameter setzt die Größe des User Buffer-Erweiterungsbereichs. Er bestimmt die Größe des Natural-Source-Bereichs, der von den Natural-Editoren benutzt wird.

<b>Mögliche Werte</b>	2 - 32767	Größe des Buffer-Erweiterungsbereichs in KB. <b>Anmerkung:</b> In einer Laufzeitumgebung (in der die Editoren nicht gebraucht werden) können Sie nur einen Wert setzen, der kleiner ist als die Standard-Einstellung.
<b>Standard-Einstellung</b>	28	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	

Der User Buffer-Erweiterungsbereich enthält:

- den Sourcecode des zu kompilierenden Natural-Objekts,
- die Tabelle der zurzeit aktiven PA/PF-Tasten,
- andere, intern von Natural benutzte Tabellen und Arbeitsbereiche.



### Anmerkungen:

1. In einer Produktionsumgebung werden keine Natural-Sourcen benötigt; der ESIZE-Wert kann deshalb entsprechend verringert werden.
2. Falls dieser Bereich nicht groß genug ist, um die notwendigen Informationen aufzunehmen, wird die Fehlermeldung NAT0886 ausgegeben.



# 95

## ET - Ausführung von END TRANSACTION/BACKOUT

### TRANSACTION-Statements

---

Dieser Natural-Profilparameter gibt an, für welche Datenbanken END TRANSACTION und BACKOUT TRANSACTION-Statements ausgeführt werden sollen.

<b>Mögliche Werte</b>	ON	END TRANSACTION und BACKOUT TRANSACTION-Statements werden für alle Datenbanken ausgeführt, die seit Beginn der Natural-Session oder seit der letzten Ausführung eines END TRANSACTION- und BACKOUT TRANSACTION-Statements referenziert wurden.
	OFF	END TRANSACTION- und BACKOUT TRANSACTION-Statements werden nur für Datenbanken ausgeführt, die von der Transaktion betroffen sind (und, falls zutreffend, für die Datenbank, in die Transaktionsdaten geschrieben werden).
<b>Standard-Einstellung</b>	OFF	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	



**Anmerkung:** Aktualisierungen in einer Datenbank, die nicht unter der Kontrolle von Natural ausgeführt werden (d.h. durch natives Aufrufen der Datenbank-Link-Routinen), betreffen nicht die Natural-Transaktionslogik.

Weitere transaktionsverarbeitungsbezogene Parameter: [ADAMODE](#) | [DBCLOSE](#) | [DBOPEN](#) | [ENDBT](#) | [ETDB](#) | [ETEOP](#) | [ETIO](#) | [ETSYNC](#)



# 96

## ETA - Fehlertransaktionsprogramm

---

Dieser Natural-Profilparameter liefert den Namen des Programms, das die Kontrolle erhält, wenn bei der Ausführung eines Natural-Programms eine Fehlerbedingung festgestellt wird.

<b>Mögliche Werte</b>	1 - 8 Zeichen	Name des Programms für die Fehlertransaktion.
	leer	Bei ETA= ' ' (leer) wird kein Fehlertransaktionsprogramm aufgerufen.
<b>Standard-Einstellung</b>	leer	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	ja	
<b>Anwendungsprogrammierschnittstelle</b>	USR1041N	Mit USR1041N können Sie Ihre eigene Fehlertransaktion installieren, wobei USR1041P als Beispiel dafür dient, wie eine solche Routine erstellt werden kann.  Siehe <i>SYSEXT Utility</i> in der Dokumentation <i>Debugger und Dienstprogramme</i> .



### Anmerkungen:

1. Die Einstellung dieses Parameters kann mit Hilfe eines Benutzerprogramms mit der Systemvariablen \*ERROR-TA geändert werden bzw., wenn Natural Security installiert ist, im Natural Security Library-Profil; siehe *Components of a Library Profile* in der *Natural Security*-Dokumentation.
2. Weitere Informationen siehe *Verwendung eines Fehlertransaktionsprogramms* im *Leitfaden zur Programmierung*.



# 97

## ETDB - Datenbank für Transaktionsdaten

---

Dieser Natural-Profilparameter gibt die Datenbank an, in der Transaktionsdaten, wie sie mit einem `END TRANSACTION`-Statement geliefert werden, gespeichert werden sollen.

<b>Mögliche Werte</b>	1 - 65535, außer 255	Datenbankkennung.  <b>Anmerkung:</b> Die Datenbankkennung 255 ist für logische Systemdateien für Software AG-Produkte reserviert, siehe Profilparameter <a href="#">LFILE</a> .
	0	Die Transaktionsdaten werden in die Datenbank geschrieben, in der sich die Natural Security-Systemdatei (FSEC) befindet. Falls die <a href="#">FSEC</a> nicht angegeben ist, dann wird sie als identisch mit der Natural-Systemdatei FNAT betrachtet. Falls Natural Security nicht installiert ist, werden die Transaktionsdaten in die Datenbank geschrieben, in der sich die FNAT befindet.
<b>Standard-Einstellung</b>	0	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	

Weitere transaktionsverarbeitungsbezogene Parameter: [ADAMODE](#) | [DBCLOSE](#) | [DBOPEN](#) | [ENDBT](#) | [ET](#) | [ETEOP](#) | [ETIO](#) | [ETSYNC](#)



# 98

## ETEOP - END TRANSACTION-Statement bei

### Programmende

---

Dieser Natural-Profilparameter bestimmt, ob am Ende eines Natural-Programms (d.h., bevor der NEXT-Modus erreicht wird) ein implizites END TRANSACTION-Statement abgesetzt wird oder nicht.

<b>Mögliche Werte</b>	ON	Natural setzt am Ende eines Natural-Programms ein implizites END TRANSACTION-Statement ab.
	OFF	Natural setzt am Ende eines Natural-Programms kein implizites END TRANSACTION-Statement ab.
<b>Standard-Einstellung</b>	OFF	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	

Weitere transaktionsverarbeitungsbezogene Parameter: [ADAMODE](#) | [DBCLOSE](#) | [DBOPEN](#) | [ENDBT](#) | [ET](#) | [ETDB](#) | [ETIO](#) | [ETSYNC](#)



# 99

## ETID - Adabas-Benutzerkennung

Dieser Natural-Profilparameter wird als Kennzeichnung für Adabas-bezogene Informationen verwendet; z.B. zur Kennzeichnung von Daten, die als Ergebnis eines `END TRANSACTION`-Statements gespeichert werden.

<b>Mögliche Werte</b>	1 - 8 Zeichen	Diese Einstellung wird als Benutzerkennungs-Einstellung in einem Adabas-Open-Aufruf verwendet.
	OFF	<p>Natural setzt am Beginn der Natural-Session keine Adabas-Open- oder Close-Kommandos ab. Wenn jedoch in Natural Security ETID- und/oder OPRB-Einstellungen vorhanden sind, dann werden diese Einstellungen bei den anschließenden von Natural Security abgesetzten Open-Kommandos verwendet.</p> <p>Diese Parameter-Einstellung ist für die Verwendung in Verbindung mit Natural Security vorgesehen, um zu verhindern, dass Natural-Batch-Jobs, die zeitgleich abgeschickt werden, doppelte Benutzerkennungs-Einstellungen in einem Adabas-Open-Aufruf während der Initialisierungsphase verursacht werden.</p>
	' ' (leer)	<p>Wenn der Parameter ETID auf Leerzeichen gesetzt ist, setzt Natural keine Adabas-Open- oder Close-Kommandos ab; die Einstellungen des Parameters OPRB (falls angegeben) und die ETID- und OPRB-Einstellungen in Natural Security werden ignoriert.</p> <p>Für diesen Fall wird empfohlen, den Natural-Profilparameter <code>DBCLOSE</code> auf ON zu setzen, um am Session-Ende ein Close-Kommando zu erzwingen. Andernfalls wird er Benutzer nicht aus Adabas abgemeldet und das Adabas-Benutzer-Warteschlangenelement wird nicht gelöscht. Das kann zu einer Überlaufsituation in der Adabas-Benutzer-Warteschlange führen.</p>
<b>Standard-Einstellung</b>	*INIT-USER	
<b>Dynamische Angabe</b>	ja	

Angabe innerhalb der Session	nein	
------------------------------	------	--

**Anmerkungen:**

1. Wenn die ETID-Einstellung *nicht* mit der Einstellung in der Natural-Systemvariablen \*INIT-USER übereinstimmt, setzt Natural zu Beginn der Natural-Session ein Adabas-Open-Kommando mit der angegebenen ETID-Einstellung (und der OPRB-Einstellung, falls angegeben) ab; dieses Open-Kommando bleibt bis zum Ende der Natural-Session gültig. Eventuelle ETID- und OPRB-Einstellungen in Natural Security werden ignoriert.
2. Wenn die ETID-Einstellung mit der Einstellung in der Natural-Systemvariablen \*INIT-USER übereinstimmt oder wenn der ETID-Parameter nicht angegeben ist, setzt Natural zu Beginn der Natural-Session ein Adabas-Open-Kommando mit der \*INIT-USER-Einstellung (und der OPRB-Einstellung, falls angegeben) als ETID ab. Falls ein Natural Security-Logon (Erst-Logon oder anschließendes Logon) die zurzeit gültige ETID- oder OPRB-Einstellung (aufgrund der Library-/Benutzer-spezifischen Angaben in Natural Security) ändert, dann setzt Natural Security ein neues Open-Kommando mit den neuen ETID- und OPRB-Einstellungen ab. Falls die Einstellungen nach dem Logon dieselben bleiben, setzt Natural Security kein neues Open-Kommando ab.
3. Die Einstellungen von ETID und \*INIT-USER können beim Session-Start über den User Exit NATUEX1 geändert werden. Siehe *NATUEX1 - User Exit for Authorization Control* in der *Operations-Dokumentation*.
4. Weitere ETID-Optionen bei ETID=OFF, die in Natural Security gesetzt werden können, siehe *Library and User Preset Values* in der *Natural Security-Dokumentation*.

# 100

## ETIO - END TRANSACTION-Statement bei

### Terminal-Ein-/Ausgabe

---

Dieser Natural-Profilparameter bestimmt, ob implizite END TRANSACTION-Statements bei Terminal-Ein-/Ausgaben abgesetzt werden sollen oder nicht.

<b>Mögliche Werte</b>	ON	Natural setzt ein END TRANSACTION-Statement ab, wenn eine Terminal-Ein-/Ausgabe stattfindet.  <b>Anmerkung:</b>  1. Immer wenn ein Transaktionsmonitor die zugeordneten Datenbanken wegen einer Terminal-Ein-/Ausgabe durch Commit bestätigt, werden alle damit in Verbindung stehenden Datenbanken ebenfalls durch Commit bestätigt. Dies ist für die Synchronisierung von Datenbanktransaktionen nützlich.  2. Es ist möglich, dass Natural-Zusatzprodukte (außer Natural Security) mit ETIO=ON nicht richtig funktionieren.
	OFF	Natural setzt bei Terminal-Ein-/Ausgaben keine implizite END TRANSACTION-Statements ab.
<b>Standard-Einstellung</b>	OFF	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	

Weitere transaktionsverarbeitungsbezogene Parameter: [ADAMODE](#) | [DBCLOSE](#) | [DBOPEN](#) | [ENDBT](#) | [ET](#) | [ETDB](#) | [ETEOP](#) | [ETSYNC](#)



# 101 ETRACE - Externe Trace-Funktion

## aktivieren/deaktivieren

Mit diesem Natural-Profilparameter können Sie die (normale) externe Trace-Funktion oder die unter z/OS und TSO angebotene Generalized Trace Facility (GTF) aktivieren oder deaktivieren.

 **Vorsicht:** Benutzen Sie diesen Parameter nicht ohne vorherige Rücksprache mit dem Software AG Support.

<b>Mögliche Werte</b>	ON	Aktiviert die (normale) externe Trace-Funktion.
	OFF	Deaktiviert die (normale) externe Trace-Funktion.
	(ON , GTF) (OFF , GTF)	Aktiviert/deaktiviert die Generalized Trace Facility (GTF). Die Trace-Datensätze werden nach GTF geschrieben.
	(ON , NOGTF) (OFF , NOGTF)	Aktiviert/deaktiviert die externe Trace-Funktion.
	( , GTF)	Gleichbedeutend mit ETRACE=GTF. Trace-Daten werden nach GTF geschrieben. ON oder OFF wird nicht geändert.
<b>Standard-Einstellung</b>	OFF	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	ja	Innerhalb einer Natural-Session kann das Terminalkommando %TRE benutzt werden, um die externe Trace-Funktion (außer GTF) zu aktivieren/deaktivieren.

 **Anmerkungen:**

1. Die Trace-Funktion ist in erster Linie zur Software AG-internen Verwendung für Debugging-Zwecke gedacht. Sie schreibt in Abhängigkeit von der TP-Umgebung, in der Natural läuft, Trace-Daten in einen externen Trace Dataset.
2. In Batch- und TSO-Umgebungen ist für den externen Trace ein Dataset erforderlich (siehe auch *CMTRACE - Optional Report Output for Natural Tracing* in der *Operations* Dokumentation).



# 102

## ETSYNC - Syncpoint bei END TRANSACTION/BACKOUT

### TRANSACTION

---

Dieser Natural-Profilparameter bestimmt, ob ein impliziter Syncpoint abgesetzt wird, wenn ein END TRANSACTION- oder BACKOUT TRANSACTION-Statement abgesetzt werden soll.

<b>Mögliche Werte</b>	ON	Natural setzt ein implizites Syncpoint-COMMIT ab, wenn ein END TRANSACTION-Statement abgesetzt werden soll.  Natural setzt ein implizites Syncpoint-ROLLBACK ab, wenn ein BACKOUT TRANSACTION-Statement abgesetzt werden soll.  Dies ist nützlich bei der Synchronisierung von Datenbank-Transaktionen, die aus 3GL-Programmen heraus ausgeführt werden.
	OFF	Natural setzt keinen impliziten Syncpoint ab, wenn ein END TRANSACTION- oder BACKOUT TRANSACTION-Statement abgesetzt werden soll.
<b>Standard-Einstellung</b>	OFF	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	

#### Anmerkungen:

Zum Absetzen von Syncpoints benutzt Natural:

- Resource Recovery Services (RRS) unter TSO und in Batch-Modus, um die Wiederherstellungseinheit durch Commit oder Rollback zu bestätigen,
- CICS-Kommandos SYNCPOINT und SYNCPOINT ROLLBACK unter CICS,
- System-Dienst-Aufrufe CHECKPOINT (CHKP) und ROLLBACK (ROLB) unter IMS/TM.

Die Verarbeitungsreihenfolge ist wie folgt:

- ein END TRANSACTION / BACKOUT TRANSACTION-Statement wird an die mit den Profilparameter [ETDB](#) angegebene Datenbank abgesetzt,
- das Syncpoint COMMIT / ROLLBACK wird abgesetzt,
- END TRANSACTION- oder BACKOUT TRANSACTION-Statements werden an die übrigen Datenbanken abgesetzt.

### Einschränkungen:

- Diese Funktionalität steht unter dem Betriebssystem z/OS zur Verfügung:
  - im Batch-Modus,
  - unter dem TP-Monitor CICS,
  - unter dem TP-Monitor TSO,
  - unter dem TP-Monitor IMS/TM in einer nicht nachrichtengesteuerten BMP (in allen übrigen Umgebungen unter IMS/TM wird nur ein ROLLBACK ausgeführt, jedoch kein CHECKPOINT).
- Um Adabas-Transaktionen zu synchronisieren, muss der Adabas Transaction Manager (ATM) installiert sein.
- Für Transaktionen im Batch-Modus oder unter TSO, die in einer Db2-Datenbank gespeicherte Daten aktualisieren, müssen Sie Natural for Db2 und/oder Ihre 3GL-Anwendung so konfigurieren, dass das RRSAP Interface benutzt wird.
- Für Transaktionen im Batch-Modus, die in einer DL/I-Datenbank gespeicherte Daten aktualisieren, werden wegen DL/I-Restriktionen die Resource Recovery Services nicht unterstützt. Falls zusätzlich in einer Db2-Datenbank gespeicherte Daten in derselben Transaktion aktualisiert werden, erfolgt die Synchronisierung mittels des DL/I-Synchronisierungsmechanismus.

Als Konsequenz ist keine Synchronisierung möglich, wenn in einer Adabas-Datenbank gespeicherte Daten zusätzlich zu in Db2- oder DL/I-Datenbanken gespeicherten Daten aktualisiert werden; selbst dann nicht, wenn der Adabas Transaction Manager installiert ist.

Weitere transaktionsverarbeitungsbezogene Parameter: [ADAMODE](#) | [DBCLOSE](#) | [DBOPEN](#) | [ENDBT](#) | [ET](#) | [ETDB](#) | [ETEOP](#) | [ETIO](#)

# 103

## EXCSIZE - Puffergröße für Natural Expert C Interface

---

Dieser Natural-Profilparameter ist obsolet und wird nur noch aus Kompatibilitätsgründen akzeptiert.



# 104

## EXRSIZE - Puffergröße für Natural Expert-Regeltabellen

---

Dieser Natural-Profilparameter ist obsolet und wird nur noch aus Kompatibilitätsgründen akzeptiert.



# 105 FAMSTD - Überschreiben der

## Arbeitsdatei-Zugriffsmethoden-Zuweisungen

---

Dieser Natural-Profilparameter steuert das automatische Überschreiben der Arbeitsdatei-Zugriffsmethoden-Zuweisungen während der Session-Initialisierung gemäß der Dataset-Definition in der Job Control.



**Anmerkung:** Siehe auch Subparameter AM der Makros [NTPRINT](#) und [NETWORK](#).

<b>Mögliche Werte</b>	ON	Allen Druck- und Arbeitsdatei-Datasets wird automatisch die Batch-Zugriffsmethode AM=STD zugeordnet, wenn der logische Dataset-Name (definiert mit dem Subparameter DEST) per Job Control definiert wird (gleiches Verhalten wie bei Natural Version 2.2).
	OFF	Eine automatische Zuordnung von Druck- und Arbeitsdateien zu AM=STD erfolgt nur, wenn die Datei nicht einer anderen Zugriffsmethode zugeordnet ist, z.B. AM=NAF.  <b>Anmerkung:</b> Wenn AM=OFF angegeben ist, erfolgt keine automatische Zuordnung. Geben Sie AM=0 an, wenn Sie die Zugriffsmethode zurücksetzen und eine automatische Zuordnung zulassen wollen.
<b>Standard-Einstellung</b>	OFF	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	



# 106

## FC - Füllzeichen für INPUT-Statement

---

Dieser Natural-Profilparameter dient zur Angabe des Standard-Füllzeichens, das bei Feldern verwendet werden soll, die bei einem INPUT-Statement angezeigt werden.

Mögliche Werte	Beliebiges Zeichen	Standard-Füllzeichen.
Standard-Einstellung	X'00'	Für TTY-Modus, Web-Ein-/Ausgabemodus (über das Natural Web I/O Interface) oder Batch-Modus ist die Standard-Einstellung X'40', d.h. Leerzeichen im Hexadezimalformat.
Dynamische Angabe	ja	
Angabe innerhalb der Session	nein	



### Anmerkungen:

1. Das Standard-Füllzeichen wird zum Vorbelegen von nicht geschützten Eingabefeldern (Feldattributangabe AD=A) verwendet, wenn die Felder mit einem INPUT-Statement an ein Terminal gesendet werden.
2. Bei änderbaren Eingabefeldern (Feldattributangabe AD=M) wird es verwendet, um den Rest des Feldes aufzufüllen.



# 107

## FC - 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.

<b>Mögliche Werte</b>	beliebiges Zeichen	Füllzeichen für einzelne Überschriften.
<b>Standard-Einstellung</b>	leer	
<b>Angabe innerhalb der Session</b>	ja	
<b>Gültige Statements</b>	DISPLAY FORMAT	
<b>Gültiges Kommando</b>	keines	



### Anmerkungen:

1. 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.
2. Im Gegensatz zum Parameter [GC](#), der für Überschriften über eine Reihe von Spalten hinweg gilt, gilt der Parameter FC für einzelne Spalten.

### Beispiel:

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



# 108

## FCDP - 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).

<b>Mögliche Werte</b>	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.	
<b>Standard-Einstellung</b>	ON		
<b>Dynamische Spezifikation</b>	ja		
<b>Spezifikation in Session</b>	ja	Gültige Statements:	SET GLOBALS
		Gültiges Kommando:	GLOBALS
<b>Programmierschnittstelle (API)</b>	USR1005N	Siehe <i>SYSEXT Utility</i> in der Dokumentation <i>Debugger und Dienstprogramme</i> .	



### Anmerkungen:

1. Je nach dem Wert dieses Parameters werden dynamisch geschützte Eingabefelder entweder mit Leerzeichen oder mit den definierten Füllzeichen angezeigt.
2. In einer Natural-Session kann der Profilparameter `FCDP` durch den Session-Parameter `FCDP` überschrieben werden.

**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 \_\_\_\_\_

# 109

## FDIC - Predict-Systemdatei

---

Dieser Natural-Profilparameter definiert Datenbankkennung, Dateinummer, Passwort, Chiffrierschlüssel und Schreibschutzkennung für die Predict-Systemdatei (FDIC), die Predict zum Abfragen und/oder Speichern von Daten benutzt.

<b>Mögliche Werte</b>	Siehe <a href="#">FDIC-Parameter-Syntax</a> .	
<b>Standard-Einstellung</b>	keine	
<b>Dynamische Angabe</b>	ja	Wenn Sie den Parameter FDIC dynamisch in Verbindung mit einem der Parameter <a href="#">DBID</a> , <a href="#">FNR</a> , <a href="#">SYSPSW</a> und <a href="#">SYSCIP</a> angeben, müssen Sie den FDIC-Parameter <i>nach</i> einem dieser anderen Parameter angeben.
<b>Angabe innerhalb der Session</b>	nein	



### Anmerkungen:

1. In einer Remote-Entwicklungsumgebung wird stattdessen ein Development Server File benutzt, siehe *SPoD - Natural Single Point of Development*-Dokumentation und *Natural Development Server*-Dokumentation.
2. Informationen zu Systemdateien siehe *Natural-Systemdateien* in der *System-Architektur* -Dokumentation.

### FDIC-Parameter-Syntax

Der Profilparameter FDIC wird wie folgt angegeben:

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

Dabei ist:

Syntax-Element	Wert	Erklärung
<i>database-ID</i>	1 - 65535, außer 255	Datenbankkennung der Datenbank, in der sich die Predict-Systemdatei befindet.  <b>Anmerkung:</b> Die Datenbankkennung 255 wird für logische Systemdateien für Software AG-Produkte verwendet, siehe Parameter <a href="#">LFILE</a> .
<i>file-number</i>	1 - 65535	Dateinummer der Datei, in der sich die Predict-Systemdatei befindet.
<i>password</i>	1 bis 8 Zeichen	Das Passwort ist nur erforderlich, wenn die Predict-Systemdatei mit der Adabas Security-Funktion passwortgeschützt worden ist.  <b>Anmerkung:</b> Wenn Natural mit VSAM-Systemdateien benutzt wird, dann wird das Passwort zur Angabe des logischen Namens (DD oder DLBL) der Systemdatei wie für VSAM definiert verwendet. Beispiel: <code>FDIC=(10,5,SYSVSAM)</code>
<i>cipher-key</i>	1 bis 8 numerische Zeichen	Chiffrierschlüssel für die Predict-Systemdatei. Ist nur erforderlich, wenn die Predict-Systemdatei mit der Adabas Security-Funktion chiffriert worden ist.
RO	-	Schreibschutzoption.  <b>Anmerkung:</b>  1. RO (Read Only) gibt an, dass die Predict-Systemdatei schreibgeschützt ist.  2. RO wird nur angegeben, wenn Änderungen an der Datei verhindert werden sollen.



**Anmerkung: Standardwerte:** Wenn einer der Subparameter in der `FDIC`-Einstellung nicht angegeben wird, dann gilt die entsprechende Einstellung der Profilparameter [DBID](#), [FNR](#), [SYSPSW](#) bzw. [SYSCIP](#) für die Predict-Systemdatei.

**Beispiele:**

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

# 110

## FL - Gleitkomma-Mantissenlänge

---

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

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

### Beispiel:

```
DISPLAY FL=5 -> +1.2345E+03
```



# 111

## FNAT - Natural-Systemdatei für Systemprogramme

---

Dieser Natural-Profilparameter definiert Datenbankkennung, Dateinummer, Passwort, Chiffrierschlüssel und Schreibschutzkennung für die Natural-Systemdatei für Systemprogramme (FNAT).

<b>Mögliche Werte</b>	Siehe <i>FNAT-Parameter-Syntax</i> .	
<b>Standard-Einstellung</b>	keine	
<b>Dynamische Angabe</b>	ja	<b>Anmerkung:</b> Wenn Sie den Parameter FNAT dynamisch in Verbindung mit einem der Parameter <i>DBID</i> , <i>FNR</i> , <i>SYSPSW</i> , <i>SYSCIP</i> oder <i>ROSY</i> angeben, müssen Sie den FNAT-Parameter <i>nach</i> einem dieser Parameter angeben.
<b>Angabe innerhalb der Session</b>	nein	



### Anmerkungen:

1. Die Natural-Systemdatei ist die Datenbankdatei, aus der alle Natural-Systemprogramme abgefragt werden und mit der alle Systemkommandos arbeiten. Außerdem enthält diese Systemdatei Fehlertexte und Natural-Hilfe-Informationen.
2. Informationen zu Systemdateien siehe *Natural-Systemdateien* in der *System-Architektur* -Dokumentation.

### FNAT-Parameter-Syntax

Der Profilparameter FNAT wird wie folgt angegeben:

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

Dabei ist:

Syntax-Element	Wert	Erklärung
<i>database-ID</i>	1 - 65535, außer 255	Datenbankkennung der Datenbank, in der sich die Natural-Systemdatei befindet.  <b>Anmerkung:</b> Die Datenbankkennung 255 wird für logische Systemdateien für Software AG-Produkte verwendet, siehe Parameter <a href="#">LFILE</a> .
<i>file-number</i>	1 - 65535	Dateinummer der Datenbankdatei, in der sich die Natural-Systemdatei befindet.
<i>password</i>	1 bis 8 Zeichen	Passwort für die Natural-Systemdatei.  <b>Anmerkung:</b>  1. Ein Passwort ist nur erforderlich, wenn die Natural-Systemdatei mit der Adabas Security-Funktion passwortgeschützt worden ist. 2. Wenn Natural mit VSAM-Systemdateien benutzt wird, dann wird das Passwort zur Angabe des logischen Namens (DD oder DLBL) der Systemdatei wie für VSAM definiert verwendet. Beispiel: FNAT=(22,5,SYSVSAM)
<i>cipher-key</i>	8 numerische Zeichen	Chiffrierschlüssel für die Natural-Systemdatei.  <b>Anmerkung:</b> Ein Chiffrierschlüssel ist nur erforderlich, wenn die Natural-Systemdatei mit der Adabas Security-Funktion chiffriert worden ist.
RO	-	Schreibschutzoption.  <b>Anmerkung:</b>  1. RO (Read Only) gibt an, dass die Predict-Systemdatei schreibgeschützt ist. 2. RO wird nur angegeben, wenn Änderungen an der Datei verhindert werden sollen.



**Anmerkungen:**

- Standardwerte:** Wenn einer der Subparameter in der FNAT-Einstellung nicht angegeben wird, dann gilt die entsprechende Einstellung der Profilparameter [DBID](#), [FNR](#), [SYSPSW](#), [SYSCIP](#) bzw. [ROSY](#) für die Natural-Systemdatei.

2. Wenn Sie eine FNAT-Datei reorganisieren oder wenn Sie Daten von der oder in die FNAT-Datei entladen oder laden (z.B. mit der ADAULD/ADALOD-Utility), müssen Sie für die ADALOD-Utility USERISN=YES angeben.

**Beispiele:**

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



# 112

## FNR - Standard-Dateinummer der Natural-Systemdateien

Dieser Natural-Profilparameter gibt die Standard-Dateinummer der Datenbankdatei an, in der sich Natural-Systemdateien (FNAT, FUSER, FDIC, FSEC, FSPPOOL, FPROF, FREG) befinden.

<b>Mögliche Werte</b>	1 - 65535	Die Standard-Dateinummer.
<b>Standard-Einstellung</b>	keine	
<b>Dynamische Angabe</b>	ja	<b>Anmerkung:</b> Wenn Sie den Profilparameter FNR dynamisch in Verbindung mit einem der Profilparameter angeben, die die Systemdateien <a href="#">FNAT</a> , <a href="#">FUSER</a> , <a href="#">FDIC</a> , <a href="#">FSEC</a> , <a href="#">FSPPOOL</a> , <a href="#">FPROF</a> und <a href="#">FREG</a> definieren, müssen Sie den FNR-Parameter <i>vor</i> dem Systemdatei-Parameter angeben.
<b>Angabe innerhalb der Session</b>	ja	



### Anmerkungen:

1. Die Standard-Dateinummer gilt für alle Natural-Systemdateien, für die keine individuellen Datenbankdateinummern angegeben sind.
2. Individuelle Dateinummern für einzelne Systemdateien können Sie mit dem Subparameter *file-number* der Profilparameter [FNAT](#), [FUSER](#), [FDIC](#), [FSEC](#), [FSPPOOL](#), [FPROF](#) und [FREG](#) angeben.

### Beispiele:

#### Beispiel 1:

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

In diesem Beispiel wird die Systemdatei für Benutzerprogramme der Datei 8 in der Datenbank 10 zugeordnet. Alle anderen Systemdateien werden der Datei 5 in Datenbank 10 zugeordnet.

#### Beispiel 2:

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

In diesem Beispiel werden alle Systemdateien der Datei 5 in Datenbank 10 zugeordnet.

Dieser Natural-Profilparameter kann verwendet werden, um eine Systemdatei für Parameterprofile (FPROF) anzugeben.

<b>Possible settings</b>	Siehe <i>FPROF-Parameter-Syntax</i> .	
<b>Default setting</b>	none	
<b>Dynamic specification</b>	yes	<b>Anmerkung:</b> Wenn Sie den Profilparameter FPROF dynamisch in Verbindung mit einem der Profilparameter <a href="#">DBID</a> , <a href="#">FNR</a> , <a href="#">SYSPSW</a> oder <a href="#">SYSCIP</a> angeben, müssen Sie den FPROF-Parameter <i>nach</i> einem dieser Profilparameter angeben.
<b>Specification within session</b>	no	



#### Anmerkungen:

1. Die Systemdatei FPROF wird benutzt, um Parameterprofile zu lesen, die mit dem Profilparameter [PROFILE](#) angegeben werden. Dies setzt voraus, dass keine Datenbankinformationen als Subparameter des Profilparameters PROFILE mitgeliefert werden.
2. Wird FPROF nicht definiert, dann wird die Systemdatei [FNAT](#) als Systemdatei für Parameterprofile benutzt.
3. Parameterprofile können mit der Utility SYSPARM verwaltet werden.
4. Informationen zu Systemdateien siehe *Natural-Systemdateien* in der *System-Architektur* -Dokumentation.

#### FPROF-Parameter-Syntax

Der Profilparameter FPROF wird wie folgt angegeben:

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

Dabei ist:

Syntax-Element	Wert	Erklärung
<i>database-ID</i>	1 - 65535, außer 255	Datenbankkennung der Datenbank, in der sich die Natural-Systemdatei für Parameterprofile befindet.  <b>Anmerkung:</b> Die Datenbankkennung 255 wird für logische Systemdateien für Software AG-Produkte verwendet, siehe Parameter <a href="#">LFILE</a> .
<i>file-number</i>	1 - 65535	Dateinummer der Datenbankdatei, in der sich die Natural-Systemdatei für Parameterprofile befindet.
<i>password</i>	1 - 8 characters	Passwort für die Natural-Systemdatei FPROF.  <b>Anmerkung:</b>  1. Ein Passwort ist nur erforderlich, wenn die Natural-Systemdatei für Parameterprofile mit der Adabas Security-Funktion passwortgeschützt worden ist.  2. Wenn Natural mit VSAM-Systemdateien benutzt wird, dann wird das Passwort zur Angabe des logischen Namens (DD oder DLBL) der Systemdatei wie für VSAM definiert verwendet. Beispiel: FPROF=(10,33,SYSVSAM)
<i>cipher-key</i>	8 characters	Chiffrierschlüssel für die Natural-Systemdatei für Parameterprofile FPROF.  <b>Anmerkung:</b> Ein Chiffrierschlüssel ist nur erforderlich, wenn die Natural-Systemdatei mit der Adabas Security-Funktion chiffriert worden ist.
RO	-	Schreibschutzoption  <b>Anmerkung:</b>  1. RO (Read Only) gibt an, dass die Natural-Systemdatei schreibgeschützt ist.  2. RO wird nur angegeben, wenn Änderungen an der Datei verhindert werden sollen. Dies würde zum Beispiel bedeuten, dass es nicht möglich ist, Profile mit der Utility SYSPARM in der Systemdatei zu speichern.



**Anmerkung: Standardwerte:** Wenn einer der Subparameter des Profilparameters FPROF nicht angegeben wird, dann gilt die entsprechende Einstellung der Profilparameter [DBID](#), [FNR](#), [SYSPSW](#) bzw. [SYSCIP](#) für die Natural-Systemdatei für Parameterprofile.

**Beispiel:**

FPROF=(10,56)

In diesem Beispiel befindet sich die Systemdatei FPROF in Systemdatei 10 in der Datenbank 56.



# 114

## FREEGDA - GDA im Utility-Modus freigeben

Dieser Natural-Profilparameter steuert, ob die aktuelle Global Data Area (GDA) und die anwendungsspezifischen Variablen (AIV) zurückgesetzt werden sollen oder nicht, wenn eine Utility im Utility-Modus aufgerufen wird (siehe *Utility Activation* in der *Utilities*-Dokumentation), d.h., bei Benutzung des Systemkommandos, das dem Namen der Utility entspricht.

<b>Mögliche Werte</b>	ON	Die aktuelle GDA und die aktuellen AIV-Variablen werden zurückgesetzt, bevor eine Utility gestartet wird.  <b>Anmerkung:</b> Dieses Verhalten entspricht der früheren Situation, als die Utility mit dem Systemkommando LOGON <i>library-name</i> aufgerufen wurde.
	OFF	Die aktuelle GDA und die aktuellen AIV-Variablen werden beibehalten, wenn eine Utility gestartet wird.  <b>Anmerkung:</b> Beachten Sie bitte, dass dies die Datengröße entsprechend erhöht und zu Thread-Problemen unter bestimmten Betriebssystemen führen.
<b>Standard-Einstellung</b>	ON	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	



# 115

## FREG - Natural-Registry-Systemdatei

---

Dieser Natural-Profilparameter definiert Datenbankkennung, Dateinummer, Passwort und Chiffrierschlüssel für die Natural-Registry-Systemdatei (FREG).

Die Natural-Registry-Systemdatei ist die Datenbankdatei, die zum Speichern der Registry-Informationen zu Natural-Sessions verwendet wird, welche mit einem Wert größer als Null für den Profilparameter `UCONMAX` gestartet werden.

<b>Mögliche Werte</b>	Siehe <i>FREG-Parameter-Syntax</i> .	
<b>Standard-Einstellung</b>	keine	
<b>Dynamische Angabe</b>	ja	<b>Anmerkung:</b> Wenn Sie den Parameter FREG dynamisch in Verbindung mit einem der Parameter <code>DBID</code> , <code>FNR</code> , <code>SYSPSW</code> oder <code>SYSCIP</code> angeben, müssen Sie den FREG-Parameter nach einem dieser Parameter angeben.
<b>Angabe innerhalb der Session</b>	nein	

### FREG-Parameter-Syntax

Der Profilparameter FREG wird wie folgt angegeben:

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

Dabei ist:

Syntax-Element	Wert	Erklärung
<i>database-ID</i>	1 - 65535, außer 255	Datenbankkennung der Datenbank, in der sich die Natural-Registry-Systemdatei befindet.  <b>Anmerkung:</b> Die Datenbankkennung 255 wird für logische Systemdateien für Software AG-Produkte verwendet, siehe Parameter <a href="#">LFILE</a> .
<i>file-number</i>	1 - 65535	Dateinummer der Datenbankdatei, in der sich die Natural-Registry-Systemdatei befindet.
<i>password</i>	1 - 8 Zeichen	Passwort für die Natural-Registry-Systemdatei.  <b>Anmerkung:</b>  1. Ein Passwort ist nur erforderlich, wenn die Natural-Registry-Systemdatei mit der Adabas Security-Funktion passwortgeschützt worden ist.  2. Wenn Natural mit VSAM-Systemdateien benutzt wird, dann wird das Passwort zur Angabe des logischen Namens (DD oder DLBL) der Systemdatei wie für VSAM definiert verwendet. Beispiel: FNAT=(22,5,SYSVSAM)
<i>cipher-key</i>	8 numerische Zeichen	Chiffrierschlüssel für die Natural-Registry-Systemdatei.  <b>Anmerkung:</b> Ein Chiffrierschlüssel ist nur erforderlich, wenn die Natural-Registry-Systemdatei mit der Adabas Security-Funktion chiffriert worden ist.



**Anmerkungen:**

1. Wenn einer der Subparameter in der FREG-Einstellung nicht angegeben wird, dann gilt die entsprechende Einstellung der Profilparameter [DBID](#), [FNR](#), [SYSPSW](#) bzw. [SYSCIP](#) für die Natural-Registry-Systemdatei.
2. Die Systemdatei FREG darf sich nicht in einer schreibgeschützten Datenbank befinden.
3. Falls die Systemdatei FREG nicht deklariert worden ist, benutzt Natural zum Registrieren der Session die Systemdatei [FSEC](#). Falls darüber hinaus auch die Systemdatei FSEC nicht deklariert ist, benutzt Natural zum Registrieren der Session die Systemdatei [FNAT](#).

**Beispiele:**

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



# 116

## FS - Format-Spezifikation für Benutzervariablen

Mit diesem Natural Profil- und Session-Parameter bestimmen Sie, ob für die Definition von Benutzervariablen im Reporting Mode Standardformat und -länge gelten sollen.



**Anmerkung:** Siehe auch *Format und Länge von Benutzervariablen* im Leitfaden zur Programmierung.

<b>Mögliche Werte</b>	ON	Einer neuen Benutzervariablen wird im Reporting Mode von Natural kein/e Standardformat/-länge zugeordnet.  <b>Anmerkung:</b> 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.	
<b>Standard-Einstellung</b>	OFF		
<b>Dynamische Angabe</b>	ja		
<b>Angabe innerhalb der Session</b>	ja	Gültige Statements:	SET GLOBALS
		Gültiges Kommando:	GLOBALS
<b>Programmierschnittstelle (API)</b>	USR1005N	Siehe <i>SYSEXT Utility</i> in der Dokumentation <i>Debugger und Dienstprogramme</i> .	



### Anmerkungen:

1. Dieser Natural Profil- und Session-Parameter gilt nur für den Reporting Mode; er hat keine Auswirkungen im Structured Mode.
2. In einer Natural-Session kann der Profilparameter FS durch den Session-Parameter FS überschrieben werden.
3. Unter Natural Security: Die Einstellung dieses Parameters kann durch die Session-Parameter-Option des Library-Profiles überschrieben werden.



Dieser Natural-Profilparameter definiert Datenbankkennung, Dateinummer, Passwort, Chiffrierschlüssel und Schreibschutzkennung für die Natural Security-Systemdatei (FSEC), die Natural Security zum Abfragen und/oder Speichern von Security-Daten benutzt.

<b>Mögliche Werte</b>	Siehe <i>FSEC-Parameter-Syntax</i> .	
<b>Standard-Einstellung</b>	keine	
<b>Dynamische Angabe</b>	ja	<b>Anmerkung:</b> Wenn Sie den Profilparameter FSEC dynamisch in Verbindung mit einem der Profilparameter <i>DBID</i> , <i>FNR</i> , <i>SYSPSW</i> , <i>SYSCIP</i> oder <i>ROSY</i> angeben, müssen Sie den FSEC-Parameter <i>nach</i> einem dieser Parameter angeben.
<b>Angabe innerhalb der Session</b>	nein	



#### Anmerkungen:

1. Dieser Natural-Profilparameter gilt nur, wenn Natural Security verwendet wird.
2. Informationen zu Systemdateien siehe *Natural-Systemdateien* in der *System-Architektur* -Dokumentation.

#### FSEC-Parameter-Syntax

Der Profilparameter FSEC wird wie folgt angegeben:

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

Dabei ist:

Syntax-Element	Wert	Erklärung
<i>database-ID</i>	1-65535, außer 255	Datenbankkennung der Datenbank, in der sich die Natural Security-Systemdatei befindet.  <b>Anmerkung:</b> Die Datenbankkennung 255 wird für logische Systemdateien für Software AG-Produkte verwendet, siehe Parameter <a href="#">LFILE</a> .
<i>file-number</i>	1-65535	Dateinummer der Datenbankdatei, in der sich die Natural Security-Systemdatei befindet.
<i>password</i>	1 - 8 Zeichen	Passwort für die Natural Security-Systemdatei.  <b>Anmerkung:</b> Ein Passwort ist nur erforderlich, wenn die Natural Security-Systemdatei mit der Adabas Security-Funktion passwortgeschützt worden ist.
<i>cipher-key</i>	8 numerische Zeichen	Chiffrierschlüssel für die Natural Security-Systemdatei.  <b>Anmerkung:</b> Ein Chiffrierschlüssel ist nur erforderlich, wenn die Natural Security-Systemdatei mit der Adabas Security-Funktion chiffriert worden ist.
RO	-	Schreibschutzoption.  <b>Anmerkung:</b>  1. RO(Read Only) gibt an, dass die Natural Security-Systemdatei schreibgeschützt ist. 2. RO wird nur angegeben, wenn Änderungen an der Natural Security-Systemdatei verhindert werden sollen.



**Anmerkung:** Wenn einer der Subparameter des Profilparameters FSEC nicht angegeben wird, dann gilt die entsprechende Einstellung der Profilparameter [DBID](#), [FNR](#), [SYSPSW](#), [SYSCIP](#) oder [ROSY](#) für die Natural Security-Systemdatei.

**Beispiel:**

FSEC=(10,8)



# 118

## FSIZE (für interne Verwendung)

---

Dieser Parameter ist für die interne Nutzung durch Natural reserviert.



**Vorsicht:** Ändern Sie seine Einstellung nicht.



Dieser Natural-Profilparameter definiert die fünf Subparameter für die Natural Advanced Facilities-Spool-Datei.

<b>Mögliche Werte</b>	Siehe <i>FSPool-Parameter-Syntax</i> .	
<b>Standard-Einstellung</b>	keine	
<b>Dynamische Angabe</b>	ja	<b>Anmerkung:</b> Wenn Sie den Profilparameter FSPool dynamisch in Verbindung mit einem der Profilparameter <i>DBID</i> , <i>FNR</i> , <i>SYSPSW</i> oder <i>SYSCIP</i> angeben, müssen Sie den FSEC-Parameter <i>nach</i> einem dieser Parameter angeben.
<b>Angabe innerhalb der Session</b>	nein	



#### Anmerkungen:

1. Dieser Natural-Profilparameter gilt nur bei Natural Advanced Facilities.
2. Die Spool-Datei ist die Datenbankdatei, die von Natural Advanced Facilities benutzt wird.
3. Als Spool-Datei ist eine andere Datei als die Systemdateien *FNAT*, *FUSER*, *FDIC*, *FSEC* und *FPROF* erforderlich.
4. Informationen zu Systemdateien siehe *Natural-Systemdateien* in der *System-Architektur* -Dokumentation.

#### FSPool-Parameter-Syntax

Der Profilparameter FSPool wird wie folgt angegeben:

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

Dabei ist:

Syntax-Element	Wert	Erklärung
<i>database-ID</i>	1 - 65535, außer 255	Datenbankkennung der Datenbank, in der sich die Spool-Datei befindet.  <b>Anmerkung:</b> Die Datenbankkennung 255 wird für logische Systemdateien für Software AG-Produkte verwendet, siehe Parameter <a href="#">LFILE</a> .
<i>file-number</i>	1 - 65535	Dateinummer der Datenbankdatei, in der sich die Spool-Datei befindet.
<i>password</i>	1 - 8 Zeichen	Passwort für die Spool-Datei.  <b>Anmerkung:</b>  1. Ein Passwort ist nur erforderlich, wenn die Spool-Datei mit der Adabas Security-Funktion passwortgeschützt worden ist. 2. Wenn Natural mit VSAM-Systemdateien benutzt wird, dann wird das Passwort zur Angabe des logischen Namens (DD oder DLBL) der Systemdatei wie für VSAM definiert verwendet. Beispiel: FSPPOOL=(10,8,SYSVSAM)
<i>cipher-key</i>	8 Zeichen	Chiffrierschlüssel für die Spool-Datei.  <b>Anmerkung:</b> Ein Chiffrierschlüssel ist nur erforderlich, wenn die Spool-Datei mit der Adabas Security-Funktion chiffriert worden ist.
RO	-	Schreibschutzoption.  <b>Anmerkung:</b>  1. RO(Read Only) gibt an, dass die Spool-Datei schreibgeschützt ist. 2. RO wird nur angegeben, wenn Änderungen an der Spool-Datei verhindert werden sollen. Das würde beispielsweise bedeuten, dass keine Reports in der Spool-Datei gespeichert werden können.

**Beispiel:**

FSPPOOL=(10,8)

# 120

## FUSER - Natural-Systemdatei für Benutzerprogramme

---

Dieser Natural-Profilparameter definiert Datenbankkennung, Dateinummer, Passwort, Chiffrierschlüssel und Schreibschutzkennung für die Natural-Systemdatei für Benutzerprogramme (FUSER).

<b>Mögliche Werte</b>	Siehe <i>FUSER-Parameter-Syntax</i> .	
<b>Standard-Einstellung</b>	keine	
<b>Dynamische Angabe</b>	ja	<b>Anmerkung:</b> Wenn Sie den Profilparameter FUSER dynamisch in Verbindung mit einem der Profilparameter <i>DBID</i> , <i>FNR</i> , <i>SYSPSW</i> , <i>SYSCIP</i> oder <i>ROSY</i> angeben, müssen Sie den FUSER-Parameter <i>nach</i> einem dieser Parameter angeben.
<b>Angabe innerhalb der Session</b>	nein	



### Anmerkungen:

1. Die Natural-Systemdatei für Benutzerprogramme (FUSER) ist die Datenbankdatei, aus der alle vom Benutzer geschriebenen Natural-Programme gelesen werden.
2. Informationen zu Systemdateien siehe *Natural-Systemdateien* in der *System-Architektur* -Dokumentation.

### FUSER-Parameter-Syntax

Der Profilparameter FUSER wird wie folgt angegeben:

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

Dabei ist:

Syntax-Element	Wert	Erklärung
<i>database-ID</i>	1-65535, außer 255	Datenbankkennung der Datenbank, in der sich die Natural-Systemdatei für Benutzerprogramme befindet.  <b>Anmerkung:</b> Die Datenbankkennung 255 wird für logische Systemdateien für Software AG-Produkte verwendet, siehe Parameter <a href="#">LFILE</a> .
<i>file-number</i>	1-65535	Dateinummer der Datenbankdatei, in der sich die Natural-Systemdatei für Benutzerprogramme befindet.
<i>password</i>	1-8 Zeichen	Passwort für die Natural-Systemdatei für Benutzerprogramme.  <b>Anmerkung:</b>  1. Ein Passwort ist nur erforderlich, wenn die Natural-Systemdatei für Benutzerprogramme mit der Adabas Security-Funktion passwortgeschützt worden ist.  2. Wenn Natural mit VSAM-Systemdateien benutzt wird, dann wird das Passwort zur Angabe des logischen Namens (DD oder DLBL) der Systemdatei wie für VSAM angegeben. Beispiel: FUSER=(22,5,SYSVSAM)
<i>cipher-key</i>	8 numerische Zeichen	Chiffrierschlüssel für die Natural-Systemdatei für Benutzerprogramme.  <b>Anmerkung:</b> Ein Chiffrierschlüssel ist nur erforderlich, wenn die Natural-Systemdatei für Benutzerprogramme mit der Adabas Security-Funktion chiffriert worden ist.
RO	-	Schreibschutzoption.  <b>Anmerkung:</b>  1. RO(Read Only) gibt an, dass die Natural-Systemdatei für Benutzerprogramme schreibgeschützt ist.  2. RO wird nur angegeben, wenn Änderungen an der Natural-Systemdatei für Benutzerprogramme verhindert werden sollen.



**Anmerkung:** Wenn einer der Subparameter des Profilparameters FUSER nicht angegeben wird, dann gilt die entsprechende Einstellung der Profilparameter [DBID](#), [FNR](#), [SYSPSW](#), [SYSCIP](#) bzw. [ROSY](#) für die Natural-Systemdatei für Benutzerprogramme.

**Beispiele:**

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



# 121

## GC - 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.

<b>Mögliche Werte</b>	beliebiges Zeichen	Füllzeichen für Gruppenüberschriften.
<b>Standard-Einstellung</b>	leer	
<b>Angabe innerhalb der Session</b>	ja	
<b>Gültige Statements</b>	<code>DISPLAY</code> <code>FORMAT</code>	
<b>Gültiges Kommando</b>	keines	

### Beispiel:

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



# 122 HC - Überschriften-Zentrierung

---

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

<b>Mögliche Werte</b>	C	Spaltenüberschriften werden zentriert ausgegeben.
	L	Spaltenüberschriften werden linksbündig ausgegeben.
	R	Spaltenüberschriften werden rechtsbündig ausgegeben.
<b>Standard-Einstellung</b>	C	
<b>Angabe innerhalb der Session</b>	ja	
<b>Gültige Statements</b>	DISPLAY FORMAT	
<b>Gültiges Kommando</b>	keines	

## Beispiel:

```
DISPLAY (HC=L)
```

---



# 123

## HCAM - Hardcopy-Zugriffsmethode

Dieser Natural-Profilparameter gibt an, welche Zugriffsmethode bei der Verarbeitung von Hardcopy-Ausgaben verwendet werden soll.

<b>Mögliche Werte</b>	Mit <code>HCAM=value</code> können Sie einen der folgenden Zugriffsmethodennamen angeben:	
	<b>Wert:</b>	<b>Zugriffsmethode:</b>
	STD	Sequenzielle Standard-Datei (Batch, TSO).
	COMP	Com-plete-Druckdatei.
	CICS	CICS Transient Data oder temporäre Speicher.
	IMS	IMS/TM-Drucker.
	NAF	Natural Advanced Facilities.
	USER	Dritthersteller-Druckerschnittstelle.
	SMARTS	SMARTS-Druckdatei.
	ESS	Entire System Server.
	ANY	Die Hardcopy-Ausgabeverarbeitung erfolgt gemäß der ersten zur Verfügung stehenden Zugriffsmethode (die Suchreihenfolge nach verfügbaren Zugriffsmethoden entspricht der Reihenfolge, in der die Zugriffsmethoden hier angegeben sind).
OFF	Die Hardcopy-Ausgabeverarbeitung erfolgt nach keiner Zugriffsmethode.	
<b>Standard-Einstellung</b>	ANY	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	



**Anmerkungen:**

1. `HCAM=value` ist gleichbedeutend mit dem Subparameter `AM` des Profilparameters `PRINT` für Druckdatei 0, d.h. `PRINT=((0),AM=value)`.
2. Das Hardcopy-Ausgabeziel wird mit dem Profilparameter `HCDEST` angegeben. Weitere Angaben zur Hardcopy-Ausgabedatei können mit dem Profilparameter `PRINT` bzw. dem Makro `NTPRINT` für den Drucker 0 gemacht werden.

# 124

## HCDEST - Ziel für Hardcopy-Ausgabe

Dieser Natural-Profilparameter dient zum Voreinstellen des Ziels für die Hardcopy-Ausgabe mittels des Terminalkommandos %H (ohne den Operanden *destination*).

<b>Mögliche Werte</b>	1 - 8 Zeichen	Gültiges Ziel für Hardcopy-Ausgabe.
	leer	<b>Anmerkung:</b> In manchen Umgebungen kann ein Standard-Ziel vom Betriebssystem oder den TP-Monitor geliefert werden. Bei Angabe von <code>HCAM=STD</code> für die Hardcopy ist das Standard-Ziel für die Hardcopy-Ausgabe der Dataset CMHCOPY.
<b>Standard-Einstellung</b>	leer	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	ja	<b>Anmerkung:</b> Während der Session kann das Ziel für die Hardcopy-Ausgabe durch Angabe von %H <i>destination</i> überschrieben werden; siehe auch Terminalkommando %H.



### Anmerkungen:

1. `HCDEST=value` ist gleichbedeutend mit dem Subparameter `DEST` des Profilparameters `PRINT` für die Druckdatei 0, d.h. `PRINT=((0),DEST=value)`.
2. Wenn Sie Natural unter TSO oder in Batch-Modus laufen lassen, muss der Dataset in der JCL oder durch dynamische Zuweisung definiert werden.
3. Unter TSO wird der mit `HCDEST` angegebene Hardcopy-Dataset nach Absetzen von %H bei der nächsten Terminal-Ein-/Ausgabe geschlossen. Der Standard-CMHCOPY-Dataset wird nicht bei einer Terminal-Ein-/Ausgabe, sondern bei Beendigung der Session geschlossen.
4. Das Hardcopy-Zugriffsmethode wird mit dem Profilparameter `HCAM` oder mit dem Subparameter `DEST` des Profilparameters `PRINT` für den Drucker 0 angegeben. Weitere Angaben zur Hardcopy-Ausgabedatei können mit dem Profilparameter `PRINT` bzw. dem Makro `NTPRINT` für den Drucker 0 gemacht werden.



# 125

## HD - 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.

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

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*).



# 126 HE - Helproutine

---

▪ HE-Parameter-Syntax .....	392
▪ Ausführung von Helproutinen .....	394
▪ Beispiele .....	395

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

<b>Mögliche Werte</b>		Siehe <i>HE-Parameter-Syntax</i> weiter unten.
<b>Standard-Einstellung</b>	keine	
<b>Angabe innerhalb der Session</b>	ja	
<b>Gültige Statements</b>	INPUT	
<b>Gültiges Kommando</b>	keines	

Helproutinen können mit dem Natural-Programm-Editor erstellt werden, Hilfe-Maske (Help Maps) mit dem Natural-Masken-Editor (Map Editor).

Um die zugewiesene Helproutine oder Hilfe-Maske dann bei der Verarbeitung des betreffenden INPUT-Statements oder der betreffenden Maske aufzurufen, können Sie eine der folgenden Vorgehensweisen benutzen:

- Geben Sie in das Feld, für das die Hilfe angefordert werden soll, das Hilfe-Aufrufzeichen an der ersten Stelle links im Feld ein und drücken Sie ENTER. Das Standard-Hilfe-Aufrufzeichen ist ein Fragezeichen (?).

Wenn Sie das Hilfe-Aufrufzeichen an einer anderen Stelle im Feld eingeben oder wenn Sie mehr als ein Zeichen eingeben, wird diese Zeichenkette als Benutzereingabe übernommen, d.h. es wird keine Hilfe aufgerufen. Wenn das Feld Nullen im Hexadezimalformat enthält, hängt es von der verwendeten Terminal-Emulation ab, ob Natural die Werte als Hilfe-Anforderung interpretieren kann.

Oder:

- Stellen Sie den Cursor in das Feld und drücken Sie die mit dem SET KEY-Statement als Hilfe-Funktionstaste definierte PF-Taste.

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-Element-Beschreibung:

Syntax-Element	Beschreibung
<i>operand1</i>	<p><i>operand1</i> ist der Name der Helproutine oder Hilfe-Maske (Help 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 (&amp;) 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 Hilfe-Masken.</p> <p>Weitere Informationen zum <i>operand1</i> in einer Maske siehe HE Helproutine-Option im Abschnitt <i>Erweiterte Feld-Bearbeitung in Masken-Editor (Map Editor)</i> in der <i>Editoren-Dokumentation</i>.</p>
<i>operand2</i>	<p>Sie können 1 bis 20 Parameter (<i>operand2</i>) angeben, die an die Helproutine oder Hilfe-Maske übergeben werden. Diese können als Konstanten oder Benutzervariablen angegeben werden, die die Parameterwerte enthalten.</p> <p>Weitere Informationen zum <i>operand2</i> in einer Maske siehe HE Helproutine-Option im Abschnitt <i>Erweiterte Feld-Bearbeitung in Masken-Editor (Map Editor)</i> in der <i>Editoren-Dokumentation</i>.</p>
=	<p>Das Gleichheitszeichen (=) wird verwendet, um den Namen eines Objekts oder Feldes an die Helproutine oder Hilfe-Maske zu übergeben.</p> <ul style="list-style-type: none"> <li>■ Wird das Gleichheitszeichen in der HE=-Angabe auf Statement-Ebene eingegeben, wird der Name des zurzeit ausgeführten Objekts (so wie er in der Systemvariablen *PROGRAM enthalten ist) an die Helproutine oder Hilfe-Maske übergeben. In <b>Beispiel 3</b> lautet der übergebene Objektname PROGRAM1.</li> <li>■ Wird das Gleichheitszeichen in der HE=-Angabe auf Feldebene eingegeben, wird der Name des Feldes an die Helproutine oder Hilfe-Maske übergeben. In <b>Beispiel 3</b> lautet der übergebene Feldname #PARM1.</li> </ul> <p>Wird mit dem Gleichheitszeichen (=) ein Feld- bzw. Masken-Name übergeben, so muss der entsprechende Parameter in der Helproutine mit Format/Länge A65 definiert werden.</p>

Syntax-Element	Beschreibung
$nX$	Die Notation $nX$ können Sie verwenden, um wegzulassende Parameter anzugeben, das heisst, Parameter, für die keine Werte übergeben werden sollen. Die entsprechenden empfangsseitigen Parameter im <code>DEFINE DATA PARAMETER</code> -Statement der aufgerufenen Helproutine müssen als <code>OPTIONAL</code> definiert werden.



#### Anmerkungen:

1. Die Operanden müssen entweder mit dem Eingabe-Begrenzungszeichen (Input-Delimiter, 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 (Decimal Character, wie mit dem Session-Parameter `DC` definiert) verwendet wird.
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 in dem Felder-Block immer zuletzt nach den Parametern definiert werden.
4. Ist das Feld, für das eine 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) zum 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 des Hilfe-Aufrufzeichens, d.h. standardmäßig eines Fragezeichens (?), in das Feld oder durch Drücken der (mit einem `SET KEY`-Statement definierten) Hilfe-Funktionstaste, oder über ein `REINPUT USING HELP`-Statement — aufgerufen wird, werden alle Werte, die bereits in andere Felder eingegebenen worden sind, erst dann verarbeitet, nachdem die Verarbeitung aller Hilfe-Anforderungen 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-Helprountine HELP1:

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

# 127

## HI - Hilfe-Aufrufzeichen

Dieser Natural-Profilparameter definiert das Zeichen, welches benutzt werden soll, um eine feldspezifische Helproutine oder eine Masken-Helproutine (falls eine für die gegebene Maske definiert ist) aufzurufen.

Mögliche Werte	Ein beliebiges Sonderzeichen	<p>Das Zeichen, das zum Aufrufen einer feldspezifischen Helproutine oder einer Masken-Helproutine benutzt werden soll.</p> <p><b>Anmerkung:</b></p> <ol style="list-style-type: none"><li>1. Das mit dem Profilparameter HI angegebene Zeichen darf nicht identisch mit dem Zeichen sein, welches mit dem Profil- bzw. Session-Parameter CF (Steuerzeichen für Großrechner-Terminalkommandos) angegeben wurde.</li><li>2. Es sollte auch nicht identisch sein mit dem Zeichen, das mit dem Profil- bzw. Session-Parameter DC (Dezimalzeichen), mit dem Profil- bzw. Session-Parameter IA (Input-Zuweisungszeichen) oder dem Profil- bzw. Session-Parameter ID (Input-Delimiterzeichen) zugewiesen worden ist.</li></ol>
	leer	<p><b>Anmerkung:</b></p> <ol style="list-style-type: none"><li>1. Numerische Felder, denen eine Helproutine zugewiesen ist, werden intern in alphanumerisches Format umgesetzt, damit der Benutzer in das Feld ein Fragezeichen eingeben kann, um die Helproutine aufzurufen.</li><li>2. Um diese interne Formatumsetzung zu verhindern (d.h., wenn Sie sicherstellen wollen, dass keine</li></ol>

		<p>alphanumerischen Zeichen in ein numerisches Feld eingegeben werden können), können Sie den Profilparameter HI auf Leerzeichen setzen.</p> <p>3. Wenn HI= ' ' gesetzt ist, muss in der Natural-Anwendung eine Hilfe-Taste definiert werden, indem das Statement SETKEY entsprechend verwendet wird; andernfalls gibt es keine Möglichkeit, eine Helproutine für ein Feld aufzurufen.</p>
<b>Standard-Einstellung</b>	?	Fragezeichen.
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	
<b>Anwendungsprogrammierschnittstelle</b>	USR0350N	Siehe <i>SYSEXT Utility</i> in der Dokumentation <i>Debugger und Dienstprogramme</i> .

# 128

## HW - Überschriftenbreite

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

<b>Mögliche Werte</b>	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.  <b>Anmerkung:</b> Dies gilt auch, wenn kein Überschriftentext ausgegeben wird, entweder weil das <code>DISPLAY</code> -Statement das Schlüsselwort <code>NOHDR</code> enthält oder weil 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.  <b>Anmerkung:</b> <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).
<b>Standard-Einstellung</b>	ON	
<b>Angabe innerhalb der Session</b>	ja	
<b>Gültige Statements</b>	<code>DISPLAY</code> <code>FORMAT</code>	
<b>Gültiges Kommando</b>	keines	

**Beispiel:**

DISPLAY (HW=OFF)

# 129

## IA - Input-Zuweisungszeichen

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

<b>Mögliche Werte</b>	jedes Sonderzeichen	Zuweisungszeichen für die Verarbeitung des Eingabeparameters in INPUT-Statements.	
<b>Standard-Einstellung</b>	=	Gleichheitszeichen.	
<b>Dynamische Angabe</b>	ja		
<b>Angabe innerhalb der Session</b>	ja	Gültige Statements:	SET GLOBALS
		Gültiges Kommando:	GLOBALS
<b>Programmierschnittstelle (API)</b>	USR0350N, USR1005N *	Siehe <i>SYSEXT Utility</i> in der Dokumentation <i>Debugger und Dienstprogramme</i> .  * Empfohlen.	



### Anmerkungen:

1. Das mit dem IA-Parameter angegebene Zeichen muss ein anderes sein als das mit dem Profil/Session-Parameter CF angegebene (Steuerzeichen für Terminalkommandos auf Großrechnern), DC (Dezimalzeichen) oder ID (Input-Begrenzungszeichen), und es sollte ein anderes sein als das mit dem HI-Profilparameter (Hilfezeichen) definierte.
2. In einer Natural-Session kann der Profilparameter IA durch den Session-Parameter IA überschrieben werden.
3. Unter Natural Security kann die Einstellung dieses Parameters durch die *Session-Parameter-Option* des Library-Profiles überschrieben werden.

**Beispiel:**

In dem folgenden Beispiel wird davon ausgegangen, dass zu Beginn 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



# 130 IC - Einfügungszeichen

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.

<b>Mögliche Werte</b>	beliebige Zeichen	Einzufügende Zeichenkette. Sie können eine Zeichenkette von eins bis zehn Zeichen eingeben.  <b>Anmerkung:</b> 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.
<b>Standard-Einstellung</b>	keine	
<b>Angabe innerhalb der Session</b>	ja	
<b>Gültige Statements</b>	FORMAT	Parameter kann dynamisch mit dem FORMAT-Statement angegeben werden.
	DISPLAY	Parameter kann auf Statement- und/oder Element-Ebene angegeben werden.
<b>Gültiges Kommando</b>	keines	



## Anmerkungen:

1. Das Einfügungszeichen wird zwischen den führenden Leerzeichen und dem Feldwert eingefügt, wohingegen das vorangestellte Zeichen vor dem führenden Leerzeichen ausgegeben wird.
2. Bei numerischen Werten werden die Einfügungszeichen vor der ersten signifikanten Stelle ausgegeben.
3. Die Parameter IC und LC schließen einander aus.

4. Der Parameter `IC` kann auch bei Feldern mit Format `U` verwendet werden.
5. Informationen zum Unicode-Format siehe *Unicode- und Codepage-Unterstützung in der Natural-Programmiersprache, Session-Parameter*, Abschnitt *EMU, ICU, LCU, TCU im Vergleich zu EM, IC, LC, TC* in der *Unicode- und Codepage-Unterstützung-Dokumentation*.
6. Der Unterschied zwischen den Session-Parametern `LC`, `LCU` und `IC`, `ICU` wird ersichtlich, wenn das entsprechende Feld rechtsbündig ausgegeben wird (Session-Parameter `AD=R`).
7. Siehe auch *Parameter zur Beeinflussung der Ausgabe von Feldern im Leitfaden zur Programmierung*.

### Beispiel:

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

# 131

## ICU - Unicode-Einfügungszeichen

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.

<b>Mögliche Werte</b>	beliebige Zeichen	Einzufügende Zeichenkette. Sie können eine Zeichenkette von eins bis zehn Zeichen eingeben.  <b>Anmerkung:</b> 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.
<b>Standard-Einstellung</b>	keine	
<b>Angabe innerhalb der Session</b>	ja	
<b>Gültige Statements</b>	<code>FORMAT</code>	Parameter kann dynamisch mit dem <code>FORMAT</code> -Statement angegeben werden.
	<code>DISPLAY</code>	Parameter kann auf Statement- und/oder Element-Ebene angegeben werden.
<b>Gültiges Kommando</b>	keines	



### Anmerkungen:

1. Der Session-Parameter `ICU` ist identisch mit dem Session-Parameter `IC`. Der Unterschied ist, dass das Einfügungszeichen immer im Unicode-Format gespeichert wird. Dadurch können Sie die Einfügungszeichen mit gemischten Zeichen aus unterschiedlichen Codepages eingeben. Es ist sichergestellt, dass immer das korrekte Zeichen angezeigt wird, unabhängig von der installierten System-Codepage.

2. Bei numerischen Werten werden die Einfügungszeichen vor der ersten signifikanten Stelle ausgegeben.
3. Die Parameter ICU und LCU schließen einander aus.

Siehe auch:

- *Parameter zur Beeinflussung der Ausgabe von Feldern im Leitfaden zur Programmierung*
- *Unicode- und Codepage-Unterstützung in der Natural-Programmiersprache, Abschnitt Session-Parameter, EMU, ICU, LCU, TCU im Vergleich zu EM, IC, LC, TC in der Unicode- und Codepage-Unterstützung-Dokumentation.*

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

<b>Mögliche Werte</b>	beliebiges Sonderzeichen oder Leerzeichen	Input-Begrenzungszeichen. <b>Anmerkung:</b> Wird ein Leerzeichen angegeben, dann wird kein Input-Begrenzungszeichen definiert. Diese Option ist nicht möglich auf UNIX- und Windows-Plattformen.	
<b>Standard-Einstellung</b>	,	Komma (,). <b>Anmerkung:</b> Falls das Begrenzungszeichen das Komma sein soll, muss dies als dynamischer Parameter in Apostrophen ID= ' , ' angegeben werden, da bei dynamischen Parametern das Komma als Begrenzungszeichen der Eingabeparameter gilt.	
<b>Dynamische Angabe</b>	ja		
<b>Angabe innerhalb der Session</b>	ja	Gültige Statements:	SET GLOBALS
		Gültiges Kommando:	GLOBALS
<b>Programmierschnittstelle (API)</b>	USR1005N	Siehe <i>SYSEXT Utility</i> in der Dokumentation <i>Debugger und Dienstprogramme</i> .	



#### Anmerkungen:

1. Das mit diesem Parameter angegebene Zeichen darf nicht dasselbe sein wie das mit dem Profil/Session-Parameter DC (Dezimalstellenzeichen) oder IA (INPUT-Zuweisungszeichen) angegebene Zeichen, und es sollte nicht dasselbe sein wie das mit dem CF-Parameter (Steuerzeichen für Großrechner-Terminalkommandos) oder HI-Parameter (Hilfe-Aufrufzeichen) angegebene Zeichen.
2. Der Punkt (.) sollte nicht als Input-Begrenzungszeichen benutzt werden, weil dies zu Situationen führen könnte, in denen ein Programmende-Punkt fälschlicherweise als Input-Begrenzungszeichen interpretiert wird. Ein Stern (\*) sollte auch nicht verwendet werden.

3. In einer Natural-Session kann der Profilparameter ID durch den Session-Parameter ID überschrieben werden.
4. Unter Natural Security kann die Einstellung dieses Parameters durch die Option *Session Parameters* im *Library Profile* überschrieben werden.

# 133

## IKEY - Verarbeitung von PA- und PF-Tasten

---

Dieser Natural-Profilparameter bestimmt die Aktion, die auszuführen ist, wenn eine Video-Terminal-Attention-Taste (PA-Taste) oder eine Programmfunktionstaste (PF-Taste) benutzt wird, um Daten einzugeben, und die Taste nicht mit dem Statement `SET KEY` in dem Natural-Programm definiert worden ist.

<b>Mögliche Werte</b>	ON	Der Wert <code>ENTR</code> wird in die Natural-Systemvariable <code>*PF-KEY</code> gesetzt; d.h., Natural reagiert so, als ob die <code>ENTER</code> -Taste gedrückt worden ist.
	OFF	Es wird eine <code>REINPUT</code> -Meldung erzeugt, die den Benutzer auffordert, eine gültige Taste zu drücken.
<b>Standard-Einstellung</b>	OFF	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	



# 134

## IM - Input-Modus

---

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

<b>Mögliche Werte</b>	F	Forms-Modus.	
	D	Delimiter-Modus.	
<b>Standard-Einstellung</b>	F		
<b>Dynamische Angabe</b>	ja		
<b>Angabe innerhalb der Session</b>	ja	Gültige Statements:	SET GLOBALS
		Gültiges Kommando:	GLOBALS
<b>Programmierschnittstelle (API)</b>	USR1005N	Siehe <i>SYSEXT Utility</i> in der Dokumentation <i>Debugger und Dienstprogramme</i> .	



### Anmerkungen:

1. In einer Natural-Session kann die Einstellung des Profilparameters IM durch den Session-Parameter IM überschrieben werden.
2. Die IM-Parametereinstellung kann auch mit den Natural Terminalkommandos %D und %F geändert werden.
3. Unter Natural Security kann die Einstellung dieses Parameters durch die Session Parameters-Option des Library-Profiles überschrieben werden.
4. Weitere Informationen zum Delimiter-Modus und Forms-Modus entnehmen Sie dem INPUT-Statement.



# 135

## IMSG - Nachricht bei Session-Initialisierungsfehler

---

Dieser Natural-Profilparameter dient dazu, den Bildschirm zu unterdrücken, der beim Auftreten eines Session-Initialisierungsfehlers erscheint. Dies kann nützlich sein, um unerwünschte Ausgaben, z.B. bei Drucker-Sessions, zu vermeiden.



**Vorsicht:** Benutzen Sie diesen Parameter mit Vorsicht, weil er die Fehlerdiagnose erschweren kann.

<b>Mögliche Werte</b>	ON	Der Bildschirm, der beim Auftreten eines Session-Initialisierungsfehlers erscheint, wird im Fehlerfall angezeigt.
	OFF	Der Bildschirm, der beim Auftreten eines Session-Initialisierungsfehlers erscheint, wird nicht angezeigt.
<b>Standard-Einstellung</b>	ON	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	



# 136

## IMSP - Allgemeine Parameter für das Natural IMS TM

### Interface

---

- NTIMSP-Makro-Syntax ..... 418
- Schlüsselwort-Subparameter ..... 418
- Beispiel für NTIMSP-Makro ..... 420

Dieser Natural-Profilparameter kann nur im Makro `NTIMSP` angegeben werden. Die dynamische Parameterangabe ist noch nicht möglich.

Das Makro `NTIMSP` dient zum Setzen von allgemeinen Parametern für das Natural IMS TM Interface.

Das Makro `NTIMSP` darf im Natural-Parametermodul nur einmal angegeben werden.

<b>Mögliche Werte</b>	Siehe unten.
<b>Standard-Einstellung</b>	Siehe unten.
<b>Dynamische Angabe</b>	nein
<b>Angabe innerhalb der Session</b>	nein

Dieses Dokument behandelt folgende Themen:

## NTIMSP-Makro-Syntax

---

Das Makro `NTIMSP` wird wie folgt angegeben:

```
NTIMSP NIINAME=value, *  
      SUBPOOL=value, *  
      THRELO=value, *  
      TRNCODE=value
```

Siehe [Schlüsselwort-Subparameter](#).

## Schlüsselwort-Subparameter

---

`NIINAME` | `SUBPOOL` | `THRELO` | `TRNCODE`

### NIINAME - Natural IMS TM Interface-Modulname

`NIINAME=value` dient zur Angabe des Namens des Natural IMS TM Interface-Moduls, welches vom aktuellen Treiber benutzt werden soll.

Wert	Erklärung
1- 8 Zeichen	Ein gültiger Modulname.
<code>NIINTFM</code>	Dies ist der Standardwert.

## SUBPOOL - Subpool für Speicheranforderungen

SUBPOOL=*value* gibt den z/OS-Subpool an, der für Speicheranforderungen benutzt werden soll.

Wert	Erklärung
0 - 127	Die Nummer eines z/OS-Task-bezogenen Subpool, der von einem Problemstatusprogramm benutzt wird.
125	Dies ist der Standardwert.

## THRELO - Thread-Verschiebung

THRELO=*value* bestimmt, ob ein Natural-Speicher-Thread nach einer Terminal-Ein-/Ausgabe einer anderen virtuellen Adresse zugewiesen werden kann.

Wert	Erklärung
ON	Thread-Verschiebung ist möglich: Der Natural-Thread kann nach einer Terminal-Ein-/Ausgabe einer anderen virtuellen Adresse zugewiesen werden.  Dies ist der Standardwert.
OFF	Thread-Verschiebung ist nicht möglich: Der Natural-Thread verbleibt an derselben virtuellen Adresse.  Die Größe des Natural-Thread wird von der ersten Natural Session bestimmt, die den Thread zuweist. Dies hat folgende Auswirkungen: <ul style="list-style-type: none"> <li>■ Der Schlüsselwort-Subparameter THSIZE (NTIMSPE-Makro) wird ignoriert.  Eine Ausnahme bildet die erste Natural-Session in einer MPP-Umgebung.</li> <li>■ Der Profilparameter RELO wird ignoriert.</li> <li>■ Der Speicher für den Natural-Thread bleibt solange zugewiesen, bis die MPP-Umgebung gestoppt wird.</li> </ul> <p><b>Anmerkung:</b> Wir empfehlen Ihnen, THRELO=OFF nur zu setzen, falls Sie Con-form verwenden.</p>

## TRNCODE - Transaktionsschlüssel identifizieren

TRNCODE=*value* kann benutzt werden, um den Transaktionsschlüssel zu identifizieren.

Wert	Erklärung
ON	Der Schlüsselwort-Subparameter TRNCODE wird aktiviert, um den TRNCODE-Parameter auszuwerten:  auf der ersten Karte der BPM-CONTROL-Datei, im Makro NIMBOOT und als Startup-Parameter für die Server-Umgebung.

Wert	Erklärung
OFF	Der TRNCODE-Parameter wird deaktiviert.  Dies ist der Standardwert.

## Beispiel für NTIMSP-Makro

---

```
NTIMSP NIINAME=NATIMS1,SUBPOOL=0
```

# 137 IMSPE - Umgebungsparameter für das Natural IMS TM

## Interface

---

■ NTIMSPE-Makro-Syntax .....	422
■ Schlüsselwort-Subparameter .....	423
■ Beispiel für NTIMSPE-Makro .....	434

Dieser Natural-Profilparameter kann nur im Makro `NTIMSPE` angegeben werden. Die dynamische Parameterangabe ist noch nicht möglich.

Das Makro `NTIMSPE` dient zur Definition von umgebungsspezifischen Parametergruppen, die zu Transaktionsdefinitionen zugewiesen werden können (Schlüsselwort-Subparameter `ENVPID` im Makro `NTIMSPT`).



**Anmerkung:** Sie können im Natural-Parametermodul nur dann ein `NTIMSPE`-Makro definieren, wenn das Modul bereits mindestens ein `NTIMSPT`-Makro enthält. Andernfalls tritt ein Assemblierungsfehler auf, wenn Sie versuchen, das Modul zu assemblieren.

<b>Mögliche Werte</b>	Siehe unten.
<b>Standard-Einstellung</b>	Siehe unten.
<b>Dynamische Angabe</b>	nein
<b>Angabe innerhalb der Session</b>	nein

Dieses Dokument behandelt folgende Themen:

## NTIMSPE-Makro-Syntax

---

Das Makro `NTIMSPE` wird wie folgt angegeben:

```

NTIMSPE ACTACTV=value, *
        ACTAHDR=value, *
        ACTARID=value, *
        ACTLOG=value, *
        BMPABER=value, *
        BROACTV=value, *
        CMBSIZE=value, *
        COLPSCR=value, *
        ENDMODN=value, *
        ENTRYNM=value, *
        ERRLHDR=value, *
        HCBSIZE=value, *
        HDENS DU=value, *
        LINPSCR=value, *
        MISIZE=value, *
        MONACTV=value, *
        MOSIZE=value, *
        NSBNAME=value, *
        PRTDRIV=value, *
        ROLLSRV=value, *
        ROLLFN=value, *
        SPASIZE=value, *
        SPATID=value, *
        SUPNONC=value, *
    
```

TERMDB= <i>value</i> ,	*
TERMIPL= <i>value</i> ,	*
THBELOW= <i>value</i> ,	*
THSIZE= <i>value</i> ,	*
USERID= <i>value</i>	

See [Keyword Subparameters](#).

## Schlüsselwort-Subparameter

[ACTACTV](#) | [ACTAHDR](#) | [ACTARID](#) | [ACTLOG](#) | [BMPABER](#) | [BROACTV](#) | [CMBSIZE](#) | [COLPSCR](#) | [ENDMODN](#) | [ENTRYNM](#) | [ERRLHDR](#) | [HCBSIZE](#) | [HDENSU](#) | [LINPSCR](#) | [MISIZE](#) | [MONACTV](#) | [MOSIZE](#) | [NSBNAME](#) | [PRTDRIV](#) | [ROLLSRV](#) | [ROLLFN](#) | [SPASIZE](#) | [SPATID](#) | [SUPNONC](#) | [TERMDB](#) | [TERMIPL](#) | [THBELOW](#) | [THSIZE](#) | [USERID](#)

### ACTACTV - Abrechnung aktivieren

Dieser Schlüsselwort-Subparameter gilt nur in dialog-orientierten Umgebungen.

ACTACTV=*value* gibt an, ob die Abrechnungsfunktion („Accounting“) aktiviert ist.

Wert	Erklärung
ON	Bei jeder Terminal-Ein-/Ausgabe wird ein Abrechnungsdatensatz geschrieben.
OFF	Es wird kein Abrechnungsdatensatz geschrieben. Dies ist der Standardwert.

### ACTAHDR - Vorsatz für Abrechnungsdatensätze

Dieser Schlüsselwort-Subparameter gilt nur für die Abrechnungsfunktion.

ACTAHDR=*value* definiert den Vorsatz für die Abrechnungsdatensätze, falls diese in die IMS-Protokolldatei („Log File“) geschrieben werden. Dieser Schlüsselwort-Subparameter wird nur ausgewertet, wenn der Schlüsselwort-Subparameter [ACTLOG](#) auf CMD gesetzt ist.

Wert	Erklärung
1 - 8 Zeichen	Ein Vorsatz-Name.
SAG\$\$\$\$	Dies ist der Standardwert.

## ACTARID - Abrechnungsdatensatzkennung

Dieser Schlüsselwort-Subparameter gilt nur für die Abrechnungsfunktion (Accounting).

ACTARID=*value* gibt die Abrechnungsdatensatzkennung an, wenn der Abrechnungsdatensatz unter Verwendung der Einstellungen LOG oder SMF des Schlüsselwort-Subparameters ACTLOG geschrieben wird.

Wert	Erklärung
A0 - FF	Der Protokollschlüssel, wenn ACTLOG auf LOG gesetzt ist.
128 - 255	Die Nummer des SMF-Datensatztyps, wenn ACTLOG auf SMF gesetzt ist.



**Anmerkung:** Es gibt keinen Standardwert.

## ACTLOG - Abrechnungsdatensätze in Protokolldatei schreiben

Dieser Schlüsselwort-Subparameter gilt nur für die Abrechnungsfunktion (Accounting).

ACTLOG=*value* gibt an, wohin die Abrechnungsdatensätze geschrieben werden.

Wert	Erklärung
CMD	Die Abrechnungsdatensätze werden unter Verwendung des CMD-Aufrufs in die IMS-Protokolldatei geschrieben.  Dies ist der Standardwert.
LOG	Die Abrechnungsdatensätze werden unter Verwendung des LOG-Aufrufs in die IMS-Protokolldatei geschrieben.
SMF	Die Abrechnungsdatensätze werden nach SMF geschrieben, und zwar unter Verwendung des <i>Authorized Services Manager</i> (siehe <i>TP Monitor Interfaces</i> -Dokumentation).

## BMPABER - Fehler für Beendigung eines BMP-Laufs

BMPABER=*value* gibt an, wie ein BMP-Lauf beendet werden soll, falls entweder ein Natural-Laufzeitfehler oder ein nicht korrigierbarer Natural IMS TM Interface-Fehler auftritt.

Wert	Erklärung
ON	Der BMP-Lauf wird mit dem Benutzer-Abbruchschlüssel U3521 beendet.
OFF	Der BMP-Lauf wird normal beendet, wobei der Natural-Beendigungsfehler als Bedingungsschlüssel verwendet wird. Falls der BMP-Lauf mit einem nicht rekonstruierbaren Natural IMS TM Interface-Fehler beendet wird, wird der Bedingungsschlüssel 1024 gesetzt.  Dies ist der Standardwert.

## BROACTV - Rundspruch-Nachrichtenfunktion aktivieren/deaktivieren

Dieser Schlüsselwort-Subparameter gilt nur in dialog-orientierten Umgebungen.

BROACTV=*value* aktiviert bzw. deaktiviert die allgemeine Rundspruch-Nachrichtenfunktion ("Broadcasting").

Wert	Erklärung
ON	Die Rundspruch-Nachrichtenfunktion ist aktiviert.
OFF	Die Rundspruch-Nachrichtenfunktion ist deaktiviert. Dies ist der Standardwert.

## CMBSIZE - Befehlspuffergröße

CMBSIZE=*value* gibt die Größe des Befehlspuffers an.

Der Befehlspuffer wird benutzt von den Anwendungsprogrammierschnittstellen (APIs) NIICMD und NIIGCMD, dem Service-Modul CMCMMND und der Abrechnungsfunktion. Weitere Informationen siehe *TP Monitor Interfaces*-Dokumentation.

Der Befehlspuffer muss groß genug sein, um die Maximallänge der zu verarbeitenden IMS-Befehle und die Maximallänge des Abrechnungsdatensatzes einschließlich der Benutzer-Erweiterung aufnehmen zu können.

Wert	Erklärung
100 - 16777216	Puffergröße in KB.
1024	Dies ist der Standardwert.

## COLPSCR - Anzahl der Bildschirmspalten

COLPSCR=*value* gibt die Anzahl der Spalten pro Bildschirm an.

Wert	Erklärung
35 - 250	Bildschirmbreite.
80	Dies ist der Standardwert.

## ENDMODN - Format des Beendigungsbildschirms

ENDMODN=*value* gibt den MOD name an, der zum Formatieren des Bildschirms benutzt wird, welcher nach erfolgreicher Beendigung einer Natural-Session erscheint.

ENDMODN ermöglicht es, Natural in ein kundenspezifisches Menü aufzunehmen.

Der Wert von ENDMODN kann durch die Anwendungsprogrammierschnittstelle (API) NIIEMOD und das Service-Modul CMEMOD überschrieben werden. Weitere Informationen siehe *TP Monitor Interfaces*-Dokumentation.

Wenn Natural mit einem Fehler beendet wird, wird standardmäßig der Wert DFSM02 zur Ausgabe der entsprechenden Natural-Fehlermeldung benutzt.

Wert	Erklärung
1 - 8 Zeichen	Ein gültiger MOD name.
DFSM02	Dies ist der Standardwert.

## ENTRYNM - Aktuellen umgebungsspezifischen Parametersatz ermitteln

ENTRYNM=*value* ermittelt den aktuellen umgebungsspezifischen Parametersatz.

Wert	Erklärung
1 - 8 Zeichen	Der Name des zurzeit benutzten umgebungsspezifischen Parametersatzes.
ENV00000	Dies ist der Standardwert.

## ERRLHDR - Vorsatz für IMS TM-Fehlerprotokolle

ERRLHDR=*value* gibt den Vorsatz für IMS TM-Protokolldatensätze an, die bei Auftreten von Fehlern im Natural IMS TM Interface geschrieben werden.

Wenn Sie nicht wollen, dass im Falle eines nicht korrigierbaren Natural IMS TM Interface-Fehlers eine Meldung in das IMS TM-Protokoll geschrieben wird, setzen Sie den Subparameter ERRLHDR auf Null, d.h., geben Sie ERRLHDR=, an.

Weitere Informationen siehe *Recovery Handling* in der *TP Monitor Interfaces*-Dokumentation.

Wert	Erklärung
1 - 8 Zeichen	Name eines Vorsatzes.
NIIERR\$\$	Dies ist der Standardwert.

**HCBSIZE - Größe des Hardcopy Print-Puffers**

HCBSIZE=*value* gibt die Größe des Puffers für gedruckte Maschinenausgaben an.

Datensätze, die unter Benutzung der Natural Hardcopy-Funktion an ein Druckerziel gesendet werden, werden zwischengespeichert.

Wert	Erklärung
0 - 16777216	Puffergröße in KB.
1024	Dies ist der Standardwert.

**HDENSU - High-Density Dump**

HDENSU=*value* gibt an, ob ein durch einen Natural IMS TM Interface-Fehler veranlasster Hauptspeicherteilauszug („Snap Dump“) als hoch verdichteter Hauptspeicherauszug auf ein Druckersystem 3800 geschrieben werden soll.

Wert	Erklärung
ON	Es wird ein hoch verdichteter Hauptspeicherauszug geschrieben.
OFF	Es wird kein hoch verdichteter Hauptspeicherauszug geschrieben. Dies ist der Standardwert.

**LINPSCR - Anzahl der Bildschirmzeilen**

LINPSCR=*value* definiert die Anzahl der Zeilen pro Bildschirm.

Wert	Erklärung
1 - 250	Bildschirmgröße.
24	Dies ist der Standardwert.

**MISIZE - Puffergröße für Eingabenachricht**

MISIZE=*value* gibt die Größe des Puffers an, der die Eingabenachricht aufnehmen soll.

Dieser Bereich muss so groß sein wie die längste von IMS TM zu empfangende Eingabenachricht.

Wert	Erklärung
100 - 16777216	Puffergröße in KB.
4096	Dies ist der Standardwert.

### MONACTV - Monitoring-Funktion aktivieren/deaktivieren

Dieser Schlüsselwort-Subparameter gilt nur in dialog-orientierten Umgebungen.

MONACTV=*value* aktiviert bzw. deaktiviert die Überwachungsfunktion. Bei aktivierter Funktion wird der Status der Session bei jeder Terminal-Ein-/Ausgabe auf den SIP Server geschrieben. Bei deaktivierter Funktion wird kein Session-Status unterstützt.

Wert	Erklärung
ON	Die Überwachungsfunktion ist aktiviert.
OFF	Die Überwachungsfunktion ist deaktiviert. Dies ist der Standardwert.

### MOSIZE - Puffergröße für Ausgabenachricht

MOSIZE=*value* gibt die Größe des Puffers an, der die Ausgabenachricht aufnehmen soll.

Dieser Bereich muss so groß sein wie die längste an IMS TM zu sendende Ausgabenachricht.

Wert	Erklärung
2048 - 16777216	Puffergröße in KB.
4096	Dies ist der Standardwert.

### NSBNAME - NSB-Namen des Natural DL/I Interface verwenden

NSBNAME=*value* setzt den Namen der NSB auf die vom Natural DL/I Interface benutzte NSB.

Wert	Erklärung
ON	Setzt den NSB-Namen auf den PSB-Namen, der mit dem Schlüsselwort-Subparameter <b>PSB</b> des Makros NTIMSPT für den verwendeten Transaktionsschlüssel definiert wurde.
OFF	Setzt den NSB-Namen auf den PSB-Namen, der von IMS TM verwendet wird. Dies ist der Standardwert.

## PRTDRIV - Treiber für IMS TM-Drucker

PRTDRIV=*value* gibt den Druckertreiber an, der für Berichte benutzt werden soll, die direkt auf einen IMS TM-Drucker geschrieben werden.

Weitere Informationen siehe *Support of the Natural WRITE (n) Statement* in der *TP Monitor Interfaces-Dokumentation*.

Wert	Erklärung
<i>driver-name</i>	Name eines Treibers, der in einer der folgenden Tabellen aufgeführt ist:  <i>Treiber für SCS-Drucker</i> <i>Treiber für SCS-Drucker</i> <i>Treiber für JES API</i>
SCS_S2	Dies ist der Standardwert.

### Treiber für SCS-Drucker

Treiber	Verwendungszweck
SCS_B1	Formularvorschub am Anfang und Ende des Berichts, Seitenanfang in Zeile 1.
SCS_B2	Formularvorschub am Anfang und Ende des Berichts, Seitenanfang in Zeile 2.
SCS_E1	Formularvorschub am Ende des Berichts, Seitenanfang in Zeile 1.
SCS_E2	Formularvorschub am Ende des Berichts, Seitenanfang in Zeile 2.
SCS_N1	Kein Formularvorschub am Anfang oder Ende des Berichts, Seitenanfang in Zeile 1.
SCS_N2	Kein Formularvorschub am Anfang oder Ende des Berichts, Seitenanfang in Zeile 2.
SCS_S1	Formularvorschub am Anfang des Berichts, Seitenanfang in Zeile 1.
SCS_S2	Formularvorschub am Anfang des Berichts, Seitenanfang in Zeile 2.

### Treiber für Nicht-SCS-Drucker

Treiber	Verwendungszweck
NSCS_B1	Formularvorschub am Anfang und Ende des Berichts, Seitenanfang in Zeile 1.
NSCS_B2	Formularvorschub am Anfang und Ende des Berichts, Seitenanfang in Zeile 2.
NSCS_E1	Formularvorschub am Ende des Berichts, Seitenanfang in Zeile 1.
NSCS_E2	Formularvorschub am Ende des Berichts, Seitenanfang in Zeile 2.
NSCS_N1	Kein Formularvorschub am Anfang oder Ende des Berichts, Seitenanfang in Zeile 1.
NSCS_N2	Kein Formularvorschub am Anfang oder Ende des Berichts, Seitenanfang in Zeile 2.
NSCS_S1	Formularvorschub am Anfang des Berichts, Seitenanfang in Zeile 1.
NSCS_S2	Formularvorschub am Anfang des Berichts, Seitenanfang in Zeile 2.

Treiber für JES API

Treiber	Verwendungszweck																								
JES	<p>Die folgenden Optionen für die Verarbeitung von Datasets für JES werden von den entsprechenden <b>NTPRINT</b>-Makro-Parametern oder den Optionen des Statements <b>DEFINE PRINTER</b> (siehe <i>Statements</i>-Dokumentation) übernommen:</p> <table border="1" data-bbox="280 436 1377 793"> <tr> <td>JES</td> <td>NTPRINT</td> <td>DEFINE PRINTER</td> </tr> <tr> <td>CLASS</td> <td>CLASS</td> <td>CLASS</td> </tr> <tr> <td>COPIES</td> <td>COPIES</td> <td>COPIES</td> </tr> <tr> <td>DEST</td> <td>DEST</td> <td>OUTPUT</td> </tr> <tr> <td>FORMS</td> <td>FORMS</td> <td>FORMS</td> </tr> <tr> <td>NAME</td> <td>NAME</td> <td>NAME</td> </tr> <tr> <td>OUTDISP</td> <td>DISP</td> <td>DISP</td> </tr> <tr> <td>PRTY</td> <td>PRTY</td> <td>PRTY</td> </tr> </table> <p>Die generierte JES API-Parameter-Zeichenkette lautet:</p> <pre data-bbox="280 877 1377 1010">IAFP=AOM, PRT0=..OUTDI(<i>disp</i>),DES(<i>dest</i>), CLA(<i>class</i>,COP(<i>copies</i>), FORMS(<i>forms</i>),NAME(<i>name</i>), PRTY(<i>prty</i>)</pre> <p><b>Anmerkung:</b> Nicht angegebene NTPRINT-Makro-Parameter bzw. DEFINE PRINTER-Statement-Optionen werden ignoriert.</p>	JES	NTPRINT	DEFINE PRINTER	CLASS	CLASS	CLASS	COPIES	COPIES	COPIES	DEST	DEST	OUTPUT	FORMS	FORMS	FORMS	NAME	NAME	NAME	OUTDISP	DISP	DISP	PRTY	PRTY	PRTY
JES	NTPRINT	DEFINE PRINTER																							
CLASS	CLASS	CLASS																							
COPIES	COPIES	COPIES																							
DEST	DEST	OUTPUT																							
FORMS	FORMS	FORMS																							
NAME	NAME	NAME																							
OUTDISP	DISP	DISP																							
PRTY	PRTY	PRTY																							
JES xxxxx	<p>Die Optionen für die Verarbeitung von Datasets für JES werden vom <b>OUTPUT JCL</b>-Statement mit dem Namen <b>JESxxxxx</b> übernommen. Dabei kann <b>xxxxx</b> bis zu 5 Zeichen lang sein.</p> <p>Die generierte JES API-Parameter-Zeichenkette lautet:</p> <pre data-bbox="280 1293 1377 1325">IAFP=AOM,OUTN=JESxxxxx</pre> <p>Das <b>OUTPUT JCL</b>-Statement folgendermaßen aussehen:</p> <pre data-bbox="280 1430 1377 1493">JESxxxxx OUTPUT OUTDISP=WRITE,DEST=<i>dest</i>, CLASS=A,COPIES=1,FORMS=<i>form</i>,...</pre> <p><b>Anmerkung:</b> Falls im Jobstrom kein <b>OUTPUT JCL</b>-Statement existiert, wird ein Fehler gemeldet.</p>																								

## ROLLSRV - Natural-Thread-Speicherung

Dieser Schlüsselwort-Subparameter gilt nur in dialog-orientierten Umgebungen.

ROLLSRV=*value* gibt das Speichermedium an, auf dem ein Natural-Thread zwischen Terminal-Ein- und Ausgabe aus- bzw. eingespeichert werden soll.

Siehe auch *Roll File and Roll Server* in der *TP Monitor Interfaces*-Dokumentation.

Wert	Erklärung
ON	Es wird der Natural Roll Server benutzt. Dies ist der Standardwert.
OFF	Es werden Dateien („Roll Files“) zum Aus- und Einspeichern benutzt, siehe <a href="#">ROLLFN</a> .

## ROLLFN - Anzahl der Roll Files

Dieser Schlüsselwort-Subparameter gilt nur in dialog-orientierten Umgebungen.

ROLLFN=*value* gibt die Anzahl der Dateien („Roll Files“) an, die zum Aus- und Einspeichern eines Natural-Thread benutzt werden sollen, wenn der Subparameter [ROLLSRV](#) auf OFF gesetzt ist.

Wert	Erklärung
1 - 5	Anzahl der zu benutzenden Dateien.
1	Dies ist der Standardwert.

## SPASIZE - Puffergröße für Scratch-Pad

Dieser Schlüsselwort-Subparameter gilt nur in dialog-orientierten Umgebungen.

SPASIZE=*value* gibt die Größe des Puffers an, der den Scratch-Pad-Bereich („Scratch Pad Area“, SPA) enthalten soll.

In einer nicht-konversationellen Umgebung ist dies ebenfalls die Größe des simulierten SPA, der auf den SIP Server geschrieben wird.

Wert	Erklärung
256 - 16777216	Puffergröße in KB.
1024	Dies ist der Standardwert.

### SPATID - Subsystem ID für ASM

SPATID=*value* gibt die Natural-Subsystemkennung für den Authorized Services Manager (ASM) an, der benutzt wird, um den Scratch-Pad-Bereich („Scratch Pad Area“, SPA) für einen nicht-konversationellen Treiber zu speichern. Der ASM wird in der *Operations*-Dokumentation beschrieben.

Wert	Erklärung
1 - 4 Zeichen	Subsystemkennung.  Dieser Wert muss bei allen Parametertabellen gleich sein und muss mit dem Wert übereinstimmen, der für SPATID im Makro NIMPIXT angegeben ist (siehe <i>TP Monitor Interfaces</i> -Dokumentation).
NAT2	Dies ist der Standardwert.

### SUPNONC - Umgebungsumschaltung ermöglichen

Dieser Schlüsselwort-Subparameter gilt nur in dialog-orientierten konversationellen Umgebungen.

SUPNONC=*value* gibt an, ob das Umschalten von einer terminalorientierten auf eine konversationelle Umgebung möglich ist oder nicht.

Wert	Erklärung
ON	Das Umschalten ist gestattet.
OFF	Das Umschalten ist nicht gestattet.  Dies ist der Standardwert.

### TERMDB - Session-Beendigung bei fehlender DL/I-Datenbank

Dieser Schlüsselwort-Subparameter gilt nur in dialog-orientierten Umgebungen.

TERMDB=*value* gibt an, ob die Natural-Session beendet werden soll, wenn eine der im Program Specification Block (PSB) angegebenen DL/I-Datenbanken nicht verfügbar ist.

Wert	Erklärung
ON	Die Natural-Session wird beendet.
OFF	Die Natural-Session wird nicht beendet.  Falls eine der Datenbanken beim Zugriff nicht verfügbar ist, wird der Natural-Transaktionsschlüssel durch IMS TM suspendiert.  Dies ist der Standardwert.

**TERMIPL - Session-Beendigung bei IPL**

Dieser Schlüsselwort-Subparameter gilt nur in dialog-orientierten Umgebungen.

TERMIPL=*value* gibt an, ob die Natural-Session mit einer Fehlermeldung beendet wird, wenn ein Initial Program Load (IPL) zwischen dem aktuellen Transaktionsschritt und dem Beginn der Session aufgetreten ist.

Wert	Erklärung
ON	Die Natural-Session wird beendet.
OFF	Die Natural-Session wird nicht beendet. Dies ist der Standardwert.

**THBELOW - Natural-Thread-Zuweisung**

THBELOW=*value* gibt an, ob der Natural-Thread unterhalb oder oberhalb der 16 MB-Grenze zugewiesen wird.

Wert	Erklärung
ON	Der Natural-Thread wird unterhalb der 16 MB-Grenze zugewiesen. Dies ist der Standardwert.
OFF	Der Natural-Thread wird oberhalb der 16 MB-Grenze zugewiesen.

**THSIZE - Natural-Thread-Größe**

THSIZE=*value* gibt die Größe des Natural-Thread an. Dies ist der Bereich, welcher alle Natural-Puffer enthält, die einen Bezug zur Benutzersitzungen haben.

Wert	Erklärung
100000 - 99999999	Eine Thread-Größe, die ein Vielfaches von 8 und größer als oder gleich 100000 ist.
1048576	Dies ist der Standardwert.

## USERID - Wert von \*INIT-USER festlegen

Dieser Schlüsselwort-Subparameter gilt nur für den BMP-Treiber.

USERID=*value* gibt an, wie der Wert der Systemvariablen \*INIT-USER (siehe *Systemvariablen-Dokumentation*) bestimmt wird.

Wert	Erklärung
ON	Die Natural-Benutzerkennung, die in der Systemvariablen *INIT-USER enthalten ist, wird entweder vom Security-Zugangskontrollblock (falls ein Security-Software-Paket aktiv ist) oder vom Parameter USER der Jobkarte übernommen.
OFF	Die Natural-Benutzerkennung, die in der Systemvariablen *INIT-USER enthalten ist, wird vom Jobnamen übernommen.  Dies ist der Standardwert.

## Beispiel für NTIMSPE-Makro

---

```

NTIMSPE ENTRYNM=ENV0021,          *
      ACTACTV=ON,                  *
      ACTLOG=LOG,                  *
      ACTARID=A1,                  *
      PRTDRIV=SCS_E1,             *
      THBELOW=OFF,                *
      THSIZE=1000000,             *
      SPATID=ASM1
    
```

# 138

## IMSPT - Transaktionsdefinitionen für Natural IMS TM

### Interface

---

▪ NTIMSPT-Makro-Syntax .....	436
▪ Schlüsselwort-Subparameter .....	436
▪ Beispiel für NTIMSPT-Makro .....	440

Dieser Natural-Profilparameter kann nur mit dem Makro `NTIMSPT` angegeben werden. Die dynamische Parameterangabe ist noch nicht möglich.

Das Makro `NTIMSPT` wird benötigt, um den Transaktionschlüssel für jede Natural-Transaktion sowie die individuellen Transaktionsparameter zu definieren. Die Schlüsselwort-Subparameter `TRAN` und `PSB` müssen explizit angegeben werden, weil es für sie keine Standardwerte gibt.

<b>Mögliche Werte</b>	Siehe unten.
<b>Standard-Einstellung</b>	Siehe unten.
<b>Dynamische Angabe</b>	nein
<b>Angabe innerhalb der Session</b>	nein

Dieses Dokument behandelt folgende Themen:

## NTIMSPT-Makro-Syntax

---

Das Makro `NTIMSPT` wird wie folgt angegeben:

```

NTIMSPT ALTPCB=value, *
        ENVPID=value, *
        HCPCB=value, *
        MSGPCB=value, *
        NRAST=value, *
        PCBS=value, *
        PSB=value, *
        TRAN=value, *
        TYPE=value, *
        WRKPCBS=value
    
```

See [Schlüsselwort-Subparameter](#).

## Schlüsselwort-Subparameter

---

[ALTPCB](#) | [ENVPID](#) | [HCPCB](#) | [MSGPCB](#) | [NRAST](#) | [PCBS](#) | [PSB](#) | [TRAN](#) | [TYPE](#) | [WRKPCBS](#)

### ALTPCB - Alternativer TP PCB

ALTPCB=*value* bestimmt den alternativen TP Program Communication Block, welcher für die Service-Module CMQUEUE, CMQUEUEX, NIIDQUMS und NIIDPURG verwendet werden soll (siehe *TP Monitor Interfaces*-Dokumentation).

Die hier angegebene Nummer kann durch die Service-Module überschrieben werden.

Wert	Erklärung
1 - 255	Die Nummer des zu verwendenden alternativen TP PCB.
1	Dies ist der Standardwert.

### ENVPID - Umgebungsparametersatz angeben

ENVPID=*value* gibt den in der Transaktionschlüsseltabelle zu verwendende Umgebungsparametersatz an.

Wert	Erklärung
1 - 8 Zeichen	Der Name des Umgebungsparametersatzes entsprechend der Definition im Schlüsselwort-Subparameter <a href="#">ENTRYNM</a> im Makro NTIMSPE.
ENV00000	Dies ist der Standardwert.

### HCPCB - PCB für Hardcopy

HCPCB=*value* gibt die PCB-Nummer an, die für die Hardcopy-Print-Funktion verwendet werden soll.

Wert	Erklärung
SYSPCB	Es wird der erste alternative P PCB verwendet. Dies ist der Standardwert.
WRKPCB	Einer der zusätzlichen alternativen TP PCBs wird verwendet. Dies ermöglicht es Ihnen, einen ausdrücklich dafür vorgesehenen TP PCB für die Hardcopy-Print-Funktion zu benutzen.

## MSGPCB - Nachrichten-Druck-PCB

Dieser Schlüsselwort-Subparameter gilt nur für nachrichten-orientierte Umgebungen und den Server-Treiber.

MSGPCB=*value* gibt den PCB an, der zum Drucken von Fehlermeldungen und Standardausgaben in einer nachrichten-orientierte Umgebung und für den Server-Treiber verwendet werden soll.

Wert	Erklärung
SYSPCB	Es wird der erste alternative TP PCB verwendet. Dies ist der Standardwert.
OWNPCB	Der zweite alternative TP PCB wird reserviert und benutzt. Dies ermöglicht es Ihnen, einen ausdrücklich dafür vorgesehenen TP PCB zum Versenden von Nachrichten zu benutzen. is reserved and used. This enables you to use an express TP PCB for sending messages.

## NRAST - Natural-NRA-Versatz innerhalb des Scratch Pad-Bereichs

NRAST=*value* definiert den Versatz des für Natural reservierten Bereichs (Natural Reserved Area, NRA) innerhalb des IMS TM Notizblock-Speicherbereichs (Scratch Pad Area, SPA) an.

Die aktuelle Länge des NRA beträgt 157 Bytes. Die Länge des NRA kann sich bei der nächsten Version des Natural IMS TM Interface ändern.



**Anmerkung:** Falls Sie eigene Informationen im SPA speichern wollen, um sie an eine Nicht-Natural-Transaktion zu übergeben, empfehlen wir Ihnen, Ihre Daten vor dem für Natural reservierten Bereich zu speichern, um versionskompatibel zu sein.

Wert	Erklärung
16 - 32600	Der Versatz des NRA innerhalb des SPA.
16	Dies ist der Standardwert.

## PCBS - PCB-Namenszuweisung

PCBS=(*pcb-1, num-1, pcb-2, num-2, ...*) dient zum Zuweisen eines logischen Namens zu einem PCB.

Wert	Erklärung
<i>pcb-n</i> (1 - 8 Zeichen)	<i>pcb-n</i> gibt den logischen Namen des PCB an.
1-255	Dieser Parameter ist optional. <i>num-n</i> gibt die Positionsnummer des PCB im PSB an. Falls nichts angegeben wird, wird die Folgenummer des PCB in der Parameterliste benutzt.

 **Anmerkung:** Es gibt keinen Standardwert.

### PSB - PSB-Name der IMS TM-Transaktion

PSB=*value* definiert den Namen des PSB, der von der IMS TM-Transaktion benutzt wird. Der PSB-Name dient dazu, den Eintrag in der IMS TM-Transaktionsschlüsseltabelle für die nicht-nachrichtengesteuerte Batch-Nachrichtenverarbeitung und für Batch-Umgebungen zu kennzeichnen.

Wert	Erklärung
1 - 8 Zeichen	Der PSB-Name, der dem aktuellen Transaktionsschlüssel entspricht. Der Name muss die Namenskonventionen erfüllen, die IMS TM im Makro APPLCTN verwendet.  Es gibt keinen Standardwert.

### TRAN - IMS TM-Transaktionsschlüssel

TRAN=*value* gibt den IMS TM-Transaktionsschlüssel an.

Dieser Transaktionsschlüssel wird in nicht-nachrichten-gesteuerten BMP-Umgebungen und in Batch-Umgebungen ignoriert.

Wert	Erklärung
1 - 8 Zeichen	Ein Transaktionsschlüssel. Der Name muss die Namenskonventionen erfüllen, die IMS TM im Makro TRANSACT verwendet.  Es gibt keinen Standardwert.

### TYPE - Natural-Transaktionstyp

TYPE=*value* definiert den Natural-Transaktionstyp.

Wert	Erklärung
CONV	Konversationale Natural-Session.  Dies ist der Standardwert.
NONC	Nicht-konversationelle Natural-Session.
SFE	Natural Development Server- und/oder Natural Web I/O Interface Server-Session.

## WRKPCBS - Anzahl der alternativen PCBs zum Drucken

WRKPCBS=*value* gibt zusätzlich zum ersten TP PCB und ggf. zum Subparameter MSGPCB die Anzahl der zum Drucken verwendeten alternativen PCBs an.

Wert	Erklärung
0	Es steht kein IMS TM-Drucker zur Verfügung.  Dies ist der Standardwert.
1 - 32	Die Anzahl der zum Drucken verwendeten alternativen TP PCBs.  Beispiel 1:  MSGPCB=SYSPCB WRKPCBS=2  Der PSB muss 3 alternative TP PCBs enthalten.  Beispiel 2:  MSGPCB=OWNPCB WRKPCBS=2  Der PSB muss 3 alternative TP PCBs enthalten. Der zweite alternative TP PCB ist für Fehlermeldungen und die Standardausgabe der nachrichten-orientierten Umgebung reserviert.

## Beispiel für NTIMSPT-Makro

---

```
NTIMSPT TRAN=MYNAT,PSB=MYPST
```

# 139

## INTENS - Drucken von hervorgehobenen Feldern

---

Dieser Natural-Profilparameter gibt an, wie oft ein hervorgehobenes Feld oder ein unterstrichenes Zeichen überdruckt werden soll, wenn es auf einem Druck-Ausgabegerät gedruckt wird.

<b>Mögliche Werte</b>	1 - 10	Anzahl, wie oft ein hervorgehobenes Feld oder ein unterstrichenes Zeichen überdruckt wird.  <b>Anmerkung:</b> Ein unterstrichenes Zeichen wird nur dann gedruckt, wenn der Parameter auf einen Wert > 1 gesetzt ist. Wenn INTENS=1, werden unterstrichene Felder ohne Unterstreichung gedruckt.
<b>Standard-Einstellung</b>	3	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	



# 140

## IP - Eingabeaufforderungstext

---

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

<b>Mögliche Werte</b>	ON	Eingabe-/Ausgabefeldern eines INPUT-Statements, denen kein Textelement 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.	
<b>Standard-Einstellung</b>	ON		
<b>Angabe innerhalb der Session</b>	ja	Gültige Statements:	FORMAT INPUT
		Gültiges Kommando:	Keines

### Beispiel:

```
FORMAT IP=OFF
```



# 141

## IS - 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.

<b>Mögliche Werte</b>	ON	Wenn IS=ON gesetzt ist, wird der Wert eines Feldes nicht angezeigt, falls er mit dem vorherigen Wert des Feldes identisch ist.  <b>Anmerkung:</b> Erzeugt ein DISPLAY- oder WRITE-Statement unter Verwendung der VERT-Option oder der Schrägstrich-Notation (/) mehrzeilige Ausgaben, so gilt IS=ON nur für die jeweils erste Zeile.
	OFF	Es wird keine automatische Unterdrückung angewendet.
<b>Standard-Einstellung</b>	OFF	
<b>Angabe innerhalb der Session</b>	ja	
<b>Gültige Statements</b>	DISPLAY FORMAT WRITE	
<b>Gültiges Kommando</b>	none	



### Anmerkungen:

1. Mit dem Statement `SUSPEND IDENTICAL SUPPRESS` können Sie die Wirkung von IS=ON für einen einzelnen Datensatz unterdrücken.
2. Der IS-Parameter kann in Verbindung mit den Parametern `ES` und `ZP` zur Unterdrückung der Ausgabe von Leerzeilen eingesetzt werden.
3. Siehe auch *Parameter zur Beeinflussung der Ausgabe von Feldern im Leitfaden zur Programmierung*.

**Beispiel:**

```
FORMAT IS=ON
```

# 142

## ISIZE - Größe des Puffers für Initialisierung

---

Dieser Natural-Profilparameter bestimmt die Größe des Puffers für die Natural-Initialisierung (ISIZE).

<b>Mögliche Werte</b>	8 - 32767	Puffergröße in KB.
<b>Standard-Einstellung</b>	16	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	



### Anmerkungen:

1. Der ISIZE-Puffer dient dazu, die Parameter vorzuhalten, mit denen Natural initialisiert wird, sowie die Arbeitsbereiche und Tabellen, die Natural während der Initialisierung benutzt.
2. Der Profilparameter ISIZE wird ignoriert, wenn er in einer Parameterkette angegeben wird, die mit einem Profilparameter [SYS](#) oder [PROFILE](#) oder in einem alternativen Natural-Parametermodul (gemäß Angabe im Profilparameter [PARM](#)) aktiviert wird.



# 143

## ITERM - Session-Beendigung bei Initialisierungsfehler

Dieser Natural-Profilparameter gibt an, ob die Natural-Session beim Auftreten eines Initialisierungsfehlers fortgesetzt werden soll oder nicht.

<b>Mögliche Werte</b>	ON	Wenn ein Session-Initialisierungsfehler auftritt, wird die Session unmittelbar nach der Initialisierungsfehlermeldung beendet.
	OFF	<p>Wenn ein Session-Initialisierungsfehler auftritt, geschieht Folgendes:</p> <ul style="list-style-type: none"> <li>■ Im Online-Modus werden die Initialisierungsfehler angezeigt, und Sie können wählen, ob Sie die Session fortsetzen oder sie beenden wollen.</li> <li>■ Im Batch-Modus werden die Initialisierungsfehler angezeigt, und die Session wird fortgesetzt - was möglicherweise Fehler oder unerwünschte Ergebnisse zur Folge haben kann. Falls keine weiteren Fehler während der Session-Ausführung auftreten, wird während der Session-Ausführung anstelle der Meldung NAT9995 die Meldung NAT9964 (mit Bedingungscode 4) ausgegeben.</li> </ul> <p><b>Anmerkung:</b> Die Einstellung ITERM=OFF wird nicht berücksichtigt, wenn zu Beginn der Natural-Session das Systemkommando INPL auf dem Natural-Stack abgelegt wird, d.h. bei STACK=INPL. In diesem Fall wird zwangsweise die Einstellung ITERM=ON hergestellt.</p>
<b>Standard-Einstellung</b>	OFF	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	



**Anmerkung:** Zu beachten ist, dass die Einstellung von ITERM irrelevant ist, wenn der Profilparameter IMSG auf OFF gesetzt ist, weil dann sämtliche Initialisierungsfehler unterdrückt werden und die Session fortgesetzt wird.



# 144

## ITRACE - Interne Trace-Funktion aktivieren/deaktivieren

---

Dieser Natural-Profilparameter dient zum Aktivieren bzw. Deaktivieren der internen Trace-Funktion.



**Wichtig:** Benutzen Sie diesen Parameter nicht ohne vorherige Rücksprache mit dem Software AG Support.

<b>Mögliche Werte</b>	ON	Die Trace-Daten werden an die SYSRDC-Utility übergeben.
	OFF	Es werden keine Trace-Daten an die SYSRDC-Utility übergeben.
<b>Standard-Einstellung</b>	OFF	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	ja	Innerhalb einer Natural-Session kann das Terminalkommando %TRI benutzt werden, um die interne Trace-Funktion zu aktivieren bzw. zu deaktivieren.



**Anmerkung:** Die interne Trace-Funktion ist in erster Linie zur internen Benutzung durch die Software AG für Debugging-Zwecke gedacht.



# 145

## KD - Anzeige der PF-Tastennamen

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

<b>Mögliche Werte</b>	ON	Die den PF-Tasten zugewiesenen Namen werden angezeigt.	
	OFF	Die den PF-Tasten zugewiesenen Namen werden nicht angezeigt.	
<b>Standard-Einstellung</b>	OFF		
<b>Angabe innerhalb der Session</b>	ja	Gültige Statements:	FORMAT
		Gültiges Kommando:	Keines



### Anmerkungen:

1. Wenn `KD=ON` gesetzt ist, werden die PF-Tastenzuweisungsinformationen automatisch bei jeder mit `INPUT`, `WRITE`, `DISPLAY` und `PRINT` erzeugten Ausgabe am unteren Bildschirmrand angezeigt.
2. Da die Anzeige der PF-Tastenzuweisungsinformationen zwei Zeilen in Anspruch nimmt, muss die logische Seitenlänge (siehe Session-Parameter `PS`) entsprechend um zwei Zeilen reduziert werden.

### Beispiel:

```
FORMAT KD=ON
```



# 146

## KEY - PA-, PF- und CLEAR-Tastenbelegungen zuweisen

Dieser Natural-Profilparameter dient dazu, den PA-, PF- und CLEAR-Tasten bei Video-Terminals Belegungen zuzuweisen.

<b>Mögliche Werte</b>	Beliebige Zeichenkette	Belegungen können den Tasten PA1 bis PA3, PF1 bis PF24 und der Taste CLEAR zugewiesen werden.  <b>Anmerkung:</b> Die den einzelnen Tasten zugewiesene Belegung kann eine beliebige Zeichenkette sein. Diese Zeichenkette muss ein Natural-Systemkommando oder ein benutzerspezifisches (Benutzerprogramm-)Kommando sein. Wenn die Belegung eingebettete Leerzeihen enthält, muss sie in Hochkommas gesetzt werden.
<b>Standard-Einstellung</b>	keine	
<b>Dynamische Angabe</b>	nein	
<b>Angabe innerhalb der Session</b>	ja	
<b>Anwendungsprogrammierschnittstelle</b>	USR4005N	Siehe <i>SYSEXT Utility</i> in der Dokumentation <i>Debugger und Dienstprogramme</i> .



### Anmerkungen:

1. Die mit dem Profilparameter KEY angegebenen Belegungen sind nur am Natural-NEXT-Eingabeaufforderungszeichen gültig.
2. Die gesamte Zeichenkette, die mit dem Profilparameter KEY angegeben wird, muss in Klammern gesetzt werden (außer KEY=OFF). KEY=OFF macht alle Tastenbelegungen rückgängig.

**Beispiele:**

```
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
```

Dieser Natural-Profilparameter steuert die Umsetzung von Eingabezeichen in Kleinschreibung in die entsprechenden Zeichen in Großschreibung.

<b>Mögliche Werte</b>	ON	Es erfolgt keine Umsetzung der Eingabezeichen von Klein- in Großbuchstaben.
	OFF	Alle Zeichen in Kleinschreibung werden von Natural in die entsprechenden Zeichen in Großschreibung umgesetzt. Ausnahme: Eingabe-Daten, die mit dem STACK-Statement auf dem Natural-Stack abgelegt wurden.
<b>Standard-Einstellung</b>	OFF	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	ja	<b>Anmerkung:</b> Um die Umsetzung von Klein- in Großschreibung dynamisch während der aktiven Natural-Session zu aktivieren bzw. zu deaktivieren, sollten Sie die Terminalkommandos %L or %U benutzen.
<b>Anwendungsprogrammierschnittstelle</b>	USR1005N	Siehe <i>SYSEXT Utility</i> in der Dokumentation <i>Debugger und Dienstprogramme</i> .



#### Anmerkungen:

1. Dieser Parameter gilt nicht für Natural-Stack-Daten, die mit dem STACK-Statement auf dem Natural-Stack abgelegt wurden.
2. Die Umsetzung von Klein- in Großschreibung kann auch von einem TP Monitor Interface ausgeführt werden, bevor die Kontrolle an Natural übergeben wird. Die entsprechenden TP-Monitor-Parameter müssen überprüft werden, um eine korrekte Umsetzung sicherzustellen.
3. Es kann auch eine vom Benutzer gelieferte Umsetzungstabelle zur Umsetzung von Klein- in Großbuchstaben benutzt werden; siehe Makro [NTUTAB1](#) (in der Beschreibung des Profilparameters UTAB1).



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.

<b>Mögliche Werte</b>	beliebige Zeichen	Sie können eine Zeichenkette mit bis zu 10 Zeichen definieren.  <b>Anmerkung:</b>  1. Sie können die Zeichenkette wahlweise in Apostrophen (') angeben; in diesem Fall darf die Zeichenkette jedes beliebige Zeichen enthalten. 2. Eine Zeichenkette, die Anführungszeichen (") oder eine schließende Klammer enthält, muss in Apostrophen stehen. 3. Ein Leerzeichen in einer nicht durch Apostrophe eingegrenzten Zeichenkette wird durch ein Zirkumflex (^) dargestellt.
<b>Standard-Einstellung</b>	Keine	
<b>Angabe innerhalb der Session</b>	ja	
<b>Gültige Statements</b>	FORMAT	Parameter kann dynamisch mit dem FORMAT-Statement angegeben werden.
	DISPLAY	Parameter kann auf Statement- und/oder Element-Ebene angegeben werden.
<b>Gültiges Kommando</b>	keines	



#### Anmerkungen:

1. Die Session-Parameter **LC** und **IC** schließen einander aus.
2. Der Parameter **LC** kann auch bei Feldern des Formats U (Unicode) benutzt werden.

3. Informationen zum Unicode-Format entnehmen Sie dem Dokument *Unicode- und Codepage-Unterstützung in der Natural-Programmiersprache, Session-Parameter*, Abschnitt *EMU, ICU, LCU, TCU im Vergleich zu EM, IC, LC, TC*.
4. Siehe auch *Parameter zur Beeinflussung der Ausgabe von Feldern im Leitfaden zur Programmierung*.

**Beispiel:**

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

# 149

## LCU - Vorangestellte Unicode-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.

<b>Mögliche Werte</b>	beliebige Zeichen	Sie können eine Zeichenkette mit bis zu 10 Zeichen definieren. <b>Anmerkung:</b> <ol style="list-style-type: none"><li>1. Sie können die Zeichenkette wahlweise in Apostrophen (') angeben; in diesem Fall darf die Zeichenkette jedes beliebige Zeichen enthalten.</li><li>2. Eine Zeichenkette, die Anführungszeichen (") oder eine schließende Klammer enthält, muss in Apostrophen stehen.</li><li>3. Ein Leerzeichen in einer nicht durch Apostrophe eingegrenzten Zeichenkette wird durch ein Zirkumflex (^) dargestellt.</li></ol>
<b>Standard-Einstellung</b>	Keine	
<b>Angabe innerhalb der Session</b>	ja	
<b>Gültige Statements</b>	FORMAT	Parameter kann dynamisch mit dem FORMAT-Statement angegeben werden.
	DISPLAY	Parameter kann auf Statement- und/oder Element-Ebene angegeben werden.
<b>Gültiges Kommando</b>	keines	



### Anmerkungen:

1. Der Session-Parameter LCU ist identisch mit dem Session-Parameter LC. Der Unterschied ist, dass die vorangestellten Zeichen immer im Unicode-Format gespeichert werden. Dadurch können Sie die vorangestellten Zeichen mit gemischten Zeichen aus unterschiedlichen Codepages

eingeben. Es ist sichergestellt, dass immer das korrekte Zeichen angezeigt wird, unabhängig von der installierten System-Codepage.

2. Die Session-Parameter LCU und ICU schließen einander aus.

Siehe auch:

- *Parameter zur Beeinflussung der Ausgabe von Feldern im Leitfaden zur Programmierung*
- *Unicode- und Codepage-Unterstützung in der Natural-Programmiersprache, Session-Parameter, Abschnitt EMU, ICU, LCU, TCU im Vergleich zu EM, IC, LC, TC.*

# 150

## LE - 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.

<b>Mögliche Werte</b>	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.  <b>Anmerkung:</b> LE=ON gilt nur für Programme, die von einer in der Systemdatei FUSER abgelegten Library geladen werden, d.h. der Library SYSTEM, oder einer Library mit einem Namen, der nicht mit dem Präfix SYS 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.	
<b>Standard-Einstellung</b>	OFF		
<b>Dynamische Angabe</b>	ja		
<b>Angabe innerhalb der Session</b>	ja	Gültige Statements:	SET GLOBALS
		Gültiges Kommando:	GLOBALS
<b>Programmierschnittstelle (API)</b>	USR1005N	Siehe <i>SYSEXT Utility</i> in der Dokumentation <i>Debugger und Dienstprogramme</i> .	



**Anmerkungen:**

1. Der Parameter `LE` gilt bei `READ`, `FIND` und `HISTOGRAM`-Statements, bei denen ein Limit angegeben ist (siehe [Beispiel](#)).
- 2.
3. Das Limit kann entweder ein (mit dem `LIMIT`-Statement angegebenes) globales Limit oder ein schleifenspezifisch festgesetztes Limit sein.
4. In einer Natural-Session kann der Profilparameter `LE` durch den Session-Parameter `LE` überschrieben werden.

### 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
```

Bei `LE=OFF`: Nach 10 Datensätzen wird die Schleife ohne eine Meldung beendet.

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

# 151

## LFILE - Logische Systemdatei-Definition

---

▪ LFILE-Parameter-Syntax .....	466
▪ NTLFILE-Makro-Syntax .....	467
▪ Beispiel für LFILE-Parameter .....	467
▪ Beispiele für NTLFILE-Makros .....	468

Dieser Natural-Profilparameter enthält Angaben über die physische Datenbankdatei, die zu einer logischen Systemdatei für Software AG-Produkte zugeordnet werden soll. Er entspricht dem Makro `NTLFILE` im Natural-Parametermodul.

<b>Mögliche Werte</b>	Siehe <i>LFILE-Parameter-Syntax</i> .	
<b>Standard-Einstellung</b>	keine	
<b>Dynamische Angabe</b>	ja	Dieser Parameter kann nur dynamisch angegeben werden. Im Natural-Parametermodul wird stattdessen das Makro <code>NTLFILE</code> benutzt.
<b>Angabe innerhalb der Session</b>	nein	
<b>Anwendungsprogrammierschnittstelle</b>	USR0011N USR2004N (empfohlen)	Siehe <i>SYSEXT Utility</i> in der Dokumentation <i>Debugger und Dienstprogramme</i> .



#### Anmerkungen:

1. `LFILE` und `NTLFILE` können für Software AG-Produkte verwendet werden, die ihre eigenen Systemdateien haben (z.B. Con-nect, Natural Review), um anzugeben, wo sich eine solche Datei befinden soll. Diese Produkte verwenden in ihren DDMs (Data Definition Modules) die Datenbankkennung (DBID) 255 und eine **logische Dateinummer** (FNR). Mit dem Profilparameter `LFILE` bzw. dem Makro `NTLFILE` geben Sie an, welche **physische Dateinummer** und Datenbankkennung (DBID) (und, falls zutreffend, Passwort und Chiffrierschlüssel) zu dieser logischen Dateinummer zugeordnet werden sollen.
2. Natural bildet die logische Dateinummer auf die physische Dateinummer und Datenbankkennung ab („Mapping“) und benutzt sie bei allen Datenbankaufrufen.
3. `LFILE` kann außerdem benutzt werden, um ein so genanntes Scratch Pad File mit der logischen Dateinummer 212 zu definieren; siehe auch Profilparameter `ROSY`, *Natural Scratch-Pad File* in der *Operations*-Dokumentation und *Defining a Scratch-Pad File in Installation for z/OS*.

## LFILE-Parameter-Syntax

Der Profilparameter `LFILE` wird wie folgt angegeben:

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

Dabei ist

Syntax-Element	Wert	Bedeutung
<i>logical-fnr</i>	1 - 251	Logische Dateinummer (LFL). Dieser Subparameter muss angegeben werden.
<i>physical-dbid</i>	0 - 65535, außer 255	Physische Datenbankkennung (DBID). Die Datenbankkennung 255 ist für logische Systemdateien für Software AG-Produkte reserviert.
<i>physical-fnr</i>	1 - 65535	Physische Dateinummer (FNR).
<i>password</i>	1 - 8 Zeichen	Passwort und Chiffrierschlüssel sind nur bei Datenbankdateien erforderlich, die mit der Adabas-Security-Funktion passwortgeschützt oder verschlüsselt worden sind.
<i>cipher-key</i>	1 - 8 numerische Zeichen	
RO	nicht zutreffend	Schreibschutzkennung (Read Only).



**Anmerkung:** Um die verschiedenen logischen Dateien zu definieren, muss der Profilparameter LFILE mehrmals angegeben werden (durch ein Komma oder ein Leerzeichen abgetrennt); siehe [Beispiel für LFILE-Parameter](#).

## NTLFILE-Makro-Syntax

```
NTLFILE logical-fnr,physical-dbid,physical-fnr,password,cipher-key,RO
```



**Anmerkungen:**

1. Beschreibung der Syntax-Elemente und mögliche Werte siehe [LFILE-Parameter-Syntax](#).
2. Um mehrere logischen Dateien zu definieren, muss das Makro NTLFILE mehrmals angegeben werden; siehe [Beispiele eines NTLFILE-Makros](#).

## Beispiel für LFILE-Parameter

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

## Beispiele für NTLFILE-Makros

---

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

# 152

## LIBNAM - Name der externen Programm-Load Library

---

Dieser Natural-Profilparameter dient zur Angabe des Namens der Load Library, aus der Programme dynamisch geladen werden sollen, wenn Natural im z/OS Batch-Modus oder unter TSO benutzt wird.

Mögliche Werte	Zeichenkette	Der 8-Byte DDNAME der Load Library.
Standard-Einstellung	keine	
Dynamische Angabe	ja	
Angabe innerhalb der Session	nein	



### Anmerkungen:

1. Dieser Natural-Profilparameter gilt nur im z/OS Batch-Modus und unter TSO.
2. Unter z/OS muss außerdem ein JCL-Statement mit einem DDNAME angegeben werden, der mit der LIBNAM-Einstellung übereinstimmt. Programme werden standardmäßig aus der Job Steplib geladen.



# 153 LOG (für interne Verwendung)

---

Dieser Parameter ist für die interne Nutzung durch Natural reserviert.



**Vorsicht:** Ändern Sie seine Einstellung nicht.



# 154

## LS - Zeilenlänge

---

- Profilparameter LS ..... 474
- Session-Parameter LS ..... 474
- Angabe in Statements ..... 475

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.

## 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.

<b>Mögliche Werte</b>	35 - 250	Höchstwert der pro Zeile zulässigen Zeichen.
	0	Die physische Zeilenlänge ist zu benutzen (meistens 132).
<b>Standard-Einstellung</b>	0	
<b>Dynamische Angabe</b>	ja	

## Session-Parameter LS

---

<b>Mögliche Werte</b>	2 - 250	Höchstwert der pro Zeile zulässigen Zeichen.
	0	Nur beim Statement SET GLOBALS gestattet. Der Wert Null wird durch die physische Zeilenlänge ersetzt.
<b>Standard-Einstellung</b>	Physische Zeilenlänge.	
<b>Gültiges Kommando</b>	GLOBALS	
<b>Gültige Statements</b>	FORMAT SET GLOBALS	
<b>Programmierschnittstelle (API)</b>	USR1005N	Siehe <i>SYSEXT Utility</i> in der Dokumentation <i>Debugger und Dienstprogramme</i> .



**Anmerkung:** Beim Anmelden 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.

## Angabe in Statements

---

Wenn der Session-Parameter LS in einem Statement angegeben wird, erfolgt die Auswertung des LS-Parameters bei der Kompilierung.

<b>Gültige Statements</b>	DISPLAY INPUT WRITE	Der Parameter kann auf Statement-Ebene angegeben werden.
---------------------------	---------------------------	--



# 155

## LT - Limit für Verarbeitungsschleifen

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

<b>Mögliche Werte</b>	1 - 2147483647 0	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.  <b>Anmerkung:</b> Innerhalb einer Session: 0 bis $n$ ; dabei ist $n$ = Wert des Profilparameters LT beim Start der Session.
<b>Standard-Einstellung</b>	99999999	
<b>Dynamische Angabe</b>	yes	
<b>Angabe innerhalb der Session</b>	yes	
<b>Gültige Statements</b>	SET GLOBALS	<b>Anmerkung:</b> 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 gesetzt wurde.
<b>Gültiges Kommando</b>	GLOBALS	
<b>Programmierschnittstelle (API)</b>	USR1005N	Siehe <i>SYSEXT Utility</i> in der Dokumentation <i>Debugger und Dienstprogramme</i> .



### Anmerkungen:

1. 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`.

2. 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`.
3. 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 jedoch 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.
4. 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.
5. In einer Natural-Session kann die Einstellung des Profilparameters LT mit dem Session-Parameter LT überschrieben werden.

# 156 MADIO - Maximale Anzahl der DBMS-Aufrufe zwischen Bildschirm-Ein-/Ausgaben

---

Dieser Natural-Profilparameter dient zur Angabe der maximal zulässigen Anzahl der DBMS-Aufrufe zwischen zwei Bildschirm-Ein-/Ausgabe-Operationen.

<b>Mögliche Werte</b>	30 - 32767	Maximale Anzahl der DBMS-Aufrufe.
	0	MADIO=0 gibt an, dass kein Limit wirksam sein soll.
<b>Standard-Einstellung</b>	512	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	
<b>Anwendungsprogrammierschnittstelle</b>	USR1005N	Siehe <i>SYSEXT Utility</i> in der Dokumentation <i>Debugger und Dienstprogramme</i> .
	USR1068N *	* Empfohlen.



**Anmerkung:** Bei Überschreitung des angegebenen Limits wird das Natural-Programm unterbrochen und der Benutzer wird davon durch die Natural-Fehlermeldung 1009 in Kenntnis gesetzt.



# 157

## MAINPR - Standard-Ausgabe-Report-Nummer

Dieser Natural-Profilparameter kann verwendet werden, um Programmausgaben von Natural-Systemausgaben zu trennen, was besonders im Batch-Modus nützlich sein kann.

<b>Mögliche Werte</b>	0 - 31	Gültige Druckernummer.
<b>Standard-Einstellung</b>	0	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	
<b>Anwendungsprogrammierschnittstelle</b>	USR6002N	Siehe <i>SYSEXT Utility</i> in der Dokumentation <i>Debugger und Dienstprogramme</i> .



### Anmerkungen:

1. Dies gilt für Programmausgaben für Report 0, so wie sie von den Statements `DISPLAY`, `PRINT`, `WRITE` oder `INPUT` erzeugt werden (Ausnahme: `INPUT`-Statements, die nicht-geschützte Eingabefelder (Feldattributangabe `AD=A`) oder änderbare Eingabefelder (`AD=M`)) haben.
2. Wird der Parameter `MAINPR` angegeben, dann werden Programm-Ausgaben für Report 0, die normalerweise auf einem dem Report 0 zugewiesenen Drucker ausgegeben werden, stattdessen auf dem mit dem Parameter `MAINPR` angegebenen Drucker ausgegeben; während Systemausgaben (`NEXT-`, `DATA`-Aufforderungszeichen usw.) immer auf dem primären Ausgabegerät (Report 0) ausgegeben werden; die `MAINPR`-Einstellung muss eine gültige Druckernummer sein (0 - 31).
3. Ein logischer Drucker, der der angegebenen Report-Nummer entspricht, muss in Natural angegeben werden. Die Definition eines Druckers erfolgt mit dem Profilparameter `PRINT` bzw. dem Makro `NTPRINT` oder automatisch mittels `JCL` (im Batch-Modus oder unter TSO).
4. Der Parameter `MAINPR` gilt nicht für Ausgaben aus Systemprogrammen in der Natural-System-Library `SYSLIB`, diese werden immer auf dem primären Ausgabegerät (Report 0) ausgegeben. Sie können jedoch mit der Option `USEMAINPR` des Systemkommandos `LIST` die Ausgabe an den in `USEMAINPR` angegebenen Drucker leiten. Beschreibung der Option `USEMAINPR` des `LIST`-Kommandos siehe *Einstellungen* in der *Systemkommandos*-Dokumentation.



# 158

## MAXCL - Maximale Anzahl an Programmaufrufen

Dieser Natural-Profilparameter dient dazu, die maximale Anzahl an Programmaufrufen anzugeben, die zwischen zwei Bildschirm-Ein-/Ausgabe-Operationen gestattet werden.

<b>Mögliche Werte</b>	10 - 32767	Maximale Anzahl an Programmaufrufen.
	0	MAXCL=0 gibt an, dass kein Limit wirksam sein soll.
<b>Standard-Einstellung</b>	50	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	
<b>Anwendungsprogrammierschnittstelle</b>	USR1005N	Siehe <i>SYSEXT Utility</i> in der Dokumentation <i>Debugger und Dienstprogramme</i> . * Empfohlen.
	USR1068N *	



**Anmerkung:** Bei Überschreiten des angegebenen Limits wird das Natural-Programm unterbrochen und der Benutzer wird mit einer entsprechenden Fehlermeldung (NAT1029) davon in Kenntnis gesetzt.



# 159

## MAXROLL - Anzahl der CMROLL-Aufrufe vor

### vorübergehender Session-Unterbrechung

---

Er dient zur Angabe der maximalen Anzahl an CMROLL-Aufrufen, nach der eine Natural-Session vorübergehend unterbrochen wird; d.h., ein potenzieller Roll-out des Natural Thread durchgeführt werden soll.

<b>Mögliche Werte</b>	1 - 32767	Anzahl der CMROLL-Aufrufe.
	0	MAXROLL=0 gibt an, dass keine bedingten CMROLL-Anforderungen abgesetzt werden.
<b>Standard-Einstellung</b>	128	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	



#### Anmerkungen:

1. Dieser Natural-Profilparameter gilt nur unter Com-plete und CICS.
2. Der MAXROLL-Parameter kann dazu verwendet werden, die Häufigkeit von bedingten CMROLL-Anforderungen zu steuern. So bedeutet z.B. MAXROLL=128, dass bei der Kompilierung nach jedem 128. Statement eine bedingte CMROLL-Anforderung abgesetzt wird.
3. In bestimmten Fällen setzt der Natural-Nukleus eine bedingte CMROLL-Anforderung ab (Wait Time = 0), dies geschieht insbesondere beim Kompilieren nach jedem Statement. Es dient dazu, das CPU-Zeitfenster (unter Com-plete) zurückzusetzen, um einen automatischen Abbruch wegen Überschreitung des CPU-Zeitfensters zu vermeiden; dies wirkt sich jedoch negativ auf die Performance aus.
4. Hinweis zu CMROLL: Das Aufrufen von CMROLL stellt die Natural-Schnittstelle zur WAIT- oder DELAY-Funktionalität dar (siehe auch Beispielprogramm SUSPEND in der Library SYSEXT); wenn Sie CMROLL aufrufen, können Sie einen Verzögerungszeitraum bzw. eine Wartezeit als Parameter mitgeben. Wenn eine Session im CMROLL warten muss, dann werden gemeinsam

genutzte Ressourcen freigegeben, z.B. ein Thread in Com-plete oder ein gemeinsam genutzter Thread in CICS, und als Folge davon wird ein potenzieller Roll-out des Natural Thread ausgeführt. Das Aufrufen von CMROLL mit einer Verzögerung von 0 wird als bedingt bezeichnet, da die Session tatsächlich nicht für eine bestimmte Zeitdauer zu warten braucht; wenn jedoch weitere Sessions auf einen Thread warten, dann wird diese Session vorübergehend unterbrochen, was dazu führen kann, dass ein Roll-out des Natural Thread erfolgt. Wenn bei CICS keine weitere Session wartet, wird nur ein EXEC CICS SUSPEND ausgeführt, um AICA-Abbrüche zu vermeiden.

# 160

## MAXYEAR - Maximale Jahresangabe bei

### Datums-/Zeitwerten

---

Dieser Natural-Profilparameter dient zur Angabe des maximalen Werts für den Jahr-Bestandteil der Datums- und Uhrzeitwerte, die als Konstante oder als Terminaleingabe eingegeben werden können.

<b>Mögliche Werte</b>	2699	Als maximales Jahr kann 2699 eingegeben werden; d.h., der maximale Datumswert, der eingegeben werden kann, ist 2699-12-31.
	9999	Als maximales Jahr kann 9999 eingegeben werden; d.h., der maximale Datumswert, der eingegeben werden kann, ist 9999-12-31.
<b>Standard-Einstellung</b>	2699	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	



#### Anmerkungen:

1. MAXYEAR=9999 ändert den maximalen Datumswert, der eingegeben werden kann, von 2699-12-31 in 9999-12-31.
2. Bevor Sie den Wert für MAXYEAR auf 9999 setzen, sollten Sie sorgfältig prüfen, ob in Ihrer Anwendung arithmetische Operationen oder Zuweisungen mit bzw. von Datums- und Zeitwerten in Feldern mit anderen Datenformaten als Datum und Uhrzeit vorkommen, und die nötigen Änderungen vornehmen. Sonst können unerwartete Überläufe auftreten, die zur Ausführungszeit zu Natural-Fehlern führen.

Beispielsweise sollten Sie Folgendes überprüfen:

- Redefinitionen von Datums-/Uhrzeitfeldern in P6/P12-Felder
- Zuweisungen von Datums-/Uhrzeitwerten zu Nicht-Datums-/Uhrzeitfeldern, z.B. P6 := D

- arithmetische Operationen mit Datums-/Uhrzeitwerten, bei denen das Ergebnis einem Nicht-Datums-/Uhrzeitfeld zugewiesen wird, z.B. `P6 := D + 7`
- Eingaben in Datums-/Uhrzeitfelder, die später in arithmetischen Operationen mit Nicht-Datums-/Uhrzeitfeldern verwendet werden, z.B.:

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

Zum Überprüfen Ihrer Anwendung empfehlen wir Ihnen die Benutzung des Software AG-Produkts Natural Engineer.

Die Einstellung von MAXYEAR hat Auswirkungen auf:

- Prüfung der Datums-/Uhrzeitkonstanten durch den Compiler, z.B. `P6 := D'2699-12-31'`
- INPUT-Statements mit eingabefähigen oder änderbaren Datums-/Uhrzeitfeldern
- MOVE EDITED-Statements mit Ursprungs- oder Ziel-Datums-/Uhrzeitfeldern
- IS (D)-Option in logischen Bedingungen
- MASK-Option in logischen Bedingungen mit Vier-Zeichen-Jahresprüfung (YYYY)
- VAL-Systemfunktion mit Datumsfeld als Operand

Sie sollten sich vergewissern, dass die MAXYEAR-Einstellungen identisch sind für

- das Katalogisieren und Ausführen einer Natural-Anwendung
- Natural RPC Server und Natural RPC Clients

Siehe auch:

- *Formate D – Datum und T - Zeit im Leitfaden zur Programmierung*
- *Datums- und Zeitkonstanten im Leitfaden zur Programmierung*
- Session-Parameter [EM](#) in der *Parameter Reference*-Dokumentation
- Profilparameter [YD](#) in der *Parameter Reference*-Dokumentation

# 161 MC - Anzahl multipler Feldwerte

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.

<b>Mögliche Werte</b>	1 - 191	Anzahl der Werte.
	0	<b>Anmerkung:</b> Wenn MC=0 angegeben wird, 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.
<b>Standard-Einstellung</b>	1	
<b>Angabe innerhalb der Session</b>	ja	
<b>Gültige Statements</b>	DISPLAY FORMAT INPUT PRINT WRITE	
<b>Gültiges Kommando</b>	keines	



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

**Beispiel:**

FORMAT MC=5

# 162

## MENU - Menü-Modus

---

Dieser Natural-Profilparameter dient zum Ein- bzw. Ausschalten des Menü-Modus.

<b>Mögliche Werte</b>	ON	Menü-Modus eingeschaltet.
	OFF	Menü-Modus ausgeschaltet.
<b>Standard-Einstellung</b>	ON	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	ja	Innerhalb einer Natural-Session kann die Einstellung des MENU-Parameters mit dem Natural-Systemkommando MAINMENU aufgehoben werden.



# 163

## ML - Position der Meldungszeile

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.

<b>Mögliche Werte</b>	B	Natural-Meldungen werden am unteren Rand des Bildschirms angezeigt.	
	T	Natural-Meldungen werden am oberen Rand des Bildschirms angezeigt.	
<b>Standard-Einstellung</b>	T		
<b>Dynamische Angabe</b>	ja		
<b>Angabe innerhalb der Session</b>	ja	Gültige Statements:	SET CONTROL 'M'
		Gültiges Kommando:	
<b>Programmierschnittstelle (API)</b>	USR1005N	Siehe <i>SYSEXT Utility</i> in der Dokumentation <i>Debugger und Dienstprogramme</i> .	



**Anmerkung:** Informationen zum Operanden 'M' siehe auch Natural Terminalkommando %M.



# 164

## MONSIZE - Größe des SYSTP Monitor-Puffers

---

Dieser Natural-Profilparameter dient zur Angabe der Größe des Puffers, der von der Monitor-Funktion der Utility SYSTP (siehe *Utilities*-Dokumentation) benutzt wird.

<b>Mögliche Werte</b>	5 - 256	Puffergröße in KB.
	0	Wenn MONSIZE=0 oder falls der angeforderte Speicher nicht verfügbar ist, kann die Monitor-Funktion der SYSTP-Utility nicht benutzt werden; außer es wurde ein Monitor-Puffer mit dem Profilparameter BPI bzw. dem Makro NTBPI definiert.
<b>Standard-Einstellung</b>	0	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	



**Anmerkung:** Alternativ können Sie den äquivalenten Natural-Profilparameter [DS](#) oder das Parametermakro [NTDS](#) benutzen.



# 165

## MP - Maximale Seitenzahl eines Reports

---

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

<b>Mögliche Werte</b>	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.
<b>Standard-Einstellung</b>	0	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	
<b>Gültige Statements</b>	DISPLAY FORMAT PRINT WRITE	
<b>Gültiges Kommando</b>	keines	



**Anmerkung:** 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.



# 166

## MS - Manuelle Cursor-Positionierung

---

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

<b>Mögliche Werte</b>	ON	Siehe Beispiel weiter unten.
	OFF	Der Cursor wird ins nächste Eingabefeld plaziert, sobald der Wert des aktuellen Feldes vollständig eingegeben ist.
<b>Standard-Einstellung</b>	OFF	
<b>Angabe innerhalb der Session</b>	ja	
<b>Gültige Statements</b>	FORMAT INPUT	
<b>Gültiges Kommando</b>	keines.	

### Beispiel:

```
INPUT (MS=ON) #A #B
```



# 167

## MSGSF - Anzeigeformat von Systemfehlermeldungen

---

Dieser Natural-Profilparameter kann benutzt werden, um die Abschneidung von Natural-Systemfehlermeldungen zu verhindern.

<b>Mögliche Werte</b>	ON	Systemfehlermeldungen werden vollständig angezeigt; d.h. Programmname, Zeilennummer und eigentlicher Meldungstext.
	OFF	Systemfehlermeldungen werden in Kurzform angezeigt; d.h., es wird nur der eigentliche Meldungstext, aber nicht der Programmname und die Zeilennummer, angezeigt.
<b>Standard-Einstellung</b>	ON	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	ja	Innerhalb einer Natural-Session kann die Einstellung des Profilparameters MSGSF mit dem Natural-Terminalkommando %MSGSF aufgehoben werden.

Natural-Systemfehlermeldungen enthalten standardmäßig folgende Angaben:

- Name des Programms,
- Nummer der Zeile, die den Fehler verursacht hat,
- eigentlicher Text der Meldung.

Je nach Größe des Fensters, in dem die Meldung angezeigt wird, kann der Text abgeschnitten werden. Mit diesem Parameter können Sie eine solche Abschneidung verhindern.



# 168

## MT - Maximale CPU-Zeit

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



**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 von Endlosschleifen (bspw. aufgrund von Anwendungsfehlern) zu vermeiden. Diese Empfehlung gilt für Natural RPC- und Natural Development-Server.

<b>Mögliche Werte</b>	1 - 9999999	Maximale CPU-Zeit in Sekunden.  <b>Anmerkung:</b>  1. Wenn Natural Security installiert ist, kann der Profilparameter MT in Natural Security überschrieben werden. 2. Mit Natural Security ist der Höchstwert für den Profilparameter MT gleich 32767. 3. Um einen höheren Wert zu benutzen als mit dem MT-Profil- oder Session-Parameter angegeben, geben Sie MT=0 in Natural Security an.	
	0	MT=0 legt fest, dass kein Natural-CPU-Zeitlimit gelten soll.	
<b>Standard-Einstellung</b>	60		
<b>Dynamische Angabe</b>	ja		
<b>Angabe innerhalb der Session</b>	ja	Gültige Statements:	SET GLOBALS
		Gültiges Kommando:	GLOBALS
<b>Programmierschnittstelle (API)</b>	USR1005N	Siehe <i>SYSEXT Utility</i> in der Dokumentation <i>Debugger und Dienstprogramme</i> .	



**Anmerkungen:**

1. Dieser Natural-Profil- und Session-Parameter gilt nur für Programme, die im Batch-Betrieb, unter Natural Development Server (SPoD) oder unter Natural for TSO ausgeführt werden.
2. 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-Ein-/Ausgabe neu gestartet.
3. Das Limit für im interaktiven Betrieb laufende Programme wird von dem benutzten TP-Monitor gesteuert.
4. Der benutzbare Höchstwert wird von der Betriebssystemumgebung festgelegt. Überschreitet der mit dem MT-Parameter gesetzte Wert das vom Betriebssystem unterstützte Maximum, wird der Wert entsprechend der Betriebssystem-Vorgaben verringert.
5. Bei Systemumgebungen, die keine CPU-Zeitmessung unterstützen, wird das Limit als die verstrichene Zeit interpretiert. Bei Systemen ohne Zeitmessung wird das CPU-Zeitlimit ignoriert.
6. In einer Natural-Session kann der Profilparameter MT durch den Session-Parameter MT überschrieben werden.
7. Im Betrieb mit aktivierter zIIP-Unterstützung unter z/OS gilt der Profilparameter MT separat sowohl für die Verarbeitungsmodi TCB (außer unter CICS) als auch SRB (zIIP). Wenn im SRB-Modus  $MT=0$  gesetzt ist, benutzt Natural das existierende z/OS TCB-Zeitlimit, um Endlosschleifen zu vermeiden, weil es kein z/OS-CPU-Zeitlimit (z.B. den Parameter JCL TIME) für SRBs gibt.
8. Der Parameter MT wird unter CICS SRBs (zIIP), jedoch nicht unter CICS TCBs unterstützt. Wenn  $MT=0$  gesetzt ist, benutzt Natural im SRB-Modus das Runaway-Zeitlimit als CPU-Zeitbegrenzung. Ein Programmabbruch aufgrund einer Überschreitung des CPU-Zeitlimits, der bei Benutzung eines offenen CICS TCB auftritt, kann nicht wiederhergestellt werden. Er verursacht einen sofortigen AICA-Abbruch der CICS-Task sowie den Abbruch der Natural-Sitzung.

# 169

## NAFSIZE - Puffergröße für Natural Advanced Facilities

---

Dieser Natural-Profilparameter dient zur Angabe der Größe des Arbeitspufferspeichers, der von Natural Advanced Facilities benutzt wird.

<b>Mögliche Werte</b>	1 - 64	Pufferspeichergröße in KB.
	0	Wenn NAFSIZE=0 kann Natural Advanced Facilities nicht benutzt werden.
<b>Standard-Einstellung</b>	0	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	



### Anmerkungen:

1. Dieser Natural-Profilparameter gilt nur, wenn Natural Advanced Facilities installiert ist.
2. Alternativ können Sie den äquivalenten Natural-Profilparameter [DS](#) oder das Parametermakro [NTDS](#) benutzen.
3. Wenn Natural Advanced Facilities benutzt werden soll, siehe *NATSPPOOL-Initialisierung* in der *Natural Advanced Facilities-Dokumentation*.
4. Falls der angeforderte Speicher nicht verfügbar ist, kann Natural Advanced Facilities nicht benutzt werden.



# 170

## NAFUPF - Benutzerprofil für Natural Advanced Facilities

---

Dieser Natural-Profilparameter dient zur Angabe des Namens des Benutzerprofils für Natural Advanced Facilities.

<b>Mögliche Werte</b>	1 - 8 Zeichen	Name des Benutzerprofils.
<b>Standard-Einstellung</b>	keine	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	



### Anmerkungen:

1. Dieser Natural-Profilparameter gilt nur, wenn Natural Advanced Facilities installiert ist.
2. Siehe *NATSPool-Initialisierung* in der *Natural Advanced Facilities*-Dokumentation.



# 171

## NC - Verwendung von Natural-Systemkommandos

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

<b>Mögliche Werte</b>	ON	Systemkommandos können nicht verwendet werden.  <b>Ausnahmen:</b>  FIN, LAST, LOGOFF, LOGON, MAINMENU, RENUMBER, RETURN, SETUP und TECH.  <b>Anmerkung:</b>  1. Ist Natural Security installiert, sind ungeachtet der Einstellung des NC-Profilparameters alle von Ihnen mit Natural Security gesetzten Systemkommando-Beschränkungen (Command Restrictions) gültig. 2. In einer Natural Development Server-Umgebung auf Großrechnern wird der Wert OFF für den Natural Development Server angenommen, auch wenn NC=ON angegeben wurde. 3. Wurde NC=ON auf der Client-Seite angegeben, werden client-seitig abgesetzte Systemkommandos im Anschluss daran zurückgewiesen (siehe oben).	
	OFF	Alle Systemkommandos können verwendet werden.	
<b>Standard-Einstellung</b>	OFF		
<b>Dynamische Angabe</b>	ja		
<b>Angabe innerhalb der Session</b>	ja	Gültige Statements:	SET GLOBALS
		Gültiges Kommando:	
<b>Programmierschnittstelle (API)</b>	USR1005N	Siehe <i>SYSEXT Utility</i> in der Dokumentation <i>Debugger und Dienstprogramme</i> .	



**Anmerkungen:**

1. In einer Natural-Session kann der Profilparameter `NC` durch den Session-Parameter `NC` überschrieben werden.
2. Natural-Terminalkommandos und Benutzerkommandos (Objektmodul-Namen) sind vom `NC`-Parameter nicht betroffen.

# 172 NISN (für interne Verwendung)

---

Dieser Parameter ist für die interne Nutzung durch Natural reserviert.



**Vorsicht:** Ändern Sie seine Einstellung nicht.



# 173

## NL - 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.

<b>Mögliche Werte</b>	$n.m$	Die Länge wird in der Form $nn.m$ angegeben, wobei $n$ für die Stellen vor dem Komma (Dezimalpunkt) und $m$ für die Stellen nach dem Komma steht.  Die Angabe von Stellen nach dem Komma ist optional. Die Summe aus $n$ und $m$ darf nicht größer als 29 sein.  <b>Anmerkung:</b>  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.
<b>Standard-Einstellung</b>	Keine	
<b>Angabe innerhalb der Session</b>	ja	
<b>Gültige Statements</b>	DISPLAY FORMAT INPUT PRINT WRITE	
<b>Gültiges Kommando</b>	keines	



### Anmerkungen:

1. Der Parameter NL darf nicht für Gruppen angegeben werden.
2. Eine für ein Feld definierte Editiermaske setzt den NL-Parameter für dieses Feld außer Kraft.

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

**Beispiel:**

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

# 174

## NUCNAME - Name des gemeinsam genutzten Nukleus

---

Dieser Natural-Profilparameter dient zur Angabe des Namens des (umgebungsunabhängigen) gemeinsam genutzten Nukleus an, wenn er dynamisch geladen werden und nicht zum umgebungsabhängigen Natural-Nukleus hinzugelinkt werden soll.

<b>Mögliche Werte</b>	1 - 8 Zeichen	Gültiger Load-Modulname.
<b>Standard-Einstellung</b>	keine	
<b>Dynamische Angabe</b>	ja	<b>Anmerkung:</b> Wenn Sie diesen Parameter dynamisch angeben, können Sie verschiedene gemeinsam genutzte Nuklei (z.B. für die Produktion und zum Testen) zusammen mit demselben umgebungsabhängigen Natural-Nukleus benutzen, ohne den Nukleus wieder hinzu linken zu müssen.
<b>Angabe innerhalb der Session</b>	nein	



**Anmerkung:** Der Profilparameter NUCNAME wird ignoriert, wenn er in einer Parameterzeichenkette angegeben wird, die vom Profilparameter **SYS** oder **PROFILE** oder in einem alternativen Parametermodul (gemäß Angabe im Profilparameter **PARM**) aktiviert wird.

Weitere Informationen siehe *Environment-Independent Nucleus* in der *Installation for z/OS*-Dokumentation.



# 175 OBJIN - Benutzung von CMOBJIN als

## Natural-Eingabedatei

---

Dieser Natural-Profilparameter gibt an, ob die Datei CMOBJIN (siehe *Natural in Batch Mode* in der *Operations*-Dokumentation) für Eingabedaten verwendet werden soll, die mit dem INPUT-Statement im Batch-Modus geliefert werden.

<b>Mögliche Werte</b>	Y	Daten für ein Natural-INPUT-Statement werden von der Datei CMOBJIN gelesen.
	N	Die Datei CMOBJIN wird nicht benutzt. Sämtliche Daten für ein INPUT-Statement werden aus der Datei CMSYNIN gelesen.
	R	Natural bestimmt, anhand des Vorhandenseins bzw. Nichtvorhandenseins des Statements CMOBJIN DD/FILE in der Natural-Ausführungs JCL bzw. JCS, welche Option für eine bestimmte Session gewählt wurde.
<b>Standard-Einstellung</b>	R	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	ja	



**Anmerkung:** Dieser Natural-Profilparameter gilt nur im Batch-Modus.



# 176 OPF - 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.

<b>Mögliche Werte</b>	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.	
<b>Standard-Einstellung</b>	ON		
<b>Dynamische Angabe</b>	ja		
<b>Angabe innerhalb der Session</b>	ja	Gültige Statements:	SET GLOBALS
		Gültiges Kommando:	GLOBALS
<b>Programmierschnittstelle (API)</b>	USR1005N	Siehe <i>SYSEXT Utility</i> in der Dokumentation <i>Debugger und Dienstprogramme</i> .	



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



# 177

## OPRB - Datenbank-Open/Close-Befehlsverarbeitung

---

▪ OPRB-Parameter-Syntax .....	522
▪ Dynamisch angegebener OPRB-Parameter bei Natural Security .....	523
▪ OPRB für VSAM .....	524
▪ OPRB für Adabas .....	524
▪ NTOPRB-Makro-Syntax .....	526
▪ Beispiele für NTOPRB-Makros .....	526

Der Natural-Profilparameter `OPRB` steuert die Datenbank-Open/Close-Kommandos während einer Natural-Session, die auf eine Adabas- und VSAM-Datenbank zugreift.

Das Makro `NTOPRB` kann im Makro `NTPRM` als Alternative zum Profilparameter `OPRB` verwendet werden. Die maximale Länge einer `OPRB`-Parameterangabe beträgt 256 Bytes. Wenn Sie eine längere Angabe benötigen, benutzen Sie anstelle des Parameters `OPRB` das Makro `NTOPRB`.

<b>Mögliche Werte</b>	Siehe <a href="#">OPRB-Parameter-Syntax</a> .	
<b>Standard-Einstellung</b>	keine	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	



**Anmerkungen:**

1. Der Profilparameter `OPRB` und das Makro `NTOPRB` gelten nur bei Adabas- und VSAM-Datenbanken.
2. Grundsätzlich wird beim `OPRB`-Parameter eine der oben aufgeführten Syntax-Varianten verwendet, wobei die möglichen Inhalte der Zeichenketten *strings* vom Datenbanksystem abhängen.
3. Anstelle des `OPRB`-Parameters können Sie auch das Makro `NTOPRB` im Natural-Parametermodul benutzen.
4. Wenn Sie `OPRB`-Angaben machen möchten, die nicht für alle Datenbanken gelten sollen, sollten Sie den Parameter `OPRB` im Makro `NTPRM` benutzen (anstelle eines `NTOPRB`-Makros).

## OPRB-Parameter-Syntax

---

Der Profilparameter `OPRB` wird wie folgt angegeben:

### Variante 1 - Open-Anforderung für alle Datenbanken

```
OPRB=(string)
```

Mit dieser Syntax geben Sie eine Open-Anforderung für *alle* Datenbanken an.

## Variante 2 - Open-Anforderung für bestimmte Datenbanken

```
OPRB=(DBID=nn1,string,DBID=nn2,string,...)
```

Mit dieser Syntax geben Sie eine Open-Anforderung für bestimmte, individuelle Datenbanken an. Gemäß den Angaben im Makro `NTDB` kennzeichnet die angegebene Datenbankkennung (DBID) den Typ der Datenbank.

## Variante 3 - Open-Anforderung für bestimmte Datenbanken und Standard-Open-Anforderung für alle nicht ausdrücklich angegebenen Datenbanken

```
OPRB=(string1,DBID=nn1,string2,DBID=nn2,string3,...)
```

Mit dieser Syntax geben Sie eine Open-Anforderung für bestimmte, individuelle Datenbanken an (*string2* und *string3*) und außerdem eine Standard-Open-Anforderung (die initiale Zeichenkette *string1*), die für alle Datenbanken gilt, für die Sie keine individuelle Zeichenkette (*string*) angeben.

## Variante 4 - Open-Anforderung mittels (nicht) eingeschränktem Aufruf

```
OPRB=(DBID=nn1,NR=value,string,...)
```

Mit dem Subparameter `NR=value` geben Sie an, ob das Adabas-Open-Kommando als eingeschränkter (Restricted) oder nicht eingeschränkter (Non-Restricted) Aufruf ausgeführt werden soll. Dies steuert den in der Command Option 1 (COPT1) des OP-Kommandos eingestellten Wert.

Dabei ist:

Wert	Bedeutung
NR=OFF	Bewirkt ein eingeschränktes Open bei COPT1=R. Dies ist die Standardeinstellung.
NR=ON	Bewirkt ein <i>nicht</i> eingeschränktes Open, wenn COPT1=<leer> ist.

## Dynamisch angegebener OPRB-Parameter bei Natural Security

Ein dynamisch angegebener OPRB-Parameter gilt für alle Anmeldungen in Libraries, in deren Security-Profilen kein OPRB-Parameter angegeben ist. Für eine Anmeldung bei einer Library, in deren Security-Profil der OPRB-Parameter angegeben ist, wird jeder dynamisch angegebene OPRB-Parameter ignoriert. Es gilt immer der Parameter aus dem Security-Profil.

## OPRB für VSAM

Zeichenketten (*strings*), die für VSAM-Datenbanken angegeben werden können, sind im Abschnitt *OPRB Parameter for VSAM Databases* in der *Natural for VSAM*-Dokumentation beschrieben.

## OPRB für Adabas

Bei Adabas-Datenbanken ist der OPRB-Parameter erforderlich, wenn eine der beiden folgenden Bedingungen für die Natural-Session zutrifft:

- Es muss eine explizite Liste mit Adabas-Dateien zur Verfügung gestellt werden, auf die zugegriffen bzw. die geändert (Update) werden sollen. Dies ist zum Beispiel nötig, wenn eine Adabas Cluster-Änderung oder eine exklusive Dateikontrolle angefordert werden soll.
- Wenn sich eine einzelne logische Transaktion über zwei oder mehr Natural-Programme erstrecken soll und es deshalb nicht erwünscht ist, dass Natural beim Beenden eines gegebenen Natural-Programms ein `END TRANSACTION`-Statement und ein `CLOSE`-Kommando absetzt.

Möglicher Inhalt der Parameter-Zeichenkette	Bedeutung
ACC=( <i>file-list</i> )	Angabe einer Zugriffserlaubnis (Lesen) für die Dateien in der Dateiliste.
UPD=( <i>file-list</i> )	Angabe einer Zugriffs-/Änderungserlaubnis (Lesen/Schreiben) für die Dateien in der Dateiliste.
EXF=( <i>file-list</i> )	Angabe einer exklusiven Dateikontrolle: Keine anderen Benutzer dürfen auf die Datei zugreifen bzw. sie ändern.
EXU=( <i>file-list</i> )	Angabe einer exklusiven Änderungserlaubnis (exklusives Lesen/Schreiben) für die Dateien in der Dateiliste.
ACODE	Angabe der Option zum Erzwingen einer Benutzer-Zeichencodierung (Encoding) für A-Felder.  <b>Anmerkung:</b> Die erforderliche Zeichencodierung für ACODE wird von der aktuellen Einstellung des Parameters CP in der Natural-Session abgeleitet.
WCODE	Angabe der Option zum Erzwingen einer Benutzer-Zeichencodierung (Encoding) für W-Felder.  <b>Anmerkung:</b> Die erforderliche Zeichencodierung für WCODE ist immer 4095.
ARC	Angabe einer speziellen Datenarchitektur für Felder in dem Datensatz (Record) und in den Wertepuffern. Diese Definition hebt die Definition des Architekturschlüssels für Remote-Aufrufe in Entire Net-Work auf.



### Anmerkungen:

1. Weitere Informationen zu diesen Zeichenketten siehe Beschreibung des Adabas-Kommandos `OP` in der Adabas *Command Reference*-Dokumentation.
2. Falls der `OPRB`-Parameter im Natural-Parametermodul weggelassen wird oder `OPRB=OFF` als dynamischer Parameter angegeben wird, beginnt eine Natural-Session mit einem Adabas-Open-Kommando, wodurch ein `UPD` (Update: Zugriff/Änderung) der Natural-Systemdatei angefordert wird. Außerdem setzt Natural `RELEASE CID` (Adabas `RC`)-Kommandos ab, um alle ISN-Listen freizugeben (die in einer `RETAIN`-Klausel eines Natural-`FIND`-Statements angegebenen ISN-Listen werden nicht freigegeben).

Der Adabas-Datensatzpuffer (Record Buffer), der mit dem initialen Adabas `OP`-Kommando benutzt werden soll, kann explizit mitgegeben werden. Das Format ähnelt dem Format, das in einem Adabas-Datensatzpuffer für das `OP`-Kommando verwendet wird, jedoch dürfen keine eingeschlossenen Leerzeichen vorhanden sein und die gesamte Angabe muss in Klammern (nicht in Hochkommas) gesetzt werden.

**Beispiel 1:**

```
OPRB=(ACC=2,4,6,UPD=8.)
```

Gibt an, dass die Adabas-Dateien 2, 4 und 6 nur für den Zugriff verfügbar gemacht werden sollen und dass die Adabas-Datei 8 für die Änderung (Update), was auch den Zugriff impliziert) verfügbar gemacht werden soll.

**Beispiel 2:**

```
OPRB=(EXU=1,2,3.)
```

Gibt an, dass die Adabas-Dateien 1, 2 und 3 für diese Natural-Session unter exklusive Kontrolle gestellt werden sollen.

Bei Kombination der Schlüsselwörter `ACC`, `UPD` und `EXU` müssen die in der betreffenden Adabas-Dokumentation definierten Regeln beachtet werden. Wenn diese Schlüsselwörter angegeben werden, setzt Natural zu Beginn einer Natural-Session ein `OP`-Kommando ab und am Ende der Natural-Session ein `CL`-Kommando. Am Ende eines Natural-Kommandos werden nur die erforderlichen `RC`-Kommandos zur Freigabe von im Hold gehaltenen ISN-Listen abgesetzt.

In allen oben aufgeführten Situationen enthält das `OP`-Kommando, das immer zu Beginn einer Natural-Session abgesetzt wird, die Benutzerkennung für die Natural-Session im Feld Additions 1 im Adabas Control Block. Im Batch-Modus ist dies der Name des Jobs. Im TP-Modus ist es die Einstellung, die bei der Systeminitialisierung vom Natural-Interface-Modul geliefert wird. In beiden Fällen steht die verwendete Einstellung in der Natural-Systemvariablen `*INIT-USER` zur Verfügung.

## NTOPRB-Makro-Syntax

---

Die Syntax des Makros `NTOPRB` wird wie folgt angegeben:

```
NTOPRB dbid, 'string'
```



### Anmerkungen:

1. Mögliche Werte wie beim Parameter `OPRB`. Wenn Sie Natural mit VSAM benutzen, siehe auch die *Natural for VSAM*-Dokumentation.
2. Falls die Zeichenkette (*string*) sehr lang ist, können Sie sie in bis zu fünf Zeichenketten aufteilen, die durch Kommas voneinander getrennt werden müssen (siehe unten), da der Assembler nur einzelne Zeichenketten mit einer Länge von maximal 256 Bytes zulässt.

## Beispiele für NTOPRB-Makros

---

```
NTOPRB 12, 'ACC=40,UPD=20'  
NTOPRB 15, 'EXU=1, ', '2,3'
```

# 178

## OPT - Steuerung des Natural Optimizer Compiler

---

- OPT-Parameter-Syntax ..... 528
- NTOPT-Makro-Syntax ..... 528

Dieser Parameter dient zum Aktivieren bzw. Deaktivieren des Natural Optimizer Compiler und steuert die verschiedenen Kompilierungsoptionen. Er entspricht dem Makro `NTOPT` im Natural-Parametermodul.

<b>Mögliche Werte</b>	Siehe <i>Dynamischer Profilparameter OPT</i> in der <i>Optimizer Compiler</i> -Dokumentation.	
<b>Standard-Einstellung</b>	OFF	
<b>Dynamische Angabe</b>	ja	Dieser Parameter kann nur dynamisch angegeben werden. Im Natural-Parametermodul wird stattdessen das Makro <code>NTOPT</code> benutzt.
<b>Angabe innerhalb der Session</b>	ja	



**Anmerkung:** Dieser Natural-Profilparameter gilt nur, wenn der Natural Optimizer Compiler verwendet wird.

## OPT-Parameter-Syntax

Die Syntax für den Parameter `OPT` sieht z.B. wie folgt aus:

```
OPT=(INDX,OVFLW,ZD=OFF)
```

Weitere Syntaxbeispiele siehe *Dynamischer Profilparameter OPT* in der *Optimizer Compiler*-Dokumentation.

## NTOPT-Makro-Syntax

Die Syntax für das Makro `NTOPT` sieht z.B. wie folgt aus:

```
NTOPT 'INDX,OVFLW,ZD=OFF'
```

Weitere Syntaxbeispiele siehe *Makro NTOPT* in der *Optimizer Compiler*-Dokumentation.

# 179

## OSP - Parameter für z/OS Batch

---

▪ OSP-Parameter-Syntax .....	530
▪ NTOSP-Makro-Syntax .....	530
▪ Schlüsselwort-Subparameter .....	531
▪ Beispiel für OSP-Parameter .....	534
▪ Beispiel für NTOSP-Makro .....	534

Die Parameter für z/OS Batch können als Subparameter des Profilparameters [OSP](#) oder des Makros [NTOSP](#) angegeben werden.

<b>Mögliche Werte</b>	Siehe <a href="#">OSP-Parameter-Syntax</a> .	
<b>Standard-Einstellung</b>	Siehe <a href="#">Schlüsselwort-Subparameter</a> .	
<b>Dynamische Angabe</b>	ja	Der Parameter OSP kann nur dynamisch angegeben werden. Im Natural-Parametermodul wird das Makro <a href="#">NTOSP</a> verwendet.
<b>Angabe innerhalb der Session</b>	nein	

## OSP-Parameter-Syntax

Der Profilparameter OSP wird wie folgt angegeben:

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

Informationen zu Namen und Werten von Schlüsselwort-Subparametern siehe [Schlüsselwort-Subparameter](#).

## NTOSP-Makro-Syntax

Das Makro NTOSP wird wie folgt angegeben:

```
NTOSP ABEXIT=value, *
      DUMPDSN=value, *
      LBPNAME=value, *
      LEHDLR=value, *
      SUBPOOL=value, *
      TIOBSZ=(value1,value2), *
      USERID=value
```

Siehe [Schlüsselwort-Subparameter](#).

## Schlüsselwort-Subparameter

ABEXIT | DUMPDSN | LBPNAME | LEHDLR | SUBPOOL | TIOBSZ | USERID

### ABEXIT – Abbruch-Verarbeitung

ABEXIT=*value* gibt die Art der Abbruch-Verarbeitung innerhalb von Natural an.

Wert:	Erklärung:
ESTAE	Natural fängt alle Abbrüche ab und gibt entsprechende Fehlermeldungen aus. Dies ist der Standardwert.
SPIE	Nur Programmprüfungen (SOCX-Abbrüche) werden abgefangen.
OFF	Natural fängt überhaupt keine Abbrüche bzw. Programmprüfungen ab. Dieser Wert entspricht dem Profilparameter DU=FORCE.



**Anmerkung:** Die Einstellung ABEXIT=OFF wird nicht empfohlen, weil einige Funktionen, die das Abfangen von Abbrüchen benötigen, dann nicht mehr funktionieren. Durch Benutzung des Profilparameters MT wird anstelle des Fehlers NAT0953 ein Abbruch U0322 bewirkt, wenn die CPU-Zeitgrenze erreicht wird.

### DUMPDSN - Namenspräfix für Dump-Dataset

DUMPDSN=*value* kann benutzt werden, um das Namenspräfix für einen dynamisch zugewiesenen Dump-Dataset zu definieren. Danach schreibt der z/OS Service IEATDUMP die Daten jedes einzelnen Dumps in einen separaten Dataset statt in den Standard-Dataset (SYSUDUMP oder SYSMDUMP). Dies kann vor allem in Batch-Server-Umgebungen hilfreich sein, wenn mehrere Dumps von verschiedenen Natural-Bereichen geschrieben werden müssen.

Der vollständige Name des Dump-Dataset hat folgende Struktur:

```
value.D&YYMMDD.T&HHMMSS.&SYSNAME.&JOBNAME
```

Wert:	Erklärung:
1 - 8 Zeichen	High-Level Qualifier für den Dump-Dataset-Namen.
' ' (leer)	Es werden keine Dump-Datasets mittels IEATDUMP geschrieben. Stattdessen werden die Dumps in den Standard-Dump-Dataset (SYSUDUMP or SYSMDUMP) geschrieben (falls zugewiesen). Dies ist der Standardwert.

## LBPNAME - Gemeinsame Nutzung Lokaler Buffer Pools

LBPNAME=*value* steuert die gemeinsame Nutzung von lokalen Buffer Pools, wenn mehrere Natural-Sessions innerhalb desselben Region laufen.

Wert:	Erklärung:
<i>name</i>	Name der gemeinsam genutzten lokalen Buffer Pool-Umgebung. Länge: 1-8 Zeichen.
' '	LBPNAME=' ' bedeutet, die lokalen Buffer Pools werden nicht gemeinsam genutzt. Dies ist der Standardwert.



### Anmerkungen:

1. LBPNAME definiert den Namen der gemeinsam genutzten lokalen Buffer Pool-Umgebung und dient zur Standortfestlegung des gemeinsam genutzten lokalen Buffer Pool.
2. Wenn mehrere Natural-Sessions in einer z/OS-Batch- oder TSO-Region gleichzeitig laufen, dann ordnet jede Session Speicher für einen separaten Buffer Pool zu. Außer für den Natural-z/OS-Batch-Modus-Server werden die lokalen Buffer Pools standardmäßig nicht gemeinsam genutzt, das heißt, wenn die verschiedenen Sessions dieselben Natural-Objekte benutzen, müssen diese separat jeweils einmal für jede Session geladen werden. Wenn *name* angegeben wird, nutzen alle Natural-Sessions gemeinsam denselben lokalen Buffer Pool.

## LEHDLR – Benutzung einer LE-Fehlerbehandlungsroutine für aufrufende LE-Subprogramme

LEHDLR=*value* gibt an, ob Natural eine LE-Fehlerbehandlungsroutine für den Aufruf von LE-Subprogramme benutzt.

Wert:	Erklärung:
ON	Während des Aufrufs von LE-Subprogrammen wird von Natural eine LE-Fehlerbehandlungsroutine aktiviert. Das bedeutet, dass Natural die Kontrolle erhält, wenn bei der Ausführung eines LE-Subprogramms ein nicht behandelter LE-Fehler auftritt, und dann diesen Fehler (durch Ausgabe der Fehlermeldung NAT0954) behandeln kann. Dies ist der Standardwert.
OFF	Während des Aufrufs von LE-Subprogrammen wird von Natural keine LE-Fehlerbehandlungsroutine aktiviert. Das bedeutet, dass die LE-Enklave beendet wird und damit die Natural-Session verlorengeht, wenn bei der Ausführung eines LE-Subprogramms ein nicht behandelter LE-Fehler auftritt.



### Anmerkungen:

1. Informationen zu LE-Laufzeitoptionen siehe Beschreibung des Source-Moduls NATLEOPT in der *Installation for z/OS-Dokumentation*.

2. Informationen zum Betrieb von Natural mit der IBM Language Environment siehe *Natural Execution - Miscellaneous Topics, LE Subprograms* in der *Operations*-Dokumentation.

## SUBPOOL – Speicher-Subpool für GETMAIN-Anforderungen

SUBPOOL=*value* gibt den Speicher-Subpool für GETMAIN-Anforderungen an.

Wert:	Erklärung:
1 - 127	Subpool-Nummer.
0	Dies ist der Standardwert.



### Anmerkungen:

1. Der Subparameter SUBPOOL wird nur in dem Natural-Parametermodul berücksichtigt, das mit dem Batch-Treiber verlinkt ist, jedoch nicht in einem alternativen Parametermodul, das durch Angabe von PARM= aktiviert wird.
2. Da der Subparameter SUBPOOL nur während der Session-Initialisierung ausgewertet wird, kann er nicht als dynamischer Subparameter angegeben werden.

## TIOBSZ – Größe des primären E/A-Puffers für Batch-Verarbeitung

TIOBSZ=(*value1*, *value2*) gibt die Größe des primären E/A-Puffers für die Batch- und/oder Server-Verarbeitung an.

Wert:	Erklärung:
	Größe des primären E/A-Puffers in KB.
<i>value1</i>	Batch-Größe: 4 - 32 KB. Die Zuordnung erfolgt unterhalb der 16-MB-Grenze.
<i>value2</i>	Server-Größe: 32 - 999999 KB. Die Zuordnung erfolgt oberhalb der 16-MB-Grenze.
(8, 64)	Dies ist die Standardeinstellung.

### Alternative Angaben:

- TIOBSZ=10 oder TIOBSZ=(10) definiert nur die Batch-Größe.
- TIOBSZ=(, 33) definiert nur die Server-Größe. Der nicht angegebene Wert bleibt unverändert.

## USERID – Inhalt der Systemvariablen \*INIT-USER

USERID=*value* gibt den Inhalt der Systemvariablen \*INIT-USER an.

Wert:	Erklärung:
ON	Die Variable wird entweder auf die Benutzerkennung des Zugriffskontrollblocks (ACEE) gesetzt, wenn ein Security-Produkt (z.B. RACF or ACF2) beteiligt ist, oder auf den Wert des Parameters USER aus der Jobkarte.
OFF	Benutzerkennung ist der Jobname. Dies ist der Standardwert.



### Anmerkungen:

1. Der Inhalt von \*INIT-USER kann während der Session-Initialisierung durch den Benutzerkennungs-Exit NATUEX1 geändert werden.
2. Weitere Informationen siehe *Configuring Natural, Natural User Exits, NATUEX1 - User Exit for Authorization Control* in der *Operations*-Dokumentation.

## Beispiel für OSP-Parameter

```
OSP=(LBPNAME=NATTEST1,USERID=ON)
```

## Beispiel für NTOSP-Makro

```
NTOSP LBPNAME=NATTEST1,USERID=ON
```

# 180

## OUTDEST - Ausgabeziel für asynchrone Verarbeitung

---

Dieser Natural-Profilparameter gibt das Ziel an, an das eine Natural-Fehlermeldung, die von einer asynchronen Anwendung erzeugt wird, gesendet werden soll.

<b>Mögliche Werte</b>	1 - 8 Zeichen.	Ziel, an das eine Natural-Fehlermeldung gesendet werden soll.
<b>Standard-Einstellung</b>	Einstellung des Profilparameters <a href="#">SENDER</a>	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	



### Anmerkungen:

1. Dieser Natural-Profilparameter gilt nur bei Natural unter CICS und Com-plete.
2. Nachdem eine Fehlermeldung gesendet worden ist, beendet Natural die asynchrone Session.
3. Wann und wie Fehlermeldungen/Ausgaben von einer asynchronen Anwendung ausgegeben werden, hängt vom jeweils verwendeten TP-Monitor ab.

Weitere Informationen siehe:

- *Asynchronous Natural Processing under CICS*
- *Asynchronous Natural Processing under Com-plete/SMARTS*



# 181

## OVSIZE - Speicher-Thread-Überlaufgröße

---

Dieser Natural-Profilparameter gibt die maximale Gesamtgröße des variablen Speichers an, der von einer Natural-Session außerhalb ihres Speicher-Thread zugewiesen werden kann.

<b>Mögliche Werte</b>	0 - 2097151	Maximale Gesamtgröße des Speichers außerhalb des Thread in KB.
<b>Standard-Einstellung</b>	2097151	Der Speicher außerhalb des Thread wird nur durch die Bereichsgröße begrenzt.
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	



### Anmerkungen:

1. Wenn während einer Natural-Session der Speicher innerhalb des Thread aufgebraucht ist, kann zusätzlicher Speicher außerhalb des Thread zugewiesen werden. Der Parameter `OVSIZE` kann benutzt werden, um die Gesamtgröße des variablen Speichers zu begrenzen. Das hat keinen Einfluß auf den physischen Speicher (siehe Profilparameter `WPSIZE`), der immer außerhalb des Thread zugewiesen wird.
2. In Umgebungen ohne Thread (z.B. im Batch-Modus oder unter TSO) wird dieser Parameter nicht beachtet.



Dieser Natural-Profilparameter dient zur Angabe eines Objektmoduls, das Profilparameterdefinitionen enthält.

<b>Mögliche Werte</b>	1 - 8 Zeichen	Modulname.
<b>Standard-Einstellung</b>	keine	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	



#### Anmerkungen:

1. Diese Definitionen werden mit den verschiedenen Makros kodiert, die in der *Operations*-Dokumentation im Abschnitt *Building a Natural Parameter Module* aufgeführt sind. Die Makros werden anschließend assembliert, so dass ein Objektmodul entsteht, dessen Name vom Benutzer angegeben wird.
2. Wenn der Parameter `PARM` angegeben wird (entweder in dem angelinkten Parametermodul oder als dynamischer Parameter beim Natural-Session-Start), dann wird das entsprechende Modul geladen und die darin enthaltenen Parameterdefinitionen werden wirksam. Das Parametermodul wird dynamisch aus der Steplib geladen.
3. Unter CICS ist für dieses Parametermodul ein PPT-Eintrag erforderlich.
4. Im z/OS-Batch-Modus und unter TSO kann die aktuelle Steplib mit dem Profilparameter `LIBNAM` definiert werden.
5. Profilparameterdefinitionen, die vor der Verarbeitung des `PARM`-Parameters wirksam sind (z.B. Definitionen, die in dem angelinkten Parametermodul enthalten sind oder vorherige dynamische Parameterangaben) werden, mit Ausnahme der Profilparameter `ISIZE` und `NUCNAME`, aufgehoben, wenn das angegebene Parametermodul geladen wird. Deshalb sollten dynamische Parameter nach der Angabe des Parameters `PARM` angegeben werden.
6. Werden die Profilparameter `ISIZE` und `NUCNAME` in einem alternativen Parametermodul angegeben, bleiben sie unberücksichtigt.

7. Mit dem Makro `NTUSER` können Sie die Benutzung eines alternativen Parametermoduls einschränken.

Dieser Natural-Profilparameter gibt an, ob die PC-Zugriffsmethode bei der Benutzung von Natural Connection unterstützt werden soll oder nicht.

<b>Mögliche Werte</b>	ON	Die PC-Unterstützung ist aktiviert. Die Natural-Statements READ PC FILE or WRITE PC FILE können verwendet werden; siehe UPLOAD PC FILE und DOWNLOAD PC FILE.  <b>Anmerkung:</b> Wenn PC=ON ist, enthält die Systemvariable *DEVICE immer den Wert PC.
	OFF	Die PC-Unterstützung ist deaktiviert.
	NAM	Feldnamen werden gesendet, wenn Daten hoch- oder heruntergeladen werden.  <b>Anmerkung:</b> Dieser Wert gilt nur in Großrechnerumgebungen.
	NONAM	Es werden keine Feldnamen gesendet, wenn Daten hoch- oder heruntergeladen werden.  <b>Anmerkung:</b> Dieser Wert gilt nur in Großrechnerumgebungen.
<b>Standard-Einstellung</b>	(OFF, NAM)	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	ja	Zur Steuerung der PC-Unterstützung können die Terminalkommandos %+ und %- verwendet.



#### Anmerkungen:

1. Dieser Natural-Profilparameter gilt nur, wenn Natural Connection installiert ist.
2. Die Angabe von mehreren Werten erfolgt in einer Werteliste, siehe *Beispiel*.
3. Die für die Zugriffsmethode PC zu benutzenden Dateien müssen mit den Makros `NTPRINT` und `NETWORK` bzw. den Profilparametern `PRINT`, `WORK` oder `HCAM` definiert werden.

4. Weitere Informationen siehe Natural Connection-Dokumentation.

**Beispiel:**

PC=(ON, NONAM)

# 184

## PC - Anzahl der Periodengruppen-Ausprägungen

Dieser Session-Parameter 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.

<b>Mögliche Werte</b>	0 - 191	Anzahl der Werte.  <b>Anmerkung:</b> 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.
<b>Standard-Einstellung</b>	1	
<b>Angabe innerhalb der Session</b>	ja	
<b>Gültige Statements</b>	FORMAT INPUT DISPLAY WRITE PRINT	Der Parameter kann auf Statement- und/oder Elementebene (Feldebene) angegeben werden.
<b>Gültiges Kommando:</b>	keines	



**Anmerkung:** Dieser Session-Parameter kann nur im Reporting Mode verwendet werden.

**Beispiel:**

FORMAT PC=5

# 185

## PCNTRL - Druckersteuerzeichen

---

Dieser Natural-Profilparameter dient zur Angabe der Zeilenvorschubzeichen, die in Spalte 0 jeder Ausdruckzeile eingefügt werden.

<b>Mögliche Werte</b>	Beliebige Zeichenkette	Dieser Parameter kann im Zeichenformat oder im Hexadezimalformat angegeben werden.
<b>Standard-Einstellung</b>	' 0 - ' (IBM-Umgebungen)  <b>Anmerkung:</b> Dies ist die Standard-Einstellung gemäß dem ASA Standard, was bedeutet, dass ein Leerzeichen einen Zeilenvorschub von einer Zeile bewirkt, "0" von 2 Zeilen, und "-" von 3 Zeilen.  <b>Vorsicht:</b> Ändern Sie in einer IBM-Umgebung die Standard-Einstellung dieses Parameters nicht.	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	



# 186

## PD - 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.

<b>Mögliche Werte</b>	0 - 255	Maximale Anzahl der Seiten (Bildschirme).
<b>Standard-Einstellung</b>	50	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	ja	
<b>Gültige Statements:</b>	SET GLOBALS	
<b>Gültiges Kommando:</b>	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.



# 187

## PDPSIZE - Profiler-Daten-Pool-Größe

Dieser Natural-Profilparameter bestimmt die Größe des Profiler-Daten-Pool, der vom NaturalONE Profiler und dem Natural Profiler (siehe *Debugger und Dienstprogramme*-Dokumentation) im Batch-Modus benutzt wird. In der Profiler-Trace-Session wird dieser Daten-Pool als gemeinsam genutzter Zwischenspeicher verwendet, um die Datensätze verfolgter RDC-Sessions zu der Monitor-Session zu übertragen. Die Speicherzuordnung wird von der Monitor-Session bei der Initialisierung des NaturalONE Profiler bzw. der Profiler Utility vorgenommen.

<b>Mögliche Werte</b>	100 - 2097151	Größe des Profiler-Daten-Pool in KB.  Je nach Anzahl der zu übertragenden Ablaufverfolgungsdatensätze kann ein Erhöhen des PDPSIZE-Wertes erforderlich sein, um ein Überlaufen des des Daten-Pool-Speichers zu vermeiden. Im Falle eines Überlaufs pausiert die Ablaufverfolgungsaufzeichnung, was bei großen Anwendungen zu einer schlechteren Performance führen kann.  Nach dem Erhöhen der Speichergröße müssen Sie die Monitor-Session neu starten.
	0	Wenn PDPSIZE=0 gesetzt ist oder wenn die angegebene Größe den für den Daten-Pool verfügbaren Speicherplatz überschreitet, kann weder der NaturalONE Profiler benutzt noch die Profiler Utility im Batch-Modus ausgeführt werden.
<b>Standard-Einstellung</b>	500	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	



Dieser Natural-Profilparameter steuert beim Kompilieren, ob eine Prüfung mittels Option `ECHECK` oder `PCHECK` des Systemkommandos `COMPOPT` (siehe *Systemkommandos*-Dokumentation) beendet wird, nachdem in der Objekt-Source ein Syntaxfehler festgestellt worden ist. Außerdem kann mit `PECK` festgelegt werden, auf welche Weise auf die Syntaxfehler hingewiesen werden soll.

Mögliche Werte		
	S	Die Prüfung stoppt, wenn der erste Syntaxfehler festgestellt wird. Der Cursor wird in die Zeile gestellt, die den Fehler enthält, und es wird eine entsprechende Fehlermeldung ausgegeben (zum Beispiel: NAT0935).
	WS	Wie bei S, jedoch wird außerdem bei Beginn der Kompilierung der Meldungsspeicher (Message Buffer) geleert.
	F	<p>Das gesamte Objekt wird durchsucht, und alle Fehler werden auf einem Stapel abgelegt. Der Cursor wird in die Zeile gestellt, in der der erste Fehler gefunden wurde.</p> <p>Falls mehrere Fehler in derselben Zeile auftreten, erscheint in dieser Zeile eine Natural-Fehlermeldung, die darauf hinweist, dass bei der PCHECK/ECHECK-Validierung Inkonsistenzen festgestellt wurden.</p> <p>Falls mehrere Fehler in verschiedenen Zeilen auftreten, erscheint die erwähnte Natural-Fehlermeldung in der ersten Zeile.</p> <p>Sobald die Durchsuchung abgeschlossen ist, werden alle auf dem Stapel aufgelaufenen Fehler aufgelistet.</p>
	WF	Wie bei F, jedoch wird außerdem bei Beginn der Kompilierung der Meldungsspeicher (Message Buffer) geleert.
	WL	Wie bei L, jedoch wird außerdem bei Beginn der Kompilierung der Meldungsspeicher (Message Buffer) geleert.

<b>Standard-Einstellung</b>	L	<p>Das gesamte Objekt wird durchsucht, und alle Fehler werden auf den Stack gestellt. Der Cursor wird in die Zeile gestellt, in der der letzte Fehler gefunden wurde.</p> <p>Falls mehrere Fehler in derselben Zeile auftreten, erscheint in dieser Zeile eine Natural-Fehlermeldung, die darauf hinweist, dass bei der PCHECK/ECHECK-Validierung Inkonsistenzen festgestellt wurden.</p> <p>Falls mehrere Fehler in verschiedenen Zeilen auftreten, erscheint die erwähnte Natural-Fehlermeldung in der letzten Zeile.</p> <p>Sobald die Durchsuchung abgeschlossen ist, werden alle auf dem Stapel aufgelaufenen Fehler aufgelistet.</p>
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	

# 189

## PGP - Eigenschaften für externe Programme

---

■ PGP-Parameter-Syntax .....	554
■ NTPGP-Makro-Syntax .....	556
■ Beispiele für PGP-Parameter .....	557
■ Beispiele für NTPGP-Makros .....	557

Dieser Natural-Profilparameter kann zum Vordefinieren der Eigenschaften externer Programme benutzt werden. Er entspricht dem Makro `NTPGP` im Natural-Parametermodul.

<b>Mögliche Werte</b>	Siehe <i>PGP-Parameter-Syntax</i> .	
<b>Standard-Einstellung</b>	OFF	Es werden keine Eigenschaften für externe Programme definiert.
<b>Dynamische Angabe</b>	ja	Der Profilparameter PGP kann nur dynamisch angegeben werden. Im Natural-Parametermodul muss das Makro <code>NTPGP</code> verwendet werden.
<b>Angabe innerhalb der Session</b>	ja	Temporäres Statement <code>SET CONTROL 'P=value'</code> für jeden Aufruf des externen Programms.



**Anmerkungen:**

1. Beim Aufrufen eines externen Programms mit dem Natural-CALL-Statement stehen bereits mehrere Optionen zur Verfügung, die mit einem vorangehenden `SET CONTROL`-Statement angegeben werden können (z.B. `SET CONTROL 'P=S'`), um bestimmte Programmierschnittstelleneigenschaften für den anschließenden Aufruf anzufordern. Mit dem Natural-Profilparameter PGP bzw. dem Makro `NTPGP` besteht die Möglichkeit, diese Programmeigenschaften im Natural-Parametermodul oder dynamisch vorzudefinieren, anstatt sie im Natural-Anwendungsprogramm anzugeben, was Anlass zu Fehlern geben könnte.
2. Das Terminalkommando `%P=` gilt nur für den nächsten Aufruf eines externen Programms, und die `CALL`-Optionen werden nach Rückkehr von dem Aufruf ohne Bedingung zurückgesetzt, wohingegen der Profilparameter PGP die `CALL`-Option für ein externes Programm dauerhaft setzt.
3. Wenn für dasselbe Programm unterschiedliche Eigenschaften mit dem Profilparameter PGP oder mit einem Statement `SET CONTROL 'P=value'` definiert sind, werden diese Eigenschaften für das betreffende Programm zusammengeführt.

## PGP-Parameter-Syntax

---

Mit dem dynamischen Parameter PGP geben Sie zuerst den Programmnamen und danach eine oder mehrere Eigenschaften für dieses Programm an.

```
PGP=(program-name,property-1,property-2,...)
```

Oder:

PGP=OFF

**Anmerkungen:**

1. Der Wert OFF setzt alle zuvor definierten Eigenschaften für das Programm zurück.
2. Um PGP-Definitionen für verschiedene Programme anzulegen, muss PGP mehrfach (getrennt durch Komma oder Leerzeichen) angegeben werden.

Syntax Element	Bedeutung
<i>program-name</i>	Der Name des externen Programms, das die nachfolgend angegebenen Eigenschaften haben soll. Generische Namen werden durch Angabe eines Sterns (*) am Ende unterstützt.  Beispiel: 'TESTP*'  <b>Anmerkung:</b> Die Hochkommas müssen bei der dynamischen Angabe gesetzt werden.
<i>property-1</i> <i>property-2</i> ...	Die Werte für die verschiedenen Eigenschaften sind in der folgenden Tabelle beschrieben. Zu jeder Eigenschaft gibt es eine Lang- und eine Kurzform.  <b>Anmerkung:</b> Bei dynamischer Parameterangabe kann eine Eigenschaft durch ein vorangestelltes NO zurückgesetzt werden, z.B. NOS, um die Eigenschaft S zurückzusetzen.
OFF	PGP=OFF setzt alle zuvor definierten Eigenschaften für das Programm zurück.

**Erklärung der Eigenschaftswerte**

Eigenschaft	Kurzform	Erklärung	SET CONTROL
STDL	S	Standard Linkage unter CICS	P=S
STDLC	SC	Standard Linkage simuliert EXEC CICS LINK.	P=SC
STDLQ	SQ	Diese Eigenschaft gilt nur, wenn das Natural CICS Interface benutzt wird.  Standard-Linkage auf QR TCB unter CICS.  Diese Eigenschaft gilt nicht, wenn in Natural ein externes Subprogramm mit dem Profilparameter <b>CSTATIC</b> oder <b>RCA</b> definiert ist.  Diese Eigenschaft gilt auch für LE-Subprogramme. Informationen bezüglich der Unterstützung von IBM Language Environment (LE)-Subprogrammen durch Natural siehe Abschnitt <i>LE Subprograms</i> in der <i>Operations</i> -Dokumentation.	P=SQ
ROLL	V	Rollout für Aufruf in Thread-Umgebungen.	P=V
COMA	C	Übergabe von Werten in CICS Commarea.	P=C
CONT	CC	Übergabe von Parameterwerten in CICS Container.	P=CC

Eigenschaft	Kurzform	Erklärung	SET CONTROL
LEMAIN	L	Aufgerufenes Programm ist ein IBM LE-Hauptprogramm.  Informationen bezüglich der Unterstützung von IBM Language Environment (LE)-Subprogrammen durch Natural siehe Abschnitt <i>LE Subprograms</i> in der <i>Operations</i> -Dokumentation.	P=L
LESUB	LS	Aufgerufenes Programm ist ein IBM LE-Subprogram (nur IBM).  Informationen bezüglich der Unterstützung von IBM Language Environment (LE)-Subprogrammen durch Natural siehe Abschnitt <i>LE Subprograms</i> in der <i>Operations</i> -Dokumentation.	P=LS
IMSPCB	I	Übergabe der IBM IMS TM PCB-Adresse.	P=I
DIGR	D	Aufgerufenes Programm ist ein DIGNUS-Remote-Subprogramm.	P=D
DIGL	DL	Aufgerufenes Programm ist ein DIGNUS-Local-Subprogramm.	P=DL
SUOW	U	SUOW, separate CICS Unit of Work.	P=U
SUOWB	UT	Separate CICS Unit of Work mit Backout-Tolerierung.	P=UT
OFF		Löschung aller zuvor definierten Eigenschaften (nur bei dynamischer Parameterangabe).	

## NTPGP-Makro-Syntax

Mit dem Makro NTPGP geben Sie im Natural-Parametermodul zuerst den Programmnamen und danach eine oder mehrere Eigenschaften für dieses Programm an.

```
NTPGP program-name,property-1,property-2,...
```



### Anmerkungen:

1. Die Syntax-Elemente des Makros NTPGP entsprechen denen des Profilparameters PGP, siehe [Syntax-Elemente](#) und [Erklärung der Eigenschaftswerte](#).
2. Der Wert OFF kann beim Makro NTPGP nicht gesetzt werden.
3. Um PGP-Definitionen für verschiedene Programme anzulegen, muss das Makro NTPGP mehrfach angegeben werden.

## Beispiele für PGP-Parameter

---

```
PGP=(TESTPGM1,S)  
PGP=('ABX*',L,NOS)  
PGP=(MYPROG7,OFF,L,STD)
```

## Beispiele für NTPGP-Makros

---

```
NTPGP TESTPGM1,S  
NTPGP ABX*,L  
NTPGP MYPROG,L,STD
```



Dieser Natural-Profilparameter ermöglicht es Ihnen, eine Liste aller Profilparameter auszudrucken, die dynamisch beim Session-Start angegeben wurden.

<b>Mögliche Werte</b>	ON	<p><b>Im Batch-Modus:</b></p> <p>Zu Beginn der Session wird eine Liste der dynamisch angegebenen Profilparameter und ihrer Einstellungen in den Ausgabe-Dataset CMPLOG geschrieben. Falls CMPLOG nicht verfügbar ist, wird die Liste in den Standard-Ausgabe-Dataset CMPRINT geschrieben.</p> <p><b>Im Online-Modus unter TSO:</b></p> <p>Zu Beginn der Session wird eine Liste der dynamisch angegebenen Profilparameter und ihrer Einstellungen in den Ausgabe-Dataset CMPLOG geschrieben. Falls CMPLOG nicht verfügbar ist, wird die Liste zum Terminal geschickt.</p> <p><b>Im Online-Modus unter CICS:</b></p> <p>Zu Beginn der Session wird eine Liste der dynamisch angegebenen Profilparameter und ihrer Einstellungen zum Terminal geschickt.</p>
	OFF	Es wird keine Liste der dynamisch angegebenen Profilparameter geschrieben.
<b>Standard-Einstellung</b>	OFF	
<b>Dynamische Angabe</b>	ja	<p><b>Anmerkung:</b></p> <ol style="list-style-type: none"> <li>1. Wird der Parameter PLOG dynamisch angegeben, gilt er solange für alle nachfolgenden dynamischen Profilparameter, bis PLOG das nächste Mal angegeben wird. Damit können Sie ausschließen, dass Einzelparameter ausgedruckt werden, z.B. wenn deren Einstellungen Passwörter oder sonstige sensible Informationen enthalten, die nicht ausgedruckt werden sollen.</li> <li>2. Alle Parameter, die zwischen einer Angabe PLOG=OFF und einer nachfolgenden Angabe PLOG=ON angegeben werden, werden nicht ausgedruckt.</li> </ol>

		3. Wenn die dynamische Parameter-Zeichenkette mit PLOG=ON beginnt, werden die Parameter, die von einem im Natural-Parametermodul angegebenen Profilparameter <b>PROFILE</b> kommen, ebenfalls ausgedruckt, obwohl sie <i>vor</i> der dynamischen Parameter-Zeichenkette ausgewertet werden.
<b>Angabe innerhalb der Session</b>	nein	



**Anmerkungen:**

1. Dieser Natural-Profilparameter gilt nur im Batch-Modus, unter TSO und unter CICS.
2. Das Ausdrucken einer Liste aller Profilparameter auszudrucken, die dynamisch beim Session-Start angegeben wurden, kann nützlich sein, um festzustellen, welche dynamischen Profilparameter tatsächlich benutzt wurden, insbesondere wenn Profilparameter wie **PROFILE** oder **SYS** angegeben werden, die wiederum andere Profilparameter „enthalten“ (bei **PROFILE** oder **SYS** wird die gesamte Zeichenkette der von ihnen aktivierten Profilparameter aufgelistet).

# 191

## PM - Druck-/Anzeige-Modus

---

- Profilparameter PM ..... 562
- Session-Parameter PM ..... 563

## Profilparameter PM

Mit dem Natural-Profilparameter `PM` bestimmen Sie, wie Felder gedruckt oder angezeigt werden sollen.

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



### Anmerkungen:

1. `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.
2. Ausführliche Informationen zur Verwendung der Einstellung `PM=I` siehe *Unterstützung bidirektionaler Sprachen* in der *Unicode- und Codepage-Unterstützung*-Dokumentation.

## Session-Parameter PM

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

<b>Mögliche Werte</b>	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</i> -Dokumentation).
	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.
<b>Standard-Einstellung</b>	Keine	Es wird der Standard-Zeichensatz verwendet.
<b>Gültige Statements</b>	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	

# 192

## POS22 - Version 2.2 Algorithmus für Systemfunktion

### POS

---

Dieser Natural-Profilparameter ist obsolet und wird nur noch aus Kompatibilitätsgründen akzeptiert.



# 193

## PRINT - Druckdatei-Zuweisungen

---

▪ PRINT-Parameter-Syntax .....	568
▪ NTPRINT-Makro-Syntax .....	569
▪ Schlüsselwort-Subparameter für alle Umgebungen .....	570
▪ Schlüsselwort-Subparameter für AM=STD in allen Umgebungen .....	576
▪ Schlüsselwort-Subparameter für AM=STD in z/OS-Umgebungen .....	579
▪ Schlüsselwort-Subparameter für AM=CICS .....	581
▪ Schlüsselwort-Subparameter für AM=COMP (Com-plete) .....	583
▪ Schlüsselwort-Subparameter für AM=SMARTS (Com-plete) .....	583
▪ Schlüsselwort-Subparameter für AM=IMS .....	584
▪ Schlüsselwort-Subparameter für das DEFINE PRINTER-Statement .....	584

Dieser Natural-Profilparameter gibt an, welche Druckdateien (Printfiles) während der Session benutzt werden sollen. Er entspricht dem Makro `NTPRINT` im Natural-Parametermodul.

<b>Mögliche Werte</b>	Siehe <a href="#">PRINT-Parameter-Syntax</a> .	
<b>Standard-Einstellung</b>	Siehe Standardwerte der verschiedenen Schlüsselwort-Subparameter weiter unten.	
<b>Dynamische Angabe</b>	ja	Der Parameter <code>PRINT</code> kann nur dynamisch angegeben werden. Im Natural-Parametermodul wird stattdessen das Makro <code>NTPRINT</code> verwendet.
<b>Angabe innerhalb der Session</b>	nein	



**Anmerkungen:**

1. Der alte dynamische Parameter `PRINTER` kann als Synonym für `PRINT` angegeben werden.
2. Innerhalb einer Session können bis zu 31 logische Druckdateien (durchnummeriert von 1 bis 31) und die Hardcopy-Druckdatei (Nummer 0) verwendet werden.
3. Die für den Zugriff auf Druckdateien in unterschiedlichen Umgebungen verwendeten Software-Komponenten werden als Zugriffsmethoden bezeichnet. Für die Dauer einer jeden Natural-Session kann jeder logischen Druckdatei nur eine Zugriffsmethode zugewiesen werden. Die Zugriffsmethode für eine Druckdatei wird mit dem Schlüsselwort-Subparameter `AM` (siehe weiter unten) festgelegt.
4. In z/OS unter TSO und im Batch-Modus brauchen Druckdateien nicht in der JCL vordefiniert zu werden. Vorausgesetzt, sie werden mit dem Subparameter `AM=STD` definiert, können sie während der Session dynamisch durch ein Natural-Programm mittels des `DEFINE PRINTER`-Statements oder der Anwendungsprogrammierschnittstelle (API) `USR2021` (in der Library `SYSEXT`) zugewiesen werden.
5. Siehe auch *Print and Work File Handling with External Datasets in a Server Environment* in der *Operations*-Dokumentation.

## PRINT-Parameter-Syntax

---

Mit dem Parameter `PRINT` geben Sie zunächst eine oder mehrere logische Druckdateinummern an, und danach mehrere Schlüsselwort-Subparameter, mit denen Sie die Eigenschaften dieser Druckdateien festlegen:

```
PRINT=((print-file-numbers),keyword-subparameter=value,...)
```

Dabei ist:

Syntax-Element	Beschreibung
<i>print-file-numbers</i>	<p>Die Dateinummern müssen zuerst und in Klammern angegeben werden.</p> <ul style="list-style-type: none"> <li>■ Die Nummern können im Bereich von 1 bis 31 liegen.</li> <li>■ Die Reihenfolge der Nummern kann beliebig sein.</li> <li>■ Die Nummern müssen durch Kommas voneinander getrennt werden.</li> <li>■ Um einen Nummernbereich anzugeben, können Sie einen Bindestrich (-) verwenden.</li> </ul>
<i>keyword-subparameters</i>	<p>Die Schlüsselwort-Subparameter (die es für verschiedene Umgebungen gibt) werden weiter unten beschrieben.</p> <p>Falls für dieselbe Druckdatei schon eine frühere (oder Standard-)Einstellung existiert, werden nur die Werte für den jeweils angegebenen Schlüsselwort-Subparameter überschrieben, alle übrigen Werte bleiben unverändert.</p>



**Anmerkung:** Der Profilparameter PRINT kann mehrmals (abgetrennt durch ein Komma oder ein Leerzeichen) angegeben werden, um unterschiedliche Druckdatei-Definitionen anzulegen.

### Beispiele:

```
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)
```

```
PRINT=(( 2,12,18 ),AM=STD,DEST='PRINT**',OPEN=INITOBJ,CLOSE=CMD),PRINT=(( 0 ),AM=STD,DEST=HARDCOPX)
```

## NTPRINT-Makro-Syntax

Mit einem Makro NTPRINT geben Sie zunächst eine oder mehrere logische Druckdateinummern an, und danach mehrere Schlüsselwort-Subparameter, mit denen Sie die Eigenschaften dieser Druckdateien festlegen:

NTPRINT (*print-file-numbers*), *keyword-subparameter=value*, ...

Dabei ist:

Syntax-Element	Beschreibung
<i>print-file-numbers</i>	<p>Die Dateinummern müssen zuerst und in Klammern angegeben werden.</p> <ul style="list-style-type: none"> <li>■ Die Nummern können im Bereich von 1 bis 31 liegen.</li> <li>■ Die Reihenfolge der Nummern kann beliebig sein.</li> <li>■ Die Nummern müssen durch Kommas voneinander getrennt werden.</li> <li>■ Um einen Nummernbereich anzugeben, können Sie einen Bindestrich (-) verwenden.</li> </ul>
<i>keyword-subparameters</i>	<p>Die Schlüsselwort-Subparameter (die es für verschiedene Umgebungen gibt) werden weiter unten beschrieben.</p> <p>Falls für dieselbe Druckdatei schon eine frühere (oder eine Standard-)Einstellung existiert, werden nur die Werte für den jeweils angegebenen Schlüsselwort-Subparameter überschrieben, alle übrigen Werte bleiben unverändert.</p>



**Anmerkung:** Das Makro NTPRINT kann mehrmals angegeben werden, um unterschiedliche Druckdatei-Definitionen anzulegen.

**Beispiele:**

```
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
```

## Schlüsselwort-Subparameter für alle Umgebungen

Die folgenden Schlüsselwort-Subparameter stehen für alle Umgebungen zur Verfügung:

AM | DEST | OPEN | CLOSE | ROUTE | CP | SHIFT

## AM - Typ der Zugriffsmethode

Mit `AM=value` geben Sie den Typ der Zugriffsmethode an, die verwendet werden soll.

Wert	Bedeutung
STD	Sequenzielle Standard-Batch-Dateien (Batch, TSO).
COMP	Com-plete-Druckdateien.
CICS	Transiente CICS-Daten oder Zwischenspeicher.
NAF	Natural Advanced Facilities
IMS	IMS/TM-Bestimmungsorte.
PC	Entire Connection.
USER	Drittanbieter-Druck-Interface.
SMARTS	SMARTS-Druckdatei.
ESS	Entire System Server.
NOM	Entire Output Management.  <b>Anmerkung:</b> Druckt in eine Entire Output Management Container-Datei, ohne die Spool-Funktion des Betriebssystems zu benutzen. Einzelheiten siehe <i>Entire Output Management</i> -Dokumentation.
OFF	Ohne Zuweisung. Bei <code>FAMSTD=OFF</code> erfolgt keine automatische Zuweisung.  <b>Anmerkung:</b> <code>PRINT=OFF</code> ist gleichbedeutend mit <code>PRINT=((1-31)), AM=OFF</code> und hat keine Auswirkung auf die Angaben bei den übrigen Schlüsselwort-Subparametern.
0	Ohne Zuweisung. Bei <code>FAMSTD=OFF</code> erfolgt die automatische Zuweisung. Dies ist die Standard-Einstellung.



### Anmerkungen:

1. Für eine Online-Session muss allen zu verwendenden Druckdateien eine spezifische Zugriffsmethode zugewiesen werden.
2. Für eine Batch-Session werden alle Druckdateien, denen keine spezifische Zugriffsmethode zugewiesen wurden, automatisch festgestellt. Ihnen wird dann die Standard-Zugriffsmethode (`AM=STD`) zugewiesen, vorausgesetzt sie wurden in der JCL vordefiniert. Siehe auch [FAMSTD - Überschreiben der Arbeitsdatei-Zugriffsmethodenzuweisungen](#).
3. `PRINT=OFF` ist gleichbedeutend mit `PRINT=((1-31)), AM=OFF`. Es hat keine Auswirkung auf Einstellungen der Schlüsselwort-Subparameter. `PRINT=((0), AM=xxx)` bzw. `NTPRINT (0), AM=xxx` bestimmt die Hardcopy-Druck-Zugriffsmethode und ist gleichbedeutend mit dem Profilparameter `HAM=xxx`.

### CCHAR - Steuerzeichen zulassen

Mit `CCHAR=value` können Sie definieren, dass hexadezimale Ausgabe-Steuerzeichen für die Druckdatei-Ein-/Ausgabe unverändert durchgelassen werden.

Wert	Bedeutung
<code>a1</code> or <code>(a1, a2, ...)</code>	Angegeben werden kann ein einzelnes Hexadezimalzeichen oder eine Liste mit Hexadezimalzeichen in Klammern. Die Hexadezimalzeichen müssen im Bereich <code>x'01'</code> bis <code>x'3F'</code> liegen. Anstelle eines Hexadezimalzeichens ist auch ein Bereich von Hexadezimalzeichen zulässig.  Es gibt keinen Standardwert.
OFF	Mit <code>CCHAR=OFF</code> werden eventuelle frühere <code>CCHAR</code> -Definitionen zurückgesetzt.



#### Anmerkungen:

- Um Fehler bei der Ein-/Ausgabe zu vermeiden, setzt Natural die Ausgabe-Steuerzeichen `x'01'` bis `x'3F'` automatisch auf `'?'` um. Es gibt jedoch Fälle, in denen bestimmte Steuerzeichen für spezielle Zwecke benötigt werden.
- Für die angegebenen Druckdateien werden die Definitionen ersetzt, die in den Ausgabe-Umsetzungstabellen `NTTAB`, `NTTAB1` und `NTTABL` im Konfigurationsmodul enthalten sind bzw. die durch den entsprechenden dynamischen Profilparameter oder durch das entsprechende Makro im Natural-Parametermodul definiert sind.

#### Beispiele:

`CCHAR=17`

`CCHAR=19-1B`

`CCHAR=(03-06,0A,1B,3A-3F)`

### DEST - Name des externen Dataset

`DEST=value` gibt den Bestimmungsort (Destination) für den Ausdruck an.

Wert	Erklärung
1 - 8 Zeichen.	Bestimmungsort (Destination) für den Ausdruck  <b>Anmerkung:</b> Der Subparameter <code>DEST</code> entspricht dem <code>OUTPUT</code> -Wert des Statements <code>DEFINE PRINTER</code> (und kann mit einer <code>DEFINE PRINTER OUTPUT</code> -Angabe überschrieben werden).

Die Bedeutung des Subparameters `DEST` ist abhängig von der Zugriffsmethode:

Zugriffsmethode	Bedeutung von DEST
AM=STD	<p>DEST ist der logische Dataset-Name (DDNAME, LINK-Name, DTF-Name).</p> <p>Wenn der Bestimmungsort für mehrere Dateien gelten soll, müssen zwei Sterne (**) für die Dateinummer angegeben werden. Diese werden für jede Druckdatei durch die entsprechende logische Dateinummer ersetzt. Ein DEST-Wert, der zwei Sterne enthält, muss in Hochkommas gesetzt werden, wenn er als dynamischer Parameter verwendet werden soll.</p> <p>Die Standard-Einstellung ist DEST='CMPRT**' für z/OS-Umgebungen.</p>
AM=CICS	<p>Für Druckdateien unter CICS gibt es keinen Standard-Wert. Hier ist der Subparameter DEST zwingend erforderlich, d.h., dass CICS-Druckdateien ohne gültige DEST-Angabe ignoriert werden.</p> <p>Das Natural CICS Interface unterstützt außerdem eine Variable (siehe Parameter-StandardEinstellung TERMVAR=&amp;TID im Generierungsmakro NTCICSP), als Teil des DEST-Werts, der bei der Angabe durch die tatsächliche CICS-Terminalkennung ersetzt wird; siehe auch <i>Natural Print and Work Files under CICS</i> in der <i>TP Monitor Interfaces</i>-Dokumentation).</p>
AM=IMS	Gibt den IMS/TM-Bestimmungsort an.



**Anmerkung:** PRINT=((0), DEST=xxx) bzw. NTPRINT (0), DEST=xxx definiert den Bestimmungsort für den Hardcopy-Ausdruck und ist gleichbedeutend mit dem Natural-Profilparameter `HDEST=xxx`.

## OPEN - Zeitpunkt der Datei-Öffnung

Mit OPEN=*value* bestimmen Sie den Zeitpunkt der Datei-Öffnung:

Wert	Die Datei wird geöffnet ...
INIT	für die Ausgabe bei der Initialisierung der Session.
OBF	gemäß der OPEN-Standard-Einstellung für unterschiedliche Umgebungen (Batch, CICS, Com-plete, TSO).
OBJ	wenn die Ausführung des ersten Objekts, das auf die Arbeitsdatei zugreift, beginnt. Dies ist die allgemeine Standard-Einstellung, außer bei AM=COMP und AM=IMS.
OBJ1	wenn die Ausführung des ersten Objekts auf Ebene 1, das auf die Datei zugreift, beginnt. Andernfalls wird sie geöffnet, wenn zum ersten Mal auf sie zugegriffen wird.
ACC	wenn zum ersten Mal durch ein Statement auf sie zugegriffen wird. Dies ist die Standard-Einstellung bei AM=COMP und AM=IMS.
INITOBF	für die Ausgabe bei der Initialisierung der Session. Bei jeder anschließenden erneuten Öffnung der Datei wird der OPEN-Standard-Wert für unterschiedliche Umgebungen (Batch, CICS, Com-plete, TSO) gesetzt.
INITOBJ	für die Ausgabe bei der Initialisierung der Session. Jede anschließende erneute Öffnung der Datei wird durchgeführt, wenn die Ausführung des ersten Objekts, das auf die Arbeitsdatei zugreift, beginnt.

Wert	Die Datei wird geöffnet ...
INITOBJ1	wenn die Ausführung des ersten Objekts auf Ebene 1, das auf die Arbeitsdatei zugreift, beginnt. Andernfalls wird sie geöffnet, wenn zum ersten Mal auf sie zugegriffen wird.
INITACC	für die Ausgabe bei der Initialisierung der Session. Jede anschließende erneute Öffnung der Datei wird durchgeführt, wenn zum ersten Mal durch ein Statement auf sie zugegriffen wird.

### CLOSE - Zeitpunkt der Datei-Schließung

Mit `CLOSE=value` bestimmen Sie den Zeitpunkt der Datei-Schließung:

Wert	Die Datei wird geschlossen ...
OBJ	entweder wenn die Verarbeitung des Objekts, in dem zuerst auf die Datei zugegriffen wurde, beendet ist oder wenn der Kommandomodus, NEXT- oder MAINMENU-Modus erreicht ist.
CMD	wenn der Kommandomodus, NEXT- oder MAINMENU-Modus erreicht ist.  Dies ist die Standard-Einstellung bei <code>AM=NAF</code> , <code>AM=COMP</code> und <code>AM=IMS</code> .
FIN	beim Session-Ende (Dies ist die Standard-Einstellung bei <code>AM=STD</code> ).  <b>Anmerkung:</b> Bei <code>CLOSE=FIN</code> , verursacht ein <code>DEFINE PRINTER</code> -Statement einen Fehler, wenn der Drucker schon geöffnet wurde. Ein <code>CLOSE PRINTER</code> -Statement für den Drucker wird ignoriert.
USER	nur wenn die Datei offen ist und eine der folgenden Bedingungen zutrifft: <ul style="list-style-type: none"> <li>■ ein <code>CLOSE PRINTER</code>-Statement wird abgesetzt,</li> <li>■ ein <code>DEFINE PRINTER</code>-Statement wird abgesetzt,</li> <li>■ beim Session-Ende.</li> </ul>

### ROUTE - Logische Druckdatei-Weiterleitung

Mit `ROUTE=value` geben Sie an, ob die Weiterleitung einer logischen Druckdatei gemäß den Angaben in der `OUTPUT`-Klausel des `DEFINE PRINTER`-Statements erfolgen soll.

Wert	Bedeutung
ON	Die Druckdatei wird weitergeleitet. Die Ziel-Druckdatei kann jede verfügbare Druckdatei außer PC sein.  Dies ist die Standard-Einstellung.
OFF	Es erfolgt keine Weiterleitung der Druckdatei.
<i>am</i>	Die Weiterleitung der Druckdatei erfolgt nur an Drucker mit angegebener Zugriffsmethode (Access Method).  Als möglicher Wert kann jede gültige Druckdatei-Zugriffsmethode verwendet werden (siehe Beschreibung des Subparameters <code>AM</code> weiter oben).  PC ist als Zugriffsmethode nicht erlaubt.



**Anmerkung:** Druckdatei-Weiterleitung bedeutet, dass die Druckausgabe an jeden beliebigen freien Drucker weitergeleitet werden kann, wenn der Name, der in der OUTPUT-Klausel des DEFINE PRINTER-Statements definiert ist, einen Druckdatei-Bestimmungsort bezeichnet, der durch einen logischen Drucker definiert wird. Wird kein Drucker mit dem angegebenen Namen gefunden, kann die Druckausgabe an einen beliebigen freien Drucker weitergeleitet werden.

### CP - Codepage für die Druckausgabe

CP=*value* definiert die Codepage für die Druckausgabe.

Wert	Bedeutung
1 - 64 Zeichen.	Der Name der gewünschten Codepage.  Möglich ist eine beliebige Zeichenkette, sie muss jedoch mit einem der Codepage-Parameter CCSID, CCSN, IANA oder ALIAS des Makros NTCPAGE im Source-Modul NATCONFIG vordefiniert werden.



#### Anmerkungen:

1. Es wird angenommen, dass alle Codepage-Daten, z.B. Natural-Sources, Inhalt von Feldern mit Format A, in dieser Codepage gespeichert werden. Wenn mit dem Subparameter CP keine Codepage angegeben wird, dann wird die aus der Auswertung der Einstellung des Profilparameters CP resultierende Codepage verwendet.
2. Bei ausgeschalteter Natural-Codepage-Unterstützung (z.B. durch Parameter CP=OFF) wird jeder für diesen Parameter gelieferte Wert ignoriert.
3. Siehe auch Profilparameter CP und *Natural-Profilparameter und Parameter-Makros in der Unicode- und Codepage-Unterstützung-Dokumentation.*

### SHIFT - Datensätze nach Rechts verschieben

SHIFT=*value* gibt an, ob die Druckdatensätze um die Anzahl an Leerstellen *nnn* nach Rechts verschoben werden sollen.

Wert	Bedeutung
0 oder 1 - 248	Die Druckdatensätze werden nach Rechts verschoben, und die angegebene Anzahl Leerzeichen wird, beginnend mit Spalte 1, vor dem Datensatz eingefügt.
0	Die Druckdatensätze werden nicht verschoben.  Dies ist der Standardwert.



#### Anmerkungen:

1. Das ASA-Steuerzeichen in Spalte 0 ist davon nicht betroffen.

2. Die angegebene Anzahl Zeichen geht am Ende der Datensätze verloren (Shift-Out).

## Schlüsselwort-Subparameter für AM=STD in allen Umgebungen

Die folgenden Schlüsselwort-Subparameter stehen für die Zugriffsmethode **AM=STD** in allen Umgebungen zur Verfügung:

**RECFM** | **BLKSIZE** | **LRECL** | **TRUNC** | **PAD** | **PADCHRO** | **ASA** | **STRIP**

### RECFM - Standard-Datensatzformat des Dataset

Mit `RECFM=value` bestimmen Sie das Standard-Datensatzformat des Dataset.

#### Unterstützte Formate

F	Fest
V	Variable
U	Undefiniert
B	In Blöcke unterteilt
S	Segmentiert
A	ASA
M	Maschinensteuerzeichen

#### Mögliche Werte bzw. Wertekombinationen

Wert	Bedeutung
RECFM=F, RECFM=FA, RECFM=FM, RECFM=FB, RECFM=FBA, RECFM=FBM, RECFM=V, RECFM=VA, RECFM=VM, RECFM=VB, RECFM=VBA, RECFM=VBM, RECFM=VBS, RECFM=VBSA, RECFM=VBSM, RECFM=U, RECFM=UA, RECFM=UM	Diese Werte bzw. Wertekombinationen können angegeben werden.
RECFM=VBA	In variable Blöcke unterteilt, mit ASA.  Dies ist der Standardwert.



**Anmerkung:** Die `RECFM`-Angabe gilt nur, wenn in der JCL oder (nur bei z/OS) im Dataset-DCB kein Datensatzformat vordefiniert ist.

## BLKSIZE - Standard-Block-Größe des Dataset

Mit `BLKSIZE=value` bestimmen Sie die Standard-Blockgröße des Dataset.

Wert	Bedeutung
0 oder 8 - 32767	Standard-Blockgröße des Dataset (in Bytes).
1016	Dies ist der Standardwert.

Die `BLKSIZE`-Angabe gilt nur, wenn in der JCL oder (nur bei z/OS) im Dataset-DCB keine Blockgröße vordefiniert ist.

## LRECL - Standard-Datensatzlänge des Dataset

Mit `LRECL=value` bestimmen Sie die Standard-Datensatzlänge (in Bytes) des Dataset.

Wert	Bedeutung
0 oder 5 - 254	Datensatzlänge des Dataset (in Bytes).
0	Dies ist der Standardwert.



### Anmerkungen:

1. Dieser Subparameter wird insbesondere zur Prüfung beim Abschneiden (Trunkierung) und Auffüllen genutzt.
2. Für `RECFM=V (B)` enthält der `LRECL`-Wert ein 4-Byte-Datensatz-Deskriptorwort.
3. Bei Angabe von `LRECL=0` gilt Folgendes: Bei `RECFM=V(B)` nimmt `LRECL` den Standardwert `BLKSIZE-4` und `254` an. Bei `RECFM=U` nimmt `LRECL` den Standardwert `BLKSIZE` an. Bei `RECFM=F(B)` wird die maximale Datensatzlänge in dem momentan ausgeführten Natural-Programm genommen, wenn die Datei geöffnet wird. Wenn beim Öffnen der Datei keine Datensatzangabe aus einem Programm verfügbar ist, z.B. bei `OPEN=INIT`, wird eine Datensatzlänge von `132` genommen (plus 1 bei ASA oder ein Maschinensteuerzeichen und/oder plus 4 für ein Datensatzdeskriptorwort, wenn das Datensatzformat variabel ist).
4. Die `LRECL`-Angabe gilt nur, wenn in der JCL oder (nur bei z/OS) im Dataset-DCB keine Datensatzlänge vordefiniert ist.

### TRUNC - Trunkierung von Ausgabe-Datensätzen

TRUNC=*value* gibt an, ob die Ausgabe-Datensätze abgeschnitten werden oder nicht:

Wert	Bedeutung
ON	Ausgabe-Datensätze, die länger sind als die Datensatzlänge (LRECL) des Dataset, werden abgeschnitten. Dies ist der Standardwert.
OFF	Die Fehlermeldung NAT1512 wird ausgegeben, wenn ein Ausgabe-Datensatz länger ist als die Datensatzlänge des Dataset.

### PAD - Auffüllen von Ausgabe-Datensätzen

PAD=*value* gibt an, ob die Ausgabe-Datensätze aufgefüllt werden oder nicht (gilt nur für Datasets mit fester Datensatzlänge):

Wert	Bedeutung
ON	Ausgabe-Datensätze, die kürzer sind als die Datensatzlänge (LRECL) des Dataset, werden mit Füllzeichen aufgefüllt, die mit dem Subparameter PADCHRO definiert werden. Dies ist der Standardwert.
OFF	Die Fehlermeldung NAT1510 wird ausgegeben, wenn ein Ausgabe-Datensatz kürzer ist als die Datensatzlänge des Dataset.

### PADCHRO - Füllzeichen für Ausgabe-Datensätze

PADCHRO=*value* gibt das Zeichen an, das zum Auffüllen verwendet wird, wenn PAD=ON für die Druckdatei definiert wird.

Wert	Bedeutung
'x'	Ein Zeichen <i>x</i> in Hochkommas.
x'xx'	Ein Hexadezimalzeichen <i>xx</i> .
' '(leer)	Leerzeichen oder <i>x'40'</i> . Dies ist der Standardwert.

## ASA - Verwendung des Datensatzformats ASA

ASA=*value* gibt an, ob das Datensatzformat ASA verwendet wird.

Wert	Bedeutung
ON	In die ausgegebenen Druckdatensätze wird ein ASA-Zeichen eingefügt. Unter z/OS wird dadurch die Verwendung des Datensatzformats ASA erzwungen, und zwar ohne Rücksicht auf die RECFM-Einstellung im DCB oder die Einstellung im Subparameter RECFM.  Dies ist der Standardwert.
OFF	In die ausgegebenen Druckdatensätze wird kein ASA-Zeichen eingefügt.

## STRIP - Entfernen von nachgezogenen Leerstellen deaktivieren

STRIP=*value* kann das Entfernen von nachgezogenen Leerstellen deaktiviert werden.

Wert	Bedeutung
ON	Nachgezogene Leerstellen werden entfernt.  Dies ist der Standardwert.
OFF	Nachgezogene Leerstellen werden nicht entfernt.



**Anmerkung:** Bei sequenziellen Batch-Dateien (Zugriffsmethode [AM=STD](#)) werden nachgezogene Leerstellen entfernt, wenn der Dataset mit variablem Datensatzformat ([RECFM=VB](#)) definiert wird, um Plattenspeicherplatz zu einzusparen. Wenn anschließend andere Anwendungen auf diesen Datensatz zugreifen, kann es wegen fehlender Leerzeichen Probleme geben. Diese Probleme lassen sich mit der Einstellung `STRIP=OFF` vermeiden.

## Schlüsselwort-Subparameter für AM=STD in z/OS-Umgebungen

Die folgenden Schlüsselwort-Subparameter stehen für die Zugriffsmethode [AM=STD](#) in z/OS-Umgebungen zur Verfügung:

[REREAD](#) | [FREE](#) | [BUFNO](#) | [DISP](#) | [VMAX](#)

## REREAD - Schließen von Tape File Datasets

REREAD=*value* setzt die REREAD-Option für das Schließen der Banddatei.

Wert	Bedeutung
ON	Die REREAD-Option wird für das CLOSE SVC gesetzt. Dadurch wird der Datenträger (Volume) neu positioniert, um den Dataset erneut zu verarbeiten.  Dies ist der Standardwert.
OFF	Die REREAD-Option wird für das CLOSE SVC nicht gesetzt.

## FREE - Dataset-Freigabe beim Schließen der Datei

FREE=*value* gibt an, ob der Dataset freigegeben wird, wenn die Datei geschlossen wird.

Wert	Bedeutung
ON	Die FREE-Option wird für das CLOSE SVC gesetzt, d.h., der Dataset wird freigegeben, wenn er geschlossen wird (und nicht bei der Step-Beendigung).
OFF	Die FREE-Option wird für das CLOSE SVC <i>nicht</i> gesetzt.  Dies ist der Standardwert.

## BUFNO - Standard-Anzahl der z/OS-E/A-Puffer des Dataset

BUFNO=*value* definiert die Standard-Anzahl der z/OS-Ein-/Ausgabepuffer des Dataset.

Wert	Bedeutung
0 oder 0 - 255	Standard-Anzahl der z/OS-E/A-Puffer des Dataset
0	In diesem Fall ordnet z/OS fünf Ein-/Ausgabepuffer zu.  Dies ist der Standardwert.



### Anmerkungen:

1. Die Anzahl der Ein-/Ausgabepuffer hat drastische Auswirkungen auf die Performance beim Druckdateizugriff. Beachten Sie, dass der Speicher für die Ein-/Ausgabepuffer unterhalb der 16 MB-Grenze zugeordnet wird.
2. Die BUFNO-Angabe gilt nur, wenn der Parameter BUFNO nicht in der JCL des Dataset angegeben wird.

## DISP - Druckdatei für Änderung öffnen

DISP=*value* bestimmt, ob die Druckdatei zur Änderung geöffnet wird.

Wert	Bedeutung
MOD	Neue Datensätze werden am Ende der Datei hinzugefügt.
NOMOD	Die Arbeitsdatei wird von Beginn an neu geschrieben. Dies ist der Standardwert.



**Anmerkung:** Dieser Subparameter entspricht dem JCL DD Statement-Parameter DISP=MOD.

## VMAX - LRECL-Einstellung für variables Datensatzformat steuern

VMAX=*value* steuert die LRECL-Einstellung für eine Ausgabedatei mit variablem Datensatzformat (RECFM=V).

Wert	Bedeutung
ON	Vorausgesetzt, dass für die Datei ein BLKSIZE-Wert existiert, der nicht Null ist, wird mit VMAX=ON für variables Datensatzformat LRECL=BLKSIZE-4 gesetzt, unabhängig von der Einstellung in der DCB oder im LRECL-Subparameter.
NAT	LRECL wird auf die Länge +4 des größten Datensatzes im Anwendungsprogramm gesetzt, wenn dieser Wert kleiner ist als im LRECL in der DCB für den Dataset.
OFF	Es wird die LRECL-Einstellung aus der DCB für den Dataset oder die Einstellung im Subparameter LRECL verwendet. Dies ist der Standardwert.

## Schlüsselwort-Subparameter für AM=CICS

Die folgenden Schlüsselwort-Subparameter stehen für die Zugriffsmethode [AM=CICS](#) zur Verfügung:

[TYPE](#) | [DISP](#)

## TYPE - Typ des CICS-Speichermediums

TYPE=*value* gibt den Typ des zu verwendenden CICS-Speichermediums an.

Wert	Bedeutung
MAIN	Temporärer Hauptspeicher.
AUX	Temporärer Hilfsspeicher.
TD	Übergangsdaten.



**Anmerkung:** Der Standardwert hängt von der Einstellung des Subparameters **DEST** ab. Wenn der Wert des **DEST**-Subparameters zu einer gültigen CICS-Übergangsdaten-Warteschlange passt, dann ändert sich der Wert des Subparameters **TYPE** auf den Standardwert **TD**, andernfalls wird **MAIN** als Standardwert genommen.

## DISP - Temporäre CICS-Speicherwarteschlangen-Disposition

DISP=(*value1*, *value2*) gibt die temporäre CICS-Speicherwarteschlangen-Disposition an.

Wertepaar	Bedeutung
(NEW, KEEP)	Die Speicherwarteschlange wird gelöscht, wenn die Datei geöffnet wird. Dies ist die Standard-Einstellung.
(NEW, DELETE)	Die Speicherwarteschlange wird gelöscht, wenn die Datei geöffnet und wenn sie geschlossen wird.
(OLD, DELETE)	Die Speicherwarteschlange wird gelöscht, wenn die Datei geschlossen wird.
(OLD, KEEP)	Die Speicherwarteschlange wird nicht gelöscht.

### Beispiel:

DISP=(NEW, DELETE)



**Anmerkung:** Die DISP-Angabe gilt nicht für CICS-Übergangsdaten-Warteschlangen außerhalb der Partition.

## Schlüsselwort-Subparameter für AM=COMP (Com-plete)

Der folgende Schlüsselwort-Subparameter steht für die Zugriffsmethode **AM=COMP** (Com-plete) zur Verfügung:

DRIVER

### DRIVER - Name des Com-plete-Druck-Treibers

DRIVER=*value* gibt den Namen des zu verwendenden Com-plete-Druck-Treibers an.

## Schlüsselwort-Subparameter für AM=SMARTS (Com-plete)

Der folgende Schlüsselwort-Subparameter steht für die Zugriffsmethode **AM=SMARTS** (Com-plete) zur Verfügung:

DEST

### DEST - Logischer Drucker

DEST=*value* gibt den logischen Drucker an.

Wert	Bedeutung
<i>print-server-queue</i>	Die Umgebungsvariable SAG_APS_LPD_xyz definiert einen logischen Drucker unter Com-plete; dabei ist xyz der Name der Print Server-Warteschlange.  Wenn die Umgebungsvariable SAG_APS_LPD_xyz für das angegebene Ziel DEST existiert, wird die Ausgabe direkt an diesen Drucker (Line Printer). Weitere Informationen siehe <i>Com-plete Initialization and Startup Manual</i> , Abschnitt <i>Defining Terminals and Printers</i> .
<i>printer-file-name</i>	Wenn für diesen Drucker keine Print Server-Warteschlange zur Verfügung steht, gibt DEST einen Drucker-Dateinamen an. Er gibt die Lage der Ausgabedatei im Dateisystem an. Der Name der Ausgabedatei wird aus der Benutzerkennung und einer Folgenummer gebildet.  Da die DEST-Klausel auf maximal 8 Zeichen eingeschränkt ist, ist es nutzlos, eine Datei mit absoluter PFS-Pfadangabe zu definieren. Der in der DEST-Klausel angegebene Name bezieht sich auf das Druckdatei-Wurzelverzeichnis. Das Druckdatei-Wurzelverzeichnis wird mit der Umgebungsvariablen NAT_PRINT_ROOT angegeben.

**Beispiel:**

```
NAT_PRINT_ROOT=/nat/printer
DEST=printer1
UserId=xyz
```

Die erste Ausgabe wird in die Datei /nat/printer/printer1/xyz1 geschrieben.

Um eine Datei mit absoluter Pfaddefinition anzugeben, kann die OUTPUT-Klausel des DEFINE PRINTER-Statements benutzt werden.

## Schlüsselwort-Subparameter für AM=IMS

---

Die folgenden Schlüsselwort-Subparameter stehen für die Zugriffsmethode **AM=IMS** zur Verfügung:

**BLKSIZE** | **DRIVER**

Mögliche Werte siehe *Support of the Natural WRITE (n) Statement* im Abschnitt *Natural under IMS/TM* in der *TP Monitor Interfaces*-Dokumentation.

### **BLKSIZE - Größe des Druck-Puffers**

**BLKSIZE=***value* gibt die Größe des an den IMS/TM-Bestimmungsort gesendeten Druck-Puffers an.

### **DRIVER - Name des Natural IMS-Druck-Treibers**

**DRIVER=***value* gibt den Namen des zu verwendenden Natural IMS-Druck-Treibers an.

## Schlüsselwort-Subparameter für das DEFINE PRINTER-Statement

---

Mit den folgenden Schlüsselwort-Subparametern können Sie die Standardwerte für die gleichlautenden Optionen des DEFINE PRINTER-Statements setzen. Wenn ein Drucker geschlossen wird, werden alle Optionen des DEFINE PRINTER-Statements auf ihre Standardwerte zurückgesetzt.

**PROFILE** | **NAME** | **FORMS** | **DISP** | **COPIES** | **CLASS** | **PTY**

**PROFILE - Name der Druckersteuerzeigentabelle**

PROFILE=*value* gibt den Namen *name* der Druckersteuerzeigentabelle an (NTCCTAB-Makro).

**NAME - Listing-Name**

NAME=*value* gibt den Namen des Listings an.

**FORMS - Name der Listing-Formulare**

FORMS=*value* gibt den Namen der Listing-Formulare an.

**DISP - Listing-Disposition**

DISP=*value* gibt die Dispositionsverarbeitung (HOLD, KEEP, DELETE oder LEAVE) des Listing an.

**COPIES - Anzahl der Exemplare**

COPIES=*value* gibt die Anzahl der zu druckenden Exemplare (1 - 255) an.

**CLASS - Spool-Klasse**

CLASS=*value* gibt die Spool-Klasse (1 Byte) an.

**PRTY - Listing-Priorität**

PRTY=*value* gibt die Listing-Priorität (1 - 255) an.



# 194

## PROFILE - Dynamisches Parameterprofil anwenden

---

- PROFILE-Parameter-Syntax ..... 589

Mit diesem Natural-Profilparameter können Sie ein Parameterprofil anwenden.

Anstatt jedes Mal, wenn Sie Natural aufrufen, eine ganze Zeichenkette mit Einzelparametern anzugeben, können Sie die Parameter-Zeichenkette einmal angeben, diese Zeichenkette dann unter einem Profilnamen speichern und anschließend Natural mit diesem Parameterprofil aufrufen. Die mit diesem Profil definierten Parameter werden dann als dynamische Parameter an Natural übergeben.



**Anmerkungen:**

1. Zum Erstellen und Pflegen von Parameterprofilen dient die Utility SYSPARM.
2. Eine deutsche Beschreibung der Utility SYSPARM existiert in der Dokumentation *Debugger und Dienstprogramme*, englische Beschreibung siehe *Utilies-Dokumentation*.

Mit dem Profilparameter **FPROF** können Sie eine Systemdatei für Parameterprofile (**FPROF**) angeben.

<b>Mögliche Werte</b>	Siehe <a href="#">PROFILE-Parameter-Syntax</a> .	
<b>Standard-Einstellung</b>	keine	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	



**Anmerkungen:**

1. Falls der Profilparameter **PROFILE** im Natural-Parametermodul angegeben ist, wird er *nach* den anderen Parametern im Parametermodul ausgewertet, jedoch *bevor* irgendwelche dynamisch angegebenen Parameter ausgewertet werden. Das bedeutet, dass Parameter, die innerhalb des Profils angegeben sind, durch einzeln angegebene dynamische Parameter aufgehoben werden können.
2. Mit dem Profilparameter **USER** können Sie die Benutzung eines Profils einschränken.
3. Im Gegensatz zu anderen Parametern kann eine **PROFILE**-Parameterangabe nicht durch ein anderes Profil überschrieben werden. Somit können Sie mehrere Parameterprofile haben, die alle nacheinander ausgewertet werden.
4. Der Profilparameter **PROFILE** kann nicht mit **ADARUN MODE=SINGLE** benutzt werden.

## PROFILE-Parameter-Syntax

Der Profilparameter `PROFILE` kann gemäß einer der beiden folgenden Syntax-Optionen angegeben werden:

Syntax 1 - Standard-Datenbank/Datei:

$$\text{PROFILE} = \left\{ \begin{array}{l} \textit{profile-name} \\ \text{AUTO} \\ \text{PROGRAM} \\ \text{TERMINAL} \end{array} \right\}$$

Syntax 2 - Angegebene Datenbank/Datei:

$$\text{PROFILE} = \left( \left\{ \begin{array}{l} \textit{profile-name} \\ \text{AUTO} \\ \text{PROGRAM} \\ \text{TERMINAL} \end{array} \right\}, \textit{dbid}, \textit{fnr}, \textit{password}, \textit{cipher-key} \right)$$

Die Syntax-Elemente sind im folgenden Abschnitt beschrieben.

Eine Erläuterung der in dem Syntax-Diagramm verwendeten Symbole entnehmen Sie dem Abschnitt *Syntax-Symbole* (in der *Statements*-Dokumentation).

### Speicherort für Parameterprofile

Die Speicherung der Parameterprofile erfolgt abhängig davon, ob beim `PROFILE`-Parameter die Datenbankkennung und die Dateinummer angegeben wird. Dabei gilt Folgendes:

- Wenn die Datenbankkennung und die Dateinummer beim `PROFILE`-Parameter nicht angegeben sind (*Syntax 1*), werden die Standardeinstellungen benutzt.
  1. Wenn der Profilparameter `FPROF` gesetzt ist, wird das Parameterprofil aus der aktuellen Systemdatei `FPROF` gelesen.
  2. Wenn der Profilparameter `FPROF` nicht gesetzt ist, wird das Parameterprofil aus der aktuellen Systemdatei `FNAT` gelesen.
- Wenn die Datenbankkennung und die Dateinummer beim `PROFILE`-Parameter angegeben sind (*Syntax 2*), werden die angegebenen Werte benutzt.

### Syntax-Element-Beschreibung

**Syntax 1** verwendet Standardeinstellungen für Datenbankkennungen und Dateinummern:

- Wenn der Profilparameter **FPROF** gesetzt ist, wird das Parameterprofil aus der aktuellen Systemdatei **FPROF** gelesen.
- Wenn der Profilparameter **FPROF** nicht gesetzt ist, wird das Parameterprofil aus der aktuellen Systemdatei **FNAT** gelesen.

**Syntax 2** verwendet Datenbankkennungen und Dateinummern gemäß den Beschreibungen in der folgenden Tabelle:

Syntax-Element	Wert	Erklärung
<i>profile-name</i>	1 - 8 Zeichen	<p>Der Name des zu verwendenden Parameterprofils.</p> <p>Wenn Sie möchten, dass alle während einer Sitzung verwendeten Profile nicht aus der Standard-Datenbank und Standard-Systemdatei, sondern aus einer anderen Datenbank und Systemdatei gelesen werden, geben Sie Folgendes an, bevor Sie das erste Profil benutzen:</p> <pre>PROFILE=( , dbid, fnr)</pre>
AUTO	nicht zutreffend	<p>Natural nimmt die aktuelle TP-Benutzerkennung (enthalten in der Systemvariablen *INIT-USER) als Profilname, was bedeutet, dass das Profil benutzt wird, das unter dem Namen gespeichert ist, der dieser Kennung entspricht.</p> <p>Wird kein solches Profil gefunden, wird stattdessen (falls vorhanden) ein Profil mit dem Namen AUTO benutzt. Sie können ein solches AUTO-Profil als Standard-Profil für Benutzer definieren, die kein individuelles Profil haben.</p>
PROGRAM	nicht zutreffend	<p>Natural nimmt den Namen des Programms, das zurzeit als Natural ausgeführt wird (enthalten in der Systemvariablen *INIT-PROGRAM) als Profilname, was bedeutet, dass das Profil benutzt wird, das unter dem Namen definiert ist, der dieser Kennung entspricht.</p> <p>Wird kein solches Profil gefunden, wird stattdessen (falls vorhanden) ein Profil mit dem Namen PROGRAM benutzt. Sie können ein solches PROGRAM-Profil als Standard-Profil für Benutzer definieren, die kein individuelles Profil haben.</p>
TERMINAL	nicht zutreffend	<p>Natural nimmt die aktuelle Terminalkennung (enthalten in der Systemvariablen *INIT-ID) als Profilname, was bedeutet, dass das Profil benutzt wird, das unter dem Namen gespeichert wird, der dieser Kennung entspricht.</p> <p>Wird kein solches Profil gefunden, wird stattdessen (falls vorhanden) ein Profil mit dem Namen TERMINAL benutzt. Sie können ein solches</p>

Syntax-Element	Wert	Erklärung
		TERMINAL-Profil als Standard-Profil für Benutzer definieren, die kein individuelles Profil haben.
<i>dbid</i>	1 - 65535, außer 255	Die Kennung der Datenbank (DBID), in der sich die Natural-Systemdatei für Parameterprofile befindet.  Falls <i>dbid</i> nicht angegeben wird, dann wird der entsprechende Wert des Profilparameters DBID benutzt.
<i>fnr</i>	1 - 65535	Die Dateinummer (FNR) der Datenbank, in der sich die Natural-Systemdatei für Parameterprofile befindet.  Falls <i>fnr</i> nicht angegeben wird, dann wird der entsprechende Wert des Profilparameters FNR benutzt.
<i>password</i>	1 - 8 Zeichen	Das Adabas-Passwort, falls bei einer durch Passwort geschützten Natural-Systemdatei erforderlich.  Falls <i>password</i> nicht angegeben wird, dann wird der entsprechende Wert des Profilparameters SYSPSW benutzt.
<i>cipher-key</i>	8 Zeichen	Der Adabas-Chiffrierschlüssel, falls bei einer durch Passwort geschützten Natural-Systemdatei erforderlich.  Falls <i>cipher-key</i> nicht angegeben wird, dann wird der entsprechende Wert des Profilparameters SYSCIP benutzt.



# 195 PROGRAM - Kontrolle durch Nicht-Natural-Programm nach Session-Beendigung

---

Dieser Natural-Profilparameter gibt ein Nicht-Natural-Back-End-Programm an, das nach der Beendigung der Natural-Session die Kontrolle erhalten soll.

<b>Mögliche Werte</b>	1 - 8 Zeichen	Nicht-Natural-Back-End-Programm
	Numerischer Wert	Die Angabe eines numerischen Werts, z.B. PROGRAM=0, bedeutet „keine Back-End- Verarbeitung“.  <b>Anmerkung:</b> Das ist vor allem relevant, wenn Natural von einem Front-End-Programm aufgerufen wird, weil eine Standard-Einstellung genommen werden kann, wenn PROGRAM leer ist oder nicht angegeben wird; siehe <i>Front-End Invoked via XCTL</i> in der <i>TP Monitor Interfaces</i> -Dokumentation.
	MSG	Diese zusätzliche Option kann gesetzt werden, um anzugeben, dass die Meldung über die Session-Beendigung ausgegeben werden soll, bevor die Kontrolle an das Back-End-Programm übergeben wird. Die Syntax lautet:

		PROGRAM=( <i>program-name</i> ,MSG)
	NOMSG	Diese zusätzliche Option kann gesetzt werden, um anzugeben, dass die Meldung über die Session-Beendigung nicht ausgegeben werden soll. Die Syntax lautet:  PROGRAM=( <i>program-name</i> ,NOMSG)
<b>Standard-Einstellung</b>	PROGRAM=( <i>program-name</i> , NOMSG)	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	ja	Das Natural-Back-End-Programm kann außerdem aus einem Natural-Programm heraus gesetzt werden. Dies geschieht durch Aufrufen des Natural-Subprogramms CMPGMSET, das in der Library SYSEXTP zur Verfügung steht. Die zusätzlichen Optionen können nicht aus einem Natural-Programm heraus gesetzt werden.
<b>Anwendungsprogrammierschnittstelle</b>	USR4001N (verfügbar auf Großrechnern) USR6204N (verfügbar auf allen Plattformen)	Siehe <i>SYSEXT Utility</i> in der Dokumentation <i>Debugger und Dienstprogramme</i> .



#### Anmerkungen:

1. Die Daten für das mit dem Parameter PROGRAM angegebene Programm können mit dem TERMINATE-Statement geliefert werden.
2. Informationen zu Konventionen für das Aufrufen von Nicht-Natural-Back-End-Programmen finden Sie im Abschnitt *Back-End Program Calling Conventions* in der *Operations*-Dokumentation.
3. Die zusätzliche Option MSG wird unter IMS TM und Com-plete nicht berücksichtigt.

#### CICS-spezifische Informationen:

Zusätzlich zu Back-End-Programmen unterstützt das Natural CICS Interface Back-End-Transaktionen, die anstelle eines Programmnamens mit RET=XXXX oder RTI=XXXX oder STR=XXXX angegeben werden können, wobei XXXX eine gültige CICS-Transaktionskennung ist.

- RET=XXXX oder RTI=XXXX gibt an, dass die Kontrolle zusammen mit einer Rückkehr-Transaktionskennung durch ein CICS-Kommando RETURN TRANSID ('XXXX') an CICS übergeben werden soll.

- `RTI=XXXX` gibt an, dass die Kontrolle zusammen mit einer Rückkehr-Transaktionskennung durch ein CICS-Kommando `RETURN TRANSID ('XXXX') IMMEDIATE` an CICS übergeben werden soll.
- `STR=XXXX` gibt an, dass eine neue Transaktion durch ein CICS-Kommando `START TRANSID ('XXXX') TERMID (*INIT-ID)` gestartet werden soll, bevor die Kontrolle über ein CICS-RETURN-Kommando abgegeben wird.



#### Anmerkungen:

1. Back-End-Transaktionen werden nicht unterstützt, wenn Sie eine Natural-Session mit einem `EXEC CICS LINK`-Befehl oder einem verteilten Programm-Link (Distributed Program Link, DPL) starten. Wenn Back-End-Transaktionen angegeben werden, werden diese nicht beachtet.
2. Rückgabe-Transaktionen (`RET=` oder `RTI=`) werden nur bei terminal-orientierten Sessions unterstützt. Werden sie bei asynchronen Sessions angegeben, dann werden sie nicht beachtet.
3. Die Angabe einer Back-End-Transaktion muss vollständig in Apostrophen eingeschlossen sein.

#### Beispiele:

```
PROGRAM=MYPGM
```

```
PROGRAM=(MYPGM,MSG)
```

```
PROGRAM=(,MSG)
```

```
PROGRAM=(MYPGM)
```

```
PROGRAM='STR=CEMT'
```

```
PROGRAM=('STR=CEMT')
```

```
PROGRAM=('STR=CEMT',MSG)
```



# 196

## PS - 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.

<b>Mögliche Werte</b>	1 - 250	Maximale Anzahl der Zeilen pro Seite.
	0	Die physische Seitenlänge soll benutzt werden. <b>Anmerkung:</b> <ol style="list-style-type: none"><li>1. Wenn PS=0 für den ersten auszugebenden Report (Report 0) angegeben wird, wird die physische Seitenlänge des verwendeten Geräts minus 1 verwendet.</li><li>2. 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.</li></ol>
<b>Standard-Einstellung</b>	0	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	ja	
<b>Gültige Statements</b>	DISPLAY FORMAT INPUT SET GLOBALS WRITE	
<b>Gültiges Kommando</b>	GLOBALS	

<b>Programmierschnittstelle (API)</b>	USR1005N	Siehe <i>SYSEXT Utility</i> in der Dokumentation <i>Debugger und Dienstprogramme</i> .
---	----------	--

Siehe auch *Seitenlänge – 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.

Dieser Natural-Profilparameter steuert die Betriebsart unter CICS.



**Anmerkung:** Wenn Natural unter der Kontrolle des TP-Monitors CICS ausgeführt wird, sind zwei Betriebsarten möglich: Konversational and pseudo-konversationell.

<b>Mögliche Werte</b>	ON	<p>PSEUDO=ON aktiviert den pseudo-konversationellen Modus.</p> <p><b>Anmerkung:</b> In diesem Modus besteht eine Natural-Session aus einer Abfolge von verschiedenen Transaktionen. Nach jeder Ausgabe an ein Terminal werden alle Natural-Arbeitsbereiche gespeichert, und die Transaktion wird beendet. Wenn der Benutzer durch Drücken von ENTER (oder einer beliebigen anderen Eingabetaste) auf eine Meldung reagiert, wird eine neue Transaktion eingeleitet. Die Natural-Arbeitsbereiche werden wiederhergestellt, die Terminal-Eingabe wird gelesen, und die Natural-Session wird fortgesetzt. Die Transaktionskennung jeder nachfolgenden Transaktion kann dynamisch durch Aufrufen der Subroutine CMTRNSET gesetzt werden. Diese Subroutine befindet sich in der Library SYSEXT.</p>
	OFF	<p>PSEUDO=OFF deaktiviert den pseudo-konversationellen Modus.</p> <p><b>Anmerkung:</b></p> <ol style="list-style-type: none"> <li>1. Im konversationellen Modus besteht eine Natural-Session aus einer Transaktion, die solange aktiv ist, wie Natural aktiv ist.</li> <li>2. Hinweis zu CICS: Die Angabe PSEUDO=OFF wird bei Natural-Server-Sessions ignoriert. Siehe <i>Natural as a Server under CICS</i> in der <i>Operations</i>-Dokumentation.</li> </ol>
<b>Standard-Einstellung</b>	ON	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	



**Anmerkung:** Weitere Informationen zu Natural unter CICS siehe *TYPE - Thread Type for Group* in der *TP Monitor Interfaces*-Dokumentation.

# 198

## RCA - Adressen von statischen

### Nicht-Natural-Programmen auflösen

---

Dieser Natural-Profilparameter steuert das *dynamische* Linken von statischen Nicht-Natural-Programmen während der Initialisierung einer Natural-Session.

<b>Mögliche Werte</b>	ON	Beim Natural-Start wird die Liste aller zu verlinkenden statischen Nicht-Natural-Programme gelesen, und es wird für alle Module, deren Adressen nicht aufgelöst sind, eine Ladeanforderung abgesetzt. Wenn eine Ladeanforderung erfolglos ist, wird keine Fehlermeldung ausgegeben.  <b>Anmerkung:</b> Die Verwendung von RCA=ON wird <i>nicht</i> empfohlen, weil sie einen hohen zusätzlichen Verarbeitungsaufwand beim Natural-Start verursacht.
	OFF	Es erfolgt keine dynamische Verlinkung von statischen Nicht-Natural-Programmen.
	Modulname oder Liste mit Modulnamen (siehe <i>Beispiele</i> weiter unten)	Wenn Sie einen Modulnamen oder Liste mit Modulnamen angeben, dann wird die Liste der statischen Nicht-Natural-Programme, die zu Natural gelinkt werden sollen, um den bzw. die angegebenen Namen erweitert. Eine Ladeanforderung für diese Module wird auch dann abgesetzt, wenn sie bereits mit dem Natural-Nucleus verlinkt sind. So ist es möglich, Module zu ersetzen, die schon statisch mit dem Natural-Nucleus verlinkt sind. Falls eine Ladeanforderung erfolglos ist, wird eine Fehlermeldung ausgegeben.
<b>Standard-Einstellung</b>	OFF	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	



#### Anmerkungen:

1. Benutzen Sie den Profilparameter `CSTATIC`, um Module als statisch mit dem Natural-Nucleus verlinkt zu definieren.
2. Falls der externe Name des zu ladenden Moduls anders lautet als der interne (vom `CALL`-Statement benutzte) Name, können Sie entweder den Profilparameter `RCALIAS` oder das Makro `NTALIAS` benutzen, um festzulegen, welcher externe Name für die Ladeanforderung verwendet werden soll.
3. Unter CICS: Um die Ladeanforderung für ein dynamisch zu ladendes Modul zuzulassen, muss ein PPT-Eintrag definiert werden. Statische verlinkte Module werden eher über Standard-Linkage-Konventionen als über `EXEC CICS LINK`-Anforderungen aufgerufen.
4. Als Voraussetzung für Attached Debugging muss der Profilparameter auf `RCA=NATATDBG` gesetzt sein. Siehe auch *Natural-Debugger*-Dokumentation.

### Beispiele

Im Falle eines einzelnen Modulnamens können Sie Folgendes angeben:

```
RCA=module-name
```

Im Falle einer Liste mit Modulnamen müssen die Namen jeweils durch ein Komma voneinander abgetrennt werden, und die gesamte Liste muss in Klammern gesetzt werden:

```
RCA=(module-name1, module-name2, module-name3)
```

# 199

## RCALIAS - Externe Namen für Nicht-Natural-Programme

---

■ RCALIAS-Parameter-Syntax .....	604
■ NTALIAS-Makro-Syntax .....	605
■ Beispiel für RCALIAS-Parameter .....	605
■ Beispiele für NTALIAS-Makro .....	605

Dieser Natural-Profilparameter kann verwendet werden, um externe Namen für statische Nicht-Natural-Programme zu definieren, die mit dem Profilparameter *RCA* definiert und zum dynamischen Linken bei der Initialisierung einer Natural-Session geladen werden. *RCALIAS* entspricht dem Makro *NTALIAS* im Natural-Parametermodul.

<b>Mögliche Werte</b>	Siehe <i>RCALIAS-Parameter-Syntax</i> .	
<b>Standard-Einstellung</b>	OFF	Siehe <i>RCALIAS-Parameter-Syntax</i> .
<b>Dynamische Angabe</b>	ja	Dieser Parameter kann nur dynamisch angegeben werden. Im Natural-Parametermodul wird stattdessen das Makro <i>NTALIAS</i> benutzt.
<b>Angabe innerhalb der Session</b>	nein	

## RCALIAS-Parameter-Syntax

Der Parameter *RCALIAS* wird wie folgt angegeben:

```
RCALIAS=(internal-module-name,external-module-name,internal-module-name,external-module-name)
```

Oder:

```
RCALIAS=OFF
```

Dabei ist

Syntax-Element	Wert	Erklärung
<i>internal-module-name</i>	1 - 12 Zeichen.	Der interne Name des Moduls, das vom CALL-Statement verwendet wird und das außerdem mit dem Parameter <i>RCA</i> oder <i>CSTATIC</i> (nur wenn <i>RCA=ON</i> ) definiert werden muss.
<i>external-module-name</i>	1 - 12 Zeichen.	Der entsprechende externe Alias-Name für die Ladeanforderung während der Session-Initialisierung.
-	OFF	Es sind keine externen Namen für Module definiert, die mit dem Profilparameter <i>RCA</i> als statisch mit dem Natural-Nukleus verlinkte Module definiert sind.

 **Anmerkungen:**

- Mit dem Profilparameter *RCALIAS* können mehrere Namen-Paare gleichzeitig angegeben werden; siehe *Beispiel für RCALIAS-Parameter*.
- Als Voraussetzung für Attached Debugging muss der Profilparameter *RCALIAS* auf (NATATDBG,NATAD*vrs*) gesetzt sein, für CICS auf z/OS auf (NATATDBG,NCIAD*vrs*).

## NTALIAS-Makro-Syntax

Das Makro NTALIAS wird wie folgt angegeben:

```
NTALIAS internal-module-name,external-module-name
```

Dabei ist:

Syntax-Element	Wert	Erklärung
<i>internal-module-name</i>	1 - 12 Zeichen.	Der interne Name des Moduls, das vom CALL-Statement verwendet wird und das außerdem mit dem Parameter <b>RCA</b> oder <b>CSTATIC</b> (nur wenn RCA=ON) definiert werden muss.
<i>external-module-name</i>	1 - 12 Zeichen.	Der entsprechende externe Alias-Name für die Ladeanforderung während der Session-Initialisierung.



### Anmerkungen:

1. Für jedes Namen-Paar muss ein eigenes NTALIAS-Makro angegeben werden; siehe [Beispiele für NTALIAS-Makros](#).
2. Der Wert OFF kann beim Makro NTALIAS nicht angegeben werden, sondern nur dynamisch mit dem Profilparameter RCALIAS.

## Beispiel für RCALIAS-Parameter

```
RCA=(PROGRAM1,PROGRAM2),RCALIAS=(PROGRAM1,ALIAS1,PROGRAM2,ALIAS2)
```

## Beispiele für NTALIAS-Makro

```
NTALIAS PROGRAM1,ALIAS1
NTALIAS PROGRAM2,ALIAS2
```



# 200 RCFIND - Maßnahme bei Adabas Response Code 113 beim FIND-Statement

---

Dieser Natural-Profilparameter gibt an, welche Maßnahme ergriffen werden soll, wenn der Adabas Response Code 113 (angeforderte ISN nicht gefunden) während der Ausführung einer FIND-Statement-Verarbeitungsschleife zurückgegeben wird.

<b>Mögliche Werte</b>	ON	Response Code 113 bewirkt, dass das Programm beendet wird.
	OFF	Response Code 113 wird nicht beachtet und die Verarbeitung der FIND-Statement-Verarbeitungsschleife wird dadurch fortgesetzt, dass der nächste Datensatz gelesen wird.
<b>Standard-Einstellung</b>	ON	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	



# 201 RCGET - Maßnahme bei Adabas Response Code 113 beim GET-Statement

---

Dieser Natural-Profilparameter gibt an, welche Maßnahme ergriffen werden soll, wenn der Adabas Response Code 113 (angeforderte ISN nicht gefunden) während der Ausführung eines GET-Statements zurückgegeben wird.

<b>Mögliche Werte</b>	ON	Response Code 113 bewirkt, dass das Programm beendet wird.
	OFF	Response Code 113 wird nicht beachtet, die Systemvariable *ISN wird auf 0 gesetzt und die Verarbeitung wird fortgesetzt.
<b>Standard-Einstellung</b>	ON	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	



# 202 RDACT - (für interne Verwendung)

---

Dieser Natural-Profilparameter ist für die interne Verwendung durch Natural reserviert.



**Vorsicht:** Ändern Sie seine Einstellung nicht.



# 203

## RDC - Konfiguration des Natural Data Collector

---

- RDC-Parameter-Syntax ..... 614
- NTRDC-Makro-Syntax ..... 615
- Schlüsselwort-Subparameter ..... 615
- Beispiel für RDC-Parameter ..... 618
- Beispiel für NTRDC-Makro ..... 618

Dieser Natural-Profilparameter wird benutzt, um den Natural Data Collector und seine Trace Recording-Funktion zu konfigurieren. Diese Funktion wird von der *SYSRDC Utility* und von der *PROFILER-Utility* benutzt. Der Profilparameter RDC entspricht dem Makro **NTRDC** im Natural-Parametermodul.

<b>Mögliche Werte</b>	Siehe <a href="#">RDC-Parameter-Syntax</a> .	
<b>Standard-Einstellung</b>	OFF	Siehe <a href="#">RDC-Parameter-Syntax</a> und <a href="#">Keyword Subparameters</a> .
<b>Dynamische Angabe</b>	ja	Der Parameter RDC kann nur dynamisch angegeben werden. Im Natural-Parametermodul wird das Makro <b>NTRDC</b> verwendet..
<b>Angabe innerhalb der Session</b>	ja	Um die Trace Recording-Funktion zu starten oder zu stoppen, siehe <i>Calling the CMRDC Interface</i> (siehe auch Subparameter <a href="#">EVENT</a> ).

Dieses Dokument behandelt folgende Themen:

## RDC-Parameter-Syntax

Der Profilparameter RDC wird wie folgt angegeben:

```
RDC=(ON,keyword-subparameter=value,keyword-subparameter=value,...)
```

Oder:

```
RDC=OFF
```

Dabei ist:

Wert	Bedeutung
ON	Die Trace Recording-Funktion wird beim Start der Natural-Session automatisch gestartet.
OFF	Der Benutzer muss die Trace Recording-Funktion mittels einer der Methoden starten, die im Abschnitt <i>Natural Data Collector aktivieren und steuern</i> in der <i>SYSRDC Utility</i> -Dokumentation beschrieben sind.  Dies ist der Standardwert.
<i>keyword-subparameter</i>	Siehe <a href="#">Schlüsselwort-Subparameter</a> .



**Anmerkung:** Der Wert ON bzw. OFF muss, falls angegeben, immer an erster Stelle stehen.

## NTRDC-Makro-Syntax

Das Makro NTRDC wird wie folgt angegeben:

```
NTRDC ON, *
      EVENT=value, *
      EXIT=value, *
      FNAT=value, *
      SIZE=value, *
      XEVENT=value
```

Oder:

```
NTRDC OFF
```

Siehe [Schlüsselwort-Subparameter](#).

Beschreibung der Werte ON und OFF siehe [RDC-Parameter-Syntax](#).

## Schlüsselwort-Subparameter

[EVENT](#) | [EXIT](#) | [FNAT](#) | [SIZE](#) | [XEVENT](#)

### EVENT - Natural Data Collector-Ereignisse, die aufgezeichnet werden sollen

EVENT=(*value*) bestimmt die Ereignisse, die im Natural Data Collector-Puffer aufgezeichnet werden sollen.

Wert	Erklärung
( <i>event</i> , <i>event</i> ,...)	<p>Jedes Ereignis (<i>event</i>) kann ein oder zwei Zeichen lang sein. Nur die angegebenen Ereignisse werden aufgezeichnet.</p> <p><b>Anmerkung:</b></p> <ol style="list-style-type: none"> <li>1. Informationen zu den Ereignistypen siehe <i>Datenerfassung auslösende Ereignisse</i> in der <i>SYSRDC Utility</i>-Dokumentation. Wenn Sie nur ein Ereignis angeben, können Sie die Klammern weglassen.</li> <li>2. Wenn nur ein Zeichen angegeben wird und mehrere Ereignisse existieren, die mit diesem Zeichen beginnen, werden alle diese Ereignisse aufgezeichnet. Beispiel: EVENT=P bedeutet, dass die Ereignisse PL, PS und PT aufgezeichnet werden; das heißt, die Definition ist gleichbedeutend mit EVENT=(PL, PS, PT).</li> </ol>

Wert	Erklärung
ALL	Alle Ereignisse werden aufgezeichnet. Voraussetzung ist, dass der Natural Data Collector-Puffer ausreichend groß definiert worden ist.  Dies ist der Standardwert.

**Anmerkungen:**

1. Diese Definition kann in der PROFILER Utility oder in jedem beliebigen Natural-Programm durch das Statement `CALL 'CMRDC' 'T'` ersetzt werden.
2. Weitere Informationen siehe *PROFILER Utility* bzw. *Aufruf des CMRDC-Interface* in der *SYSRDC Utility-Dokumentation*.

**EXIT - User Exits für den Natural Data Collector**

EXIT=(*value*) wird benutzt, um User Exits für den Natural Data Collector der Utility SYSRDC zu definieren. Optional kann auch eine Arbeitsbereichgröße für jeden Exit angegeben werden.

Wert	Erklärung
( <i>name,name,...</i> ) oder ( <i>name,size,name,size,...</i> )	<i>name</i> ist der Name des User Exit. Im Natural-Parametermodul werden die angegebenen Exit-Namen automatisch an die Liste der Angaben im Profilparameter <code>CSTATIC</code> angehängt.  Es gibt keinen Standardwert.  Optional kann nach jedem Exit-Namen die Größe ( <i>size</i> ) des Arbeitsbereichs angegeben werden:  Mögliche Arbeitsbereichsgrößen: 400 - 32760.  Standardeinstellung: 400

**Anmerkungen:**

1. Wenn er verlinkt wird, erhält der Exit die Kontrolle vom Natural Data Collector an bestimmten Stellen innerhalb von Natural. Dann werden spezifische Session-Informationen an die Exits übergeben.
2. Wenn der Subparameter EXIT dynamisch angegeben wird, müssen die Exits im Profilparameter `CSTATIC` oder `RCA` definiert werden (`RCA` kann ebenfalls dynamisch angegeben werden).
3. Als Alternative zu diesem Subparameter können Sie den Natural-Profilparameter `RDCEXIT` benutzen.
4. Einzelheiten siehe *User Exits für externes Monitoring/Accounting* in der *SYSRDC Utility-Dokumentation*.

**Beispiel:**

```
EXIT=(MYEXIT,2000,RDCEX1) ↵
```

**FNAT - Trace Recording in der Natural-Systemdatei**

FNAT=*value* steuert die Trace-Recording-Funktion, während Programme in der Natural-Systemdatei ausgeführt werden.

Wert	Erklärung
ON	Die Trace-Recording-Funktion ist aktiv, während Programme in der Natural-Systemdatei ausgeführt werden.
OFF	Kein Trace Recording in der Natural-Systemdatei. Dies ist der Standardwert.

**SIZE - Größe des Natural Data Collector-Puffers**

SIZE=*value* gibt die Größe des Natural Data Collector-Puffers an, der von der *SYSRDC Utility* und der *PROFILER-Utility* genutzt wird. Außerdem steuert er die Trace Recording-Funktion des Natural Data Collector.

Wert	Erklärung
0 or 2 - 128	<p>Puffergröße (in KB).</p> <ul style="list-style-type: none"> <li>■ Um den Natural Data Collector (ohne Trace-Aufzeichnung) zu aktivieren, geben SIZE=2 an.</li> <li>■ Um außerdem die Trace-Aufzeichnung zu aktivieren, müssen Sie SIZE auf einen Wert größer als 2 setzen.</li> <li>■ Um die Trace Recording-Funktion zu aktivieren, benötigt die PROFILER-Utility eine Puffergröße von mehr als 2.</li> <li>■ Falls der angeforderte Speicher nicht verfügbar ist, kann der Natural Data Collector nicht benutzt werden.</li> </ul>
0	<p>Deaktiviert den Natural Data Collector.</p> <p>Dies ist der Standardwert.</p>

**Anmerkungen:**

1. Als Alternative zu diesem Subparameter können Sie den äquivalenten Natural-Profilparameter [RDCSIZE](#) benutzen.
2. Weitere Informationen siehe *SYSRDC Utility*-Dokumentation.

## XEVENT - Natural Data Collector-Ereignisse, bei denen User Exits aufgerufen werden

XEVENT=(value) bestimmt Natural Data Collector-Ereignisse, bei denen User Exits aufgerufen werden. Dies kann dazu beitragen, die CPU-Inanspruchnahme und die Anzahl der TCB/SRB-Schaltvorgänge zu reduzieren, wenn ein Natural for zIIP Add-On-Produkt installiert ist.

Wert	Erklärung
(event, event, ...)	<p>Jedes Ereignis (event) kann ein Ereignistyp sein, der ein oder zwei Zeichen umfasst. User Exits werden nur bei den angegebenen Ereignissen aufgerufen werden.</p> <p><b>Anmerkung:</b></p> <ol style="list-style-type: none"> <li>1. Informationen zu möglichen Ereignistypen siehe <i>Datenerfassung auslösende Ereignisse</i> in der <i>SYSRDC Utility</i>-Dokumentation.</li> <li>2. Wenn nur ein Wert angegeben wird, können die umschließenden Klammern weggelassen werden.</li> <li>3. Wenn nur ein Buchstabe angegeben wird, aber mehrere Ereignisse mit diesem Buchstaben beginnen, werden alle diese Ereignisse aufgezeichnet. Beispiel: XEVENT=P bedeutet, dass die Ereignisse PL, PS und PT aufgezeichnet werden; das heißt, diese Definition ist gleichbedeutend mit XEVENT=(PL, PS, PT).</li> </ol>
ALL	<p>Alle User-Exits werden bei allen Ereignissen aufgerufen.</p> <p>Dies ist der Standardwert.</p>

## Beispiel für RDC-Parameter

```
RDC=(ON,EVENT=(D,P,U),XEVENT=(P,I,A),SIZE=80, FNAT=ON)
```

```
RDC=(ON,EVENT=(PS),SIZE=80, FNAT=ON,EXIT=(MYEXIT,2000,RDCEX1))
```

## Beispiel für NTRDC-Makro

```
NTRDC ON, *
      EVENT=(D,P,U), *
      XEVENT=(P,I,A), *
      SIZE=80, *
      FNAT=ON, *
      EXIT=(MYEXIT,2000,RDCEX1)
```

Dieser Natural-Profilparameter wird benutzt, um User Exits für den Natural Data Collector der *SYSRDC Utility* zu definieren. Wahlweise kann auch eine Arbeitsbereichgröße für jeden Exit angegeben werden.

<b>Mögliche Werte</b>	Siehe <a href="#">RDCEXIT-Parameter-Syntax</a> .	
<b>Standard-Einstellung</b>	keine	
<b>Dynamische Angabe</b>	ja	<b>Anmerkung:</b> Wenn RDCEXIT dynamisch angegeben wird, müssen die Exits im Profilparameter CSTATIC oder RCA definiert werden (RCA kann auch dynamisch angegeben werden).
<b>Angabe innerhalb der Session</b>	nein	



#### Anmerkungen:

1. Alternativ können Sie den gleichbedeutenden Subparameter EXIT des Profilparameters RDC oder des Makros NTRDC benutzen.
2. Wenn er verlinkt wird, erhält der Exit die Kontrolle vom Natural Data Collector an bestimmten Stellen innerhalb von Natural. Dann werden spezifische Session-Informationen an die Exits übergeben. Weitere Informationen siehe *SYSRDC Utility*-Dokumentation.

#### RDCEXIT-Parameter-Syntax

```
RDCEXIT=(name,name,...)
```

Dabei ist *name* der Name des User Exit. Im Natural-Parametermodul werden die Exit-Namen automatisch zur Liste der Einstellungen des Profilparameters CSTATIC hinzugefügt.

Es gibt keinen Standardwert.

Oder:

Optional kann nach jedem Exit-Namen die Größe (*size*) des Exit-Arbeitsbereichs angegeben werden.

```
RDCEXIT=(name,size,name,size,...)
```

Mögliche *size*-Werte: 400 - 32760.

Der Standardwert für *size* ist 400.

**Beispiel:**

```
RDCEXIT=(MYEXIT,2000,RDCEX1)
```

# 205

## RDCSIZE - Puffergröße für den Natural Data Collector

Dieser Natural-Profilparameter gibt die Puffergröße für den Natural Data Collector an, der von der *SYSRDC Utility* und von der *PROFILER-Utility* genutzt wird. Außerdem steuert er die Trace Recording-Funktion des Natural Data Collector.

<b>Mögliche Werte</b>	2 - 128	Puffergröße (in KB). <ul style="list-style-type: none"><li>■ Um den Natural Data Collector (ohne Trace-Aufzeichnung) zu aktivieren, geben Sie RDCSIZE=2 an.</li><li>■ Um außerdem die Trace-Aufzeichnung zu aktivieren, müssen Sie RDCSIZE auf einen Wert größer als 2 setzen.</li><li>■ Um die Trace Recording-Funktion zu aktivieren, benötigt die Profiler-Utility eine Puffergröße von mehr als 2.</li><li>■ Die Trace Recording-Funktion wird nur dann automatisch während der Session-Initialisierung aktiviert, wenn der Profilparameter RDC oder das Makro NTRDC auf ON gesetzt ist.</li><li>■ Falls der angeforderte Speicher nicht verfügbar ist, kann der Natural Data Collector nicht benutzt werden.</li></ul>
	0	Deaktiviert den Natural Data Collector.
<b>Standard-Einstellung</b>	0	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	



### Anmerkungen:

1. Als Alternative können Sie den gleichbedeutenden Subparameter SIZE des Profilparameters RDC oder des Makros NTRDC benutzen.

2. Alternativ können Sie den Wert für RDCSIZE auch im Natural-Profilparameter [DS](#) oder das Parametermakro [NTDS](#) angeben.
3. Weitere Informationen siehe *Natural Data Collector aktivieren und steuern* in der *SYSRDC Utility*-Dokumentation.

# 206

## RDNODE (für interne Verwendung)

---

Dieser Parameter ist für die interne Nutzung durch Natural reserviert.



**Vorsicht:** Ändern Sie seine Einstellung nicht.



# 207 RDPOR (für interne Verwendung)

---

Dieser Parameter ist für die interne Nutzung durch Natural reserviert.



**Vorsicht:** Ändern Sie seine Einstellung nicht.



Dieser Natural-Profilparameter gibt an, welche Maßnahme durchgeführt werden soll, wenn Natural eine Inkonsistenz in der Definition der GDA (Global Data Area) feststellt, die im dem zurzeit ausgeführten Programm angegeben ist; d.h. die im Programm angegebene GDA entspricht nicht der Definition der zurzeit in Benutzung befindlichen GDA.

<b>Mögliche Werte</b>	ON	Bei Feststellen einer Inkonsistenz in einem Programm und/oder in einer GDA wird eine Fehlermeldung ausgegeben. <ul style="list-style-type: none"> <li>■ Falls das korrigierte Objekt ein Objekt aus einer Steplib aufruft, das ebenfalls korrigiert werden muss, wird das Objekt aus der Steplib Library in die Library des aufrufenden Objekts kopiert.</li> <li>■ Natural korrigiert das Objekt automatisch und schaltet die Funktionen der Systemkommandos CATALOG und SAVE aus.</li> <li>■ RECAT=ON ist nicht möglich bei Programmen, die mit dem Natural Optimizer Compiler katalogisiert wurden.</li> </ul>
	OFF	Natural gibt eine Fehlermeldung aus.
<b>Standard-Einstellung</b>	OFF	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	
<b>Anwendungsprogrammierschnittstelle</b>	USR1005N	Siehe <i>SYSEXT Utility</i> in der <i>Debugger und Dienstprogramme</i> -Dokumentation.



# 209

## REINP - Interner REINPUT bei ungültigen Daten

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

<b>Mögliche Werte</b>	ON	Bei Eingabe ungültiger Daten wird ein internes REINPUT-Statement ausgeführt.	
	OFF	Bei Eingabe ungültiger Daten wird kein internes REINPUT-Statement ausgeführt.	
<b>Standard-Einstellung</b>	ON		
<b>Dynamische Angabe</b>	ja		
<b>Angabe innerhalb der Session</b>	ja	Gültige Statements:	SET GLOBALS
		Gültiges Kommando:	GLOBALS
<b>Programmierschnittstelle (API)</b>	USR1005N	Siehe <i>SYSEXT Utility</i> in der <i>Debugger und Dienstprogramme</i> -Dokumentation.	



### Anmerkungen:

1. 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.
2. In einer Natural-Session kann der Profilparameter REINP durch den Session-Parameter REINP überschrieben werden.



# 210

## RELO - Speicher-Thread-Verschiebung

Dieser Natural-Profilparameter steuert die Verschiebung eines Natural-Thread nach einer Terminal-Ein-/Ausgabe in einer Thread-Umgebung (CICS, Com-plete, IMS/TM und Natural als Server).

<b>Mögliche Werte</b>	ON	Der Natural-Thread und alle darin enthaltenen Puffer können in einen anderen Speicherbereich verschoben werden, wenn der ursprüngliche Speicherbereich nach einer Terminal-Ein-/Ausgabe durch einen anderen Benutzer belegt worden ist.
	OFF	Es erfolgt keine Verschiebung. Der Natural-Thread und alle darin enthaltenen Puffer verbleiben nach einer Terminal-Ein-/Ausgabe an derselben virtuellen Adresse.  <b>Anmerkung:</b> Diese Einstellung gilt nur für CICS, Com-plete und Server-Umgebungen. In allen anderen Thread-Umgebungen kann Natural nicht garantieren, dass der Thread an derselben Adresse bleibt.
	FORCE	Erzwingt eine Verschiebung des Natural-Thread und aller darin enthaltenen Puffer in einen anderen Speicherbereich. Dies kann in einigen Umgebungen für Testzwecke nützlich sein.
<b>Standard-Einstellung</b>	ON	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	

### Hinweise für CICS:

- Bei Verwendung von Threads mit TYPE=GETM unter CICS hat die Einstellung RELO=OFF dieselbe Wirkung wie die Einstellung PSEUDO=OFF des Profilparameters PSEUDO. Siehe auch TYPE (Thread-Typ für Gruppe) im Abschnitt *Natural under CICS* in der *TP Monitor Interfaces*-Dokumentation.
- Die Angabe RELO=OFF wird bei Natural-Server-Sessions unter CICS, die Threads mit TYPE=GETM verwenden, ignoriert; siehe *Natural as a Server under CICS* in der *TP Monitor Interfaces*-Dokumentation.



# 211

## RFILE - Datei für Aufzeichnung per Recording-Funktion

---

Dieser Natural-Profilparameter gibt an, wo Aufzeichnungen (d.h. Datenaufzeichnung mittels der Recording-Funktion) gespeichert werden.

<b>Mögliche Werte</b>	SPAD	Die Aufzeichnungen werden im Scratch-Pad File gespeichert. (Wenn kein Scratch-Pad File definiert ist, werden die Aufzeichnungen in der Systemdatei FUSER gespeichert.)  Siehe auch <i>Natural Scratch-Pad File</i> in der <i>Operations</i> -Dokumentation.
	FUSER	Die Aufzeichnungen werden in der Systemdatei FUSER gespeichert.
	FNAT	Die Aufzeichnungen werden in der Systemdatei FNAT gespeichert.
<b>Standard-Einstellung</b>	SPAD	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	



**Anmerkung:** Informationen zur Recording-Funktion siehe *Recording Utility* in der *Debugger und Dienstprogramme*-Dokumentation.



# 212

## RI - Freigabe von Adabas ISNs

Dieser Natural-Profilparameter gibt an, ob ISNs (Internal Sequence Numbers) für Datensätze, die gelesen und in den Hold-Status versetzt, aber nicht aktualisiert wurden, im Hold-Status beibehalten werden sollen.

<b>Mögliche Werte</b>	ON	Natural gibt die ISN jedes Datensatzes frei, der in den Hold-Status versetzt, aber nicht aktualisiert wurde (z.B. weil der Datensatz als Ergebnis der Auswertung einer WHERE-Klausel oder eines ACCEPT/REJECT-Statements zurückgewiesen wurde).  <b>Anmerkung:</b> Das kann jedoch zusätzlichen Performance-Aufwand verursachen, weil für jede freigegebene ISN ein Adabas-Aufruf erforderlich ist.
	OFF	Die ISN der einzelnen Datensätze, die in den Hold-Status versetzt wurden, wird <i>nicht</i> vor dem Ende der Transaktion freigegeben.
<b>Standard-Einstellung</b>	OFF	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	
<b>Anwendungsprogrammierschnittstelle</b>	USR1005N	Siehe <i>SYSEXT Utility</i> in der <i>Debugger und Dienstprogramme</i> -Dokumentation.



**Anmerkung:** In verschachtelten Verarbeitungsschleifen wird ein Datensatz, der wegen RI=ON in einer inneren Verarbeitungsschleife freigegeben wird, auch für keine der äußeren Verarbeitungsschleifen weiter im Hold-Status gehalten.



# 213

## RJESIZE - Anfangsgröße des NATRJE-Puffers

Dieser Natural-Profilparameter gibt die Anfangsgröße des NATRJE-Puffers an.

<b>Mögliche Werte</b>	1 - 2097151	Puffergröße in KB. <b>Anmerkung:</b> Falls die Anfangsgröße nicht ausreicht, erhöht Natural die Puffergröße in Schritten von 8 KB (automatisch und, falls erforderlich, mehrmals).
	0	Schaltet die NATRJE-Utility aus.
<b>Standard-Einstellung</b>	8	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	



### Anmerkungen:

1. Mit der Natural-Utility NATRJE (siehe in der Dokumentation *Debugger und Dienstprogramme*) können JCL-Jobs gesammelt und dann alle auf einmal gestartet werden. Der Profilparameter RJESIZE wird verwendet, um die Anfangsgröße des NATRJE-Puffers anzugeben, der die JCL-Jobs enthält, bevor sie gestartet werden.
2. Alternativ können Sie den äquivalenten Natural-Profilparameter [DS](#) oder das Parametermakro [NTDS](#) benutzen.



# 214

## RM - Rückübertragung von geänderten Feldern

---

Dieser Natural-Profilparameter steuert die Rückübertragung von geänderten Feldern.

<b>Mögliche Werte</b>	ON	Natural sendet immer alle geänderten Felder zurück.
	OFF	Natural sendet geänderte Felder nur dann zurück, wenn sie geändert worden sind.
<b>Standard-Einstellung</b>	OFF	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	



### Anmerkungen:

1. Manche TP-Monitore setzten die Eingabedaten automatisch in Zeichen mit Großbuchstaben um.
2. Da die in Natural eingesetzte Bildschirm-Optimierung nur geänderte Daten an den Bildschirm zurück überträgt, kann die Umsetzung durch den TP-Monitor Eingaben für ein geändertes Feld verursachen, die nicht zurück übertragen werden sollen.



# 215

## RNCONST - Zeilennummern in Konstanten

### umnummerieren

---

Mit diesem Natural-Profilparameter können Sie festlegen, ob die Zeilennummernreferenzen in alphanumerischen Konstanten innerhalb des Natural-Quellcodes umnummeriert werden.

Siehe auch *Quellcode-Zeilennummern umnummerieren im Leitfaden zur Programmierung*.

<b>Mögliche Werte</b>	ON	Die Zeilennummernreferenzen in alphanumerischen Konstanten im Natural-Quellcode werden umnummeriert.
	OFF	Die Zeilennummernreferenzen in alphanumerischen Konstanten im Natural-Quellcode werden nicht umnummeriert. Sie bleiben unverändert.
<b>Standard-Einstellung</b>	OFF	
<b>Dynamische Angabe</b>	nein	
<b>Angabe innerhalb der Session</b>	nein	
<b>Gültige Statements</b>	keine	
<b>Gültiges Kommando</b>	RENUMBER	



**Anmerkung:** Die Einstellung des Profilparameters RNCONST beeinflusst das Ausführungsverhalten des Systemkommandos RENUMBER.

---

Dieser Natural-Profilparameter unterbindet Änderungen an den Natural-Systemdateien [FNAT](#), [FUSER](#), [FDIC](#) und [FSEC](#).

<b>Mögliche Werte</b>	ON	Es können keine Daten in die Systemdateien geschrieben, dort geändert oder gelöscht werden. Statt eine datenverändernde Aktion in diesen Systemdateien auszuführen, gibt Natural eine Fehlermeldung aus.
	OFF	Daten können in die Systemdateien geschrieben, dort geändert oder gelöscht werden.
<b>Standard-Einstellung</b>	OFF	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	



#### Anmerkungen:

1. Wenn Ihre Systemdateien mit `ROSY=ON` schreibgeschützt sind, können die Natural Recording Utility und die `NATPAGE` Utility nicht benutzt werden, weil sie Daten in die Systemdatei(en) `FNAT` und/oder `FUSER` schreiben.
2. Deshalb empfehlen wir Ihnen, für die Speicherung dieser temporären Daten ein so genanntes Scratch-Pad File zuzuweisen und zu verwenden. Das Scratch-Pad-File ist optional und muss mit dem Makro `NTLFILE` bzw. dem Profilparameter `LFILE` als wiederherstellbar definiert werden. Die genannten Funktionen schreiben dann ihre Daten in diese Datei statt in `FNAT/FUSER`. Siehe auch *Natural Scratch-Pad File* in der *Operations*-Dokumentation.
3. Wenn `ROSY=OFF` gesetzt ist, sollte ein Scratch-Pad-File auch dann definiert werden, wenn Sie die Natural Recording Utility und die `NATPAGE` Utility mit Datenbank-Transaktionslogik benutzen, weil dies bei `FNAT/FUSER` zu unvorhersehbaren Ergebnissen führen kann.
4. Wenn eine Systemdatei in einem der entsprechenden Profilparameter `FNAT`, `FUSER` oder `FSEC` mit der `RO`-Option als schreibgeschützt angegeben ist, ist es nicht möglich, diese Einstellung

durch durch Setzen von `ROSY=OFF` zu unterdrücken, um Änderungen/Aktualisierungen in der betreffenden, schreibgeschützten Systemdatei zu ermöglichen.

# 217

## RPC - Remote Procedure Call-Einstellungen

---

▪ RPC-Parameter-Syntax .....	646
▪ NTRPC-Macro-Syntax .....	646
▪ Schlüsselwort-Subparameter für Client und Server .....	647
▪ Schlüsselwort-Subparameter nur für Client .....	651
▪ Schlüsselwort-Subparameter nur für Server .....	656
▪ Beispiel für RPC-Parameter .....	665
▪ Beispiel für NTRPC-Makro .....	665

Dieser Natural-Profilparameter dient dazu, die Einstellungen zur Steuerung des Betriebs des Natural RPC (Remote Procedure Call) zu setzen. Er entspricht dem Makro `NTRPC` im Natural-Parametermodul.

<b>Mögliche Werte</b>	Siehe <a href="#">RPC-Parameter-Syntax</a> .	
<b>Standard-Einstellung</b>	keine	Siehe: <ul style="list-style-type: none"> <li>■ <a href="#">Schlüsselwort-Subparameter für Client und Server</a></li> <li>■ <a href="#">Schlüsselwort-Subparameter nur für Client</a></li> <li>■ <a href="#">Schlüsselwort-Subparameter nur für Server</a></li> </ul>
<b>Dynamische Angabe</b>	ja	Dieser Parameter kann nur dynamisch angegeben werden. Im Natural-Parametermodul wird stattdessen das Makro <code>NTRPC</code> verwendet.
<b>Angabe innerhalb der Session</b>	nein	

Dieses Dokument behandelt folgende Themen:



**Anmerkung:** Siehe auch *Setting Up a Natural RPC Environment* in der *Natural RPC (Remote Procedure Call)*-Dokumentation.

## RPC-Parameter-Syntax

---

Der Parameter `RPC` hat folgende Syntax:

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

Namen und Werte für `keyword-subparameter` siehe [Schlüsselwort-Subparameter für Client und Server](#), [Schlüsselwort-Subparameter nur für Client](#) und [Schlüsselwort-Subparameter nur für Server](#).

## NTRPC-Macro-Syntax

---

Auf der Client-Seite hat das Makro `NTRPC` im Natural-Parametermodul folgende Syntax:

```

NTRPC ACIVERS=value, *
      AUTORPC=value, *
      COMPR=value, *
      CPRPC=value, *
      DFS=value, *
      MAXBUFF=value, *
      RDS=value, *
      RPCSDIR=value, *
      RPCSIZE=value, *
      SERVER=value, *
      TIMEOUT=value, *
      TRYALT=value

```

Auf der Server-Seite hat das Makro NTRPC im Natural-Parametermodul folgende Syntax:

```

NTRPC ACIVERS=value, *
      CPRPC=value, *
      LOGONRQ=value, *
      MAXBUFF=value, *
      NTASKS=value, *
      RPCSIZE=value, *
      RPCUCT=value, *
      SERVER=value, *
      SRVCMIT=value, *
      SRVNAME=value, *
      SRVNODE=value, *
      SRVRTRY=value, *
      SRVTERM=value, *
      SRVUSER=value, *
      SRVWAIT=value, *
      TRACE=value, *
      TRANSP=value

```

Siehe [Schlüsselwort-Subparameter für Client und Server](#), [Schlüsselwort-Subparameter nur für Client](#) und [Schlüsselwort-Subparameter nur für Server](#).

## Schlüsselwort-Subparameter für Client und Server

Folgende Schlüsselwort-Subparameter stehen für Client und Server zur Verfügung:

[ACIVERS](#) | [CPRPC](#) | [MAXBUFF](#) | [RPCSIZE](#) | [SERVER](#)

### ACIVERS - API-Version zur Verwendung mit EntireX Broker ACI

Dieser Parameter ist veraltet und wird ignoriert. Die höchstmögliche API-Version wird zwischen dem RPC-Nukleus und EntireX dynamisch ausgehandelt.

### CPRPC - Name der vom EntireX Broker verwendeten Codepage

CPRPC=*value* dient zur Angabe des Namens der vom EntireX Broker verwendeten Codepage.



**Anmerkung:** Zurzeit gilt er nur für den Natural RPC, wenn das Transportprotokoll ACI (d.h. EntireX Broker) verwendet wird.

<b>Mögliche Werte</b>	1 - 40 Zeichen	Gültiger Codepage-Name des EntireX Broker.
<b>Standard-Einstellung</b>	keine	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	



**Anmerkungen:**

1. Informationen zum EntireX Broker siehe Abschnitt über Internationalisierung bei der Software AG in der EntireX Broker-Dokumentation.
2. Siehe auch *Unicode- und Codepage-Unterstützung, Unicode-/Codepage-Umgebung konfigurieren und verwalten, Natural-Profilparameter und Parameter-Makros.*

### MAXBUFF - Maximale Puffergröße

MAXBUFF=*value* dient in einer Natural RPC-Umgebung zur Angabe der maximalen Pufferspeichergrößen.

<b>Mögliche Werte</b>	1 - 2097147, aber kleiner als oder gleich <i>value</i> = RPCSIZE - 4	Maximale Puffergröße in KB. <b>Anmerkung:</b> Die maximale Puffergröße muss gleich dem oder kleiner als der Wert (minus 4) sein, welcher mit dem Parameter RPCSIZE angegeben ist (für die Server-Seite siehe <a href="#">Abhängigkeit von der Parameteranzahl auf der Server-Seite</a> ).
	0	Es wird kein Puffer zugewiesen.
<b>Standard-Einstellung</b>	0	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	



**Anmerkungen:**

1. Auf der Server-Seite bestimmt `MAXBUFF` die Größe des Puffers, der vom Server vorgesehen wird, um Anforderungen (Requests) vom Client, einschließlich Daten, zu empfangen und um die Ergebnisse zurückzusenden. Der Puffer muss groß genug sein, um den größten von allen Client-Anforderungen empfangenen Datenbereich und alle an den Client zurückgesendeten Ergebnisse aufzunehmen. Wenn die Größe des Puffers für eine Anforderung nicht ausreicht, dann wird für diese Anforderung ein temporärer Puffer mit der benötigten Größe zugeordnet und benutzt.
2. Auf der Client-Seite bestimmt `MAXBUFF` die Größe des Puffers, der für die automatische Ausführung von Natural-RPC-Aufrufen vorgesehen ist. Dieser Puffer wird benutzt, um die Anforderung des Client zu erstellen und das Ergebnis vom Server zu empfangen. Der Puffer muss groß genug sein, um den größten Datenbereich von allen Client-Anforderungen empfangenen und alle an den Client zurückgesendeten Ergebnisse aufzunehmen. Wenn die Größe des Puffers für eine Anforderung nicht ausreicht, dann wird für diese Anforderung ein temporärer Puffer mit der benötigten Größe zugewiesen und benutzt.
3. Auf der Client-Seite brauchen Sie `MAXBUFF` nicht anzugeben, wenn Sie ein mit der `SYSRPC Utility-Utility` erzeugtes Interface-Objekt und `COMPAT NONE` benutzen und wenn die Parameter weder dynamische Felder noch X-Arrays oder Gruppenstrukturen enthalten.
4. Die Größe der Daten, die zwischen Client und Server ausgetauscht werden, wird von der Stub-Generierungsfunktion der `SYSRPC Utility-Utility` geliefert. Um die Größe für die RPC-Ausführung zu berechnen, können Sie das Kommando `SYSRPC CSMASS` benutzen; siehe *Benötigte Puffergröße berechnen* in der *Debugger und Dienstprogramme (Utilities)*-Dokumentation.

### Abhängigkeit von der Parameteranzahl auf der Server-Seite

Auf der Server-Seite ist der Unterschied zwischen `RPCSIZE` und `MAXBUFF` abhängig von der maximalen Zahl  $n$  der Parameter in der PDA (Parameter Data Area) und kann wie folgt berechnet werden:

- Wenn Gruppenstrukturen vorhanden sind:

$$\text{MAXBUFF} = \text{RPCSIZE} - (3 + n/10)$$

- Wenn keine Gruppenstrukturen vorhanden sind:

$$\text{MAXBUFF} = \text{RPCSIZE} - (3 + n/20)$$

Beispiel:

Wenn  $n=100$  und  $\text{RPCSIZE}=128$ , dann ist  $\text{MAXBUFF}=120$ .

### Abhängigkeit von den `ACIVERS`-Einstellungen

Im Falle eines EntireX Broker-Knotens gelten besondere Überlegungen, falls Sie Entire Net-Work als Transportschicht benutzen. Bei Entire Net-Work ist die Empfangspufferlänge, die an den EntireX Broker Broker Stub übergeben wird, durch den Startparameter `IUBL` begrenzt und darf 32 KB nicht überschreiten. Abhängig von der Einstellung des Schlüsselwort-Subparameters `ACIVERS` wird die Länge des Empfangspuffers wie folgt eingestellt:

- ACIVERS=1: 32000
- ACIVERS=2: 30K
- ACIVERS>2: der mit MAXBUFF angegebene Wert

Weitere Informationen siehe *Natural RPC (Remote Procedure Call)*-Dokumentation.

### RPCSIZE - Größe des vom Natural RPC benutzten Puffers

RPCSIZE=*value* gibt die Größe des vom Natural RPC benutzten Puffers an.

<b>Mögliche Werte</b>	1 - 2097151	Puffergröße in KB.  <b>Anmerkung:</b> Sollte die angegebene Größe nicht ausreichen, wird die Puffergröße auf Anforderung erhöht.
	0	Der Natural RPC kann nicht benutzt werden.
<b>Standard-Einstellung</b>	0	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	

### SERVER - Natural-Session als RPC-Server-Session starten

SERVER=*value* gibt an, ob die Natural-Session als RPC-Server-Session gestartet werden soll.

<b>Mögliche Werte</b>	ON	Die Natural-Session wird als RPC-Server-Session gestartet.
	OFF	Die Natural-Session wird nicht als RPC-Server-Session gestartet.
<b>Standard-Einstellung</b>	OFF	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	



#### Anmerkungen:

1. SERVER kann sowohl auf der Client- als auch auf der Server-Seite angegeben werden.
2. Für die statische Angabe steht dieser Parameter als Schlüsselwort-Subparameter des [NTRPC](#)-Makros zur Verfügung.
3. Für die dynamische Angabe steht dieser Parameter als Schlüsselwort-Subparameter des Profilparameters [RPC](#) zur Verfügung.
4. Weitere Informationen siehe *Natural RPC (Remote Procedure Call)*-Dokumentation.

## Schlüsselwort-Subparameter nur für Client

Folgende Schlüsselwort-Subparameter stehen nur für den Client zur Verfügung:

[AUTORPC](#) | [COMPR](#) | [DFS](#) | [RDS](#) | [RPCSDIR](#) | [TIMEOUT](#) | [TRYALT](#)



**Anmerkung:** Siehe auch *RPC-Client-spezifische Natural-Parameter setzen* in der *Natural RPC (Remote Procedure Call)*-Dokumentation.

### AUTORPC - Automatische Natural RPC-Ausführung

Mit `AUTORPC=value` legen Sie fest, ob der Natural RPC automatisch versucht, ein Programm, das lokal (auf der Client-Seite) nicht gefunden werden konnte, auf der Server-Seite (remote) auszuführen.

<b>Mögliche Werte</b>	ON	Der Natural RPC versucht automatisch, das lokal nicht gefundene Programm auf der Server-Seite auszuführen.
	OFF	Der Natural RPC versucht nicht, das lokal nicht gefundene Programm automatisch auf der Server-Seite (remote) auszuführen.  <b>Anmerkung:</b> Wenn <code>AUTORPC=OFF</code> gesetzt ist, können Sie <code>CALLNAT</code> -Aufrufe auf der Server-Seite (remote) mittels Schnittstellenobjekten ausführen.
<b>Standard-Einstellung</b>	OFF	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	ja	Dieser Wert kann zur Laufzeit mit Hilfe der Parameter Maintenance-Funktion der <code>YSRPC</code> -Utility überschrieben werden.

Wenn Sie ein Remote-`CALLNAT`-Statement benutzen wollen, um ein Subprogramm auf einem EntireX RPC Server auszuführen, empfehlen wir Ihnen dringend, `AUTORPC=OFF` zu setzen und ein Interface-Objekt zu benutzen. Weitere Informationen siehe *Interface-Objekte und automatische RPC-Ausführung* in der *Natural RPC (Remote Procedure Call)*-Dokumentation.

### COMPR - Komprimierung des RPC Buffer einstellen

`COMPR=value` kann benutzt werden, um die Komprimierung des Natural RPC Buffer einzustellen.

<b>Mögliche Werte</b>	0	Es erfolgt keine Komprimierung.
	1	Der Sende-Puffer enthält änderbare Felder und Ausgabefelder und den Format-Puffer. Der Antwort-Buffer enthält änderbare Felder und Eingabefelder.
	2	Wie bei <code>COMPR=1</code> , zusätzlich enthält der Antwort-Puffer auch noch den Format-Puffer.
<b>Standard-Einstellung</b>	1	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	ja	Zur Laufzeit kann dieser Wert in der <i>Parameter Maintenance</i> -Funktion der <code>SYSRPC-Utility</code> überschrieben werden.

`COMPR` ist nur dann wirksam, wenn die automatische Komprimierung des Natural RPC Buffer verwendet wird (`AUTORPC=ON`) und der `CALLNAT` ohne Stub ausgeführt wird. Bei Verwendung eines Stub ist die Komprimierung bereits bei der Stub-Generierung gesetzt worden. Einzelheiten siehe *Komprimierung verwenden* in der *Natural RPC (Remote Procedure Call)*-Dokumentation.

### DFS - Standard-Serveradresse für RPC Client angeben

`DFS=value` definiert eine Standard-RPC-Serveradresse. Es können bis zu 5 positionsgebundene Subparameter angegeben werden.

<b>Mögliche Werte</b>	Siehe <i>DFS-Subparameter-Syntax</i> .	
<b>Standard-Einstellung</b>	keine	Subparameter-Standardwerte, siehe <i>DFS-Subparameter-Syntax</i> .
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	ja	Zur Laufzeit kann dieser Wert mittels der Natural-Anwendungsprogrammierschnittstelle <code>USR2007N</code> überschrieben werden.
<b>Anwendungsprogrammierschnittstelle</b>	<code>USR2007N</code>	Siehe <i>Anwendungsprogrammierschnittstellen zur Verwendung beim Natural RPC</i> in der <i>Natural RPC (Remote Procedure Call)</i> -Dokumentation und <i>SYSEXT Utility</i> in der <i>Debugger und Dienstprogramme (Utilities)</i> -Dokumentation.

`DFS` enthält Angaben zu Servername, Serverknoten, Logon-Indikator und Transportprotokoll. Die Standard-Serveradresse wird nur dann verwendet, wenn im Service Directory kein passender Server gefunden wird. Weitere Informationen siehe *RPC-Server-Adressen festlegen* in der *Natural RPC (Remote Procedure Call)*-Dokumentation.

### DFS-Subparameter-Syntax

Der Schlüsselwort-Subparameter `DFS` wird wie folgt angegeben:

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

Dabei ist:

Syntax-Element	Wert	Erklärung
<i>server-name</i>	1 - 32 Zeichen	Gültiger Server-Name. Siehe auch Schlüsselwort-Subparameter <a href="#">SRVNAME</a> .  Es gibt keine Standard-Einstellung, der Wert muss angegeben werden.
<i>server-node</i>	1 - 192 Zeichen	Knotenname. Siehe auch Schlüsselwort-Subparameter <a href="#">SRVNODE</a> .  Es gibt keine Standard-Einstellung, der Wert muss angegeben werden.
<i>logon-indicator</i>	L	Der Client initiiert einen Natural-Logon beim Server mit dem Library-Namen der aktuellen Library auf dem Client. Auf Windows-Plattformen: Statt L anzugeben, markieren Sie das Auswahlkästchen.
	(leer)	Es wird kein Server-Logon ausgeführt. Falls nichts angegeben wird, ist „leer“ die Standard-Einstellung.
<i>transport-protocol-name</i>	ACI	Name des zu verwendende Transportprotokols. ACI ist der einzig mögliche Wert und die Standard-Einstellung.
<i>service-directory-indicator</i>	SERVDIR	Ein Service Directory muss vor der Auswertung des Schlüsselwort-Subparameter DFS vorhanden sein.
	NOSERVDIR	Es wird vor Auswertung des Schlüsselwort-Subparameter DFS kein Service Directory benutzt; d.h., auf der Client-Seite braucht kein Service Directory vorhanden zu sein.  Wenn nichts angegeben wird, ist SERVDIR die Standard-Einstellung.

### RDS - Remote Directory Server-Definition

Mit `RDS=value` können Sie in einer Natural RPC-Umgebung bis zu 10 Remote Directory Server definieren. Für jeden Remote Directory Server können Sie bis zu 5 positionsgebundene Subparameter angeben.

<b>Mögliche Werte</b>	Siehe <a href="#">RDS-Subparameter-Syntax</a> .	
<b>Standard-Einstellung</b>	keine	Subparameter-Standardwerte, siehe <a href="#">RDS-Subparameter-Syntax</a> .
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	

### RDS-Subparameter-Syntax

Der Parameter RDS wird wie folgt angegeben:

#### Bei Nutzung nur eines Servers:

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

#### Bei Nutzung von 2-10 Servern:

```
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))
```

Dabei ist:

Syntax-Element	Wert	Erklärung
<i>server-name</i>	1 - 8 Zeichen	Name des Servers.  Es gibt keine Standardeinstellung, der Wert muss angegeben werden.
<i>server-node-name</i>	1 - 8 Zeichen	Name des Serverknotens.  Es gibt keine Standardeinstellung, der Wert muss angegeben werden.
<i>subprogram</i>	1 - 8 Zeichen	Name des Subprogramms, bezeichnet als CALLNAT, das als Interface benutzt werden soll.  Der Standardname ist RDSSCDIR.
<i>logon-indicator</i>	L	Der Client initiiert ein Natural-Logon auf den Server mit dem Library-Namen der aktuellen Library auf dem Client.  Auf Windows-Plattformen: Statt der Angabe von L muss die entsprechende Auswahlbox markiert werden.
	(leer)	Leer bedeutet, dass kein Server-Logon durchgeführt wird. Wenn nichts angegeben wird, ist dies die Standardeinstellung.
<i>transport-protocol-name</i>	ACI	Der Name des zu verwendende Transportprotokolls. ACI ist der einzig mögliche Wert und der Standardwert

## RPCSDIR - Library für Service Directory

RPCSDIR=*value* gibt den Namen der Natural Library (oder den Namen einer ihrer Steplibs) an, die zur Laufzeit vom RPC Client benutzt wird.

<b>Mögliche Werte</b>	1 - 8 Zeichen	Gültiger Name einer Natural Library.
<b>Standard-Einstellung</b>	keine	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	

Der Schlüsselwort-Subparameter RPCSDIR wird von den Funktionen Service Directory Maintenance und Server Command Execution der SYSRPC Utility ausgewertet.

## TIMEOUT - Wartezeit auf Server-Antwort

TIMEOUT=*value* gibt die Anzahl der Sekunden an, die der Client auf eine RPC-Server-Antwort warten soll.

<b>Mögliche Werte</b>	0 - 32767	Zeitvorgabe in Sekunden.  <b>Anmerkung:</b> Wenn diese Zeit überschritten wird, wird der RPC-Aufruf mit einer entsprechenden Meldung beendet.
<b>Standard-Einstellung</b>	55	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	ja	Zur Laufzeit kann dieser Wert mit der <i>Parameter Maintenance</i> -Funktion der SYSRPC Utility-Utility oder mit der Anwendungsprogrammierschnittstelle USR2076N überschrieben werden.

## TRYALT - Alternative Server-Adresse versuchen

TRYALT=*value* gibt an, ob der RPC Client versuchen soll, eine RPC-Anforderung auf einem alternativen Server auszuführen oder nicht.

<b>Mögliche Werte</b>	ON	Wenn eine Anforderung nicht auf dem angegebenen Knoten ausgeführt werden konnte, versucht der RPC Client, eine alternative Server-Adresse zu finden, um die Anforderung dorthin zu senden.
	OFF	Es wird kein solcher Versuch unternommen.
<b>Standard-Einstellung</b>	OFF	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	ja	Zur Laufzeit kann der Wert mit der <i>Parameter Maintenance</i> -Funktion der SYSRPC Utility überschrieben werden.

Weitere Informationen siehe *RPC-Server-Adressen festlegen* in der *Natural RPC (Remote Procedure Call)*-Dokumentation.

## Schlüsselwort-Subparameter nur für Server

Folgende Schlüsselwort-Subparameter stehen nur für den Server zur Verfügung:

LOGONRQ | NTASKS | RPCUCT | SRVCMIT | SRVNAME | SRVNODE | SRVRTRY | SRVTERM | SRVUSER | SRVWAIT  
| TRACE | TRANSP



**Anmerkung:** Siehe auch *RPC-Server-spezifische Natural-Parameter setzen* in der *Natural RPC (Remote Procedure Call)*-Dokumentation..

### LOGONRQ - Logon für RPC Server-Anforderung

LOGONRQ=*value* bestimmt, ob bei einer RPC-Server-Anforderung Anmeldedaten erforderlich sind oder nicht.

<b>Mögliche Werte</b>	ON	Eine Anmeldung ist erforderlich; d.h., der Server akzeptiert nur Anforderungen von Clients, die Anmeldedaten übergeben. Bei konversationellen Anforderungen sind die Anmeldedaten nur nötig, wenn die Konversation eröffnet wird.  <b>Anmerkung:</b> Wenn der Natural-RPC-Server unter Natural Security läuft, wird dringend empfohlen, LOGONRQ auf ON zu setzen. Weitere Informationen siehe <i>Natural RPC mit Natural Security verwenden</i> in der <i>Natural RPC (Remote Procedure Call)</i> -Dokumentation.
	OFF	Eine Anmeldung ist <i>nicht erforderlich</i> . Die Anmeldedaten werden trotzdem verarbeitet.
<b>Standard-Einstellung</b>	OFF	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	



**Anmerkung:** Bei Natural-Clients können die Anmeldedaten wie folgt angefordert werden: Entweder durch Setzen der LOGON-Option in der Service Directory Maintenance-Funktion der SYSRPC Utility oder durch Verwendung des **Logon-Indikators** des Schlüsselwort-Subparameters DFS.

## NTASKS - Anzahl der zu startenden Server-Tasks

NTASKS=*value* gibt die minimale Anzahl an Server Taks an, die bei der Server-Initialisierung gestartet werden sollen, und die maximale Anzahl an Server-Tasks, die gleichzeitig aktiv sein sollen.

<b>Mögliche Werte:</b>	Siehe <a href="#">NTASKS-Subparameter-Syntax</a> .	
<b>Standard-Einstellung:</b>	1, 1	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	



### Anmerkungen:

1. NTASKS gilt nur für Server, die im Batch-Modus unter z/OS gestartet wurden, und für Server, die von einem RPC Server Front-end gestartet wurden.
2. Wenn der Server eine große Anzahl an Anforderungen vom Client bearbeiten muss, können Sie diesen Subparameter benutzen, um den Durchsatz zu verbessern, indem Sie mehrere Nachbildungen desselben Server Task (mit identischem Namen) starten.
3. Weitere Informationen siehe *Natural RPC (Remote Procedure Call)*-Dokumentation, und dort den Abschnitt *Besondere Aspekte bei Großrechner-Natural RPC-Servern mit Replicaten*.

## NTASKS-Subparameter-Syntax

Der Parameter NTASKS wird wie folgt angegeben:

```
NTASKS=(min,max)
```

Oder:

```
NTASKS=min
```



**Anmerkung:** Wenn nur *min* angegeben wird, können die Klammern weggelassen werden.

Dabei ist:

Syntax-Element	Wert	Erklärung
<i>min</i>	1 - 99	Minimale Anzahl der Server Tasks, die bei der Server-Initialisierung gestartet werden sollen.
<i>max</i>	1 - <i>n</i> 0 (unbegrenzt)	Maximale Anzahl an Server Tasks, die gleichzeitig aktiv sein können. <b>Anmerkung:</b> Die maximale Anzahl <i>max</i> der Server Tasks gilt nur für Server, die von einem RPC Server Front-end gestartet wurden.



**Anmerkung:** Wenn Sie für *max* keinen Wert angeben, dann wird standardmäßig der Wert 1 angenommen, wenn *min* den Wert 1 hat. Wenn *min* einen Wert größer als 1 hat, dann wird für *max* standardmäßig der Wert 0 angenommen.

### RPCUCT - Subprogrammnamen in Großschreibung umsetzen

RPCUCT=*value* gibt an, ob der Natural RPC Server den Namen des auszuführenden Remote CALLNAT in Großbuchstaben umsetzt.

<b>Mögliche Werte</b>	ON	Der Name des vom Natural RPC Server auszuführenden Remote CALLNAT-Subprogramms wird in Großbuchstaben umgesetzt, bevor der CALLNAT aufgerufen wird.  Durch diese Option werden Nicht-Natural RPC Clients unterstützt, die in Subprogrammnamen Zeichen in gemischter Groß-/Kleinschreibung verwenden.  <b>Anmerkung:</b> Auf UNIX- und Windows-Plattformen erfolgt eine implizite Umsetzung in Großschreibung durch Natural selbst. RPCUCT=ON ist daher der Kompatibilitätsmodus für Natural RPC Server auf Großrechnern und Natural RPC Server auf UNIX- und Windows-Plattformen.
	OFF	Der Name des vom Natural RPC Server auszuführenden Remote CALLNAT--Subprogramms wird nicht verändert. Falls der Name Zeichen in Kleinschreibung enthält, ist eine Fehlermeldung NAT00082 zu erwarten.
<b>Standard-Einstellung</b>	ON	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	

### SRVCMIT - Server-Bestätigungszeitpunkt

SRVCMIT=*value* legt den Zeitpunkt fest, zu dem ein Natural RPC Server eine RPC-Konversation oder eine nicht-konversationelle RPC-Anforderung implizit bestätigt.

<b>Mögliche Werte</b>	B	Der Natural RPC Server bestätigt eine Datenbanktransaktion automatisch, bevor eine Antwort an den Client gesendet wird.  <b>Anmerkung:</b> Falls die Antwort fehlschlägt, ist die Datenbanktransaktion bereits bestätigt.
	A	Der Natural RPC Server bestätigt eine Datenbanktransaktion automatisch, nachdem eine Antwort erfolgreich an den Client übermittelt worden ist.  <b>Anmerkung:</b> Falls die Antwort fehlschlägt, wird die Datenbanktransaktion rückgängig gemacht.
<b>Standard-Einstellung</b>	B	
<b>Dynamische Angabe</b>	ja	

<b>Angabe innerhalb der Session</b>	nein	
-------------------------------------	------	--

 **Anmerkung:** Dieser Parameter wird nur dann ausgewertet, wenn der Profilparameter `ETEOP` auf `ON` gesetzt ist.

### SRVNAME - Name des RPC-Servers

`SRVNAME=value` gibt den Namen des RPC-Servers an, mit dem dieser sich bei dem Knoten registriert, der mit dem Schlüsselwort-Subparameter `SRVNODE` angegeben ist.

<b>Mögliche Werte</b>	1 - 32 Zeichen	Gültiger Servername.  Im Falle eines EntireX Broker-Knotens entspricht der Wert von <code>SRVNAME</code> dem Wert des <code>SERVER</code> -Attributs eines Service-Eintrags in der Broker Attribut-Datei, wie nachfolgend angegeben:  <code>CLASS=RPC, SERVICE=CALLNAT, SERVER=svname</code>  Siehe <a href="#">Beispiel</a> .
<b>Standard-Einstellung</b>	keine	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	

#### Beispiel:

```
SRVNAME='PRODUCTION_SERVER'
```

### SRVNODE - Name des Knotens

`SRVNODE=value` gibt den Namen des Knotens an, bei dem sich der RPC Server registriert.

<b>Mögliche Werte</b>	1 - 192 Zeichen	Knotenname.  Im Falle eines EntireX Broker-Knotens kann sich ein Knotenname auf einen Entire Net-Work-Knoten oder auf eine TCP/IP-Adresse beziehen. Beachten Sie bitte, dass der zur Verwendung kommende Broker Stub die Namensnotation unterstützen muss. Einzelheiten zur Struktur von Knotennamen und zu ihrer Unterstützung durch die EntireX Broker Stubs siehe EntireX-Dokumentation.  Siehe <a href="#">Beispiele</a> .
<b>Standard-Einstellung</b>	keine	
<b>Dynamische Angabe</b>	ja	

<b>Angabe innerhalb der Session</b>	nein	
-------------------------------------	------	--

### Beispiele

Die folgenden Beispiele basieren auf der 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 with port number */
```

 **Anmerkungen:**

1. Wenn für die TCP/IP-Adresse ein Host-Name benutzt wird, muss der Name entweder bei Ihrem DNS Server bekannt sein oder er muss in der Hosts-Datei Ihrer TCP/IP-Konfiguration definiert werden.
2. Wird die Port-Nummer weggelassen, dann wird entweder die Standard-Port-Nummer vom EntireX Broker Stub benutzt oder es muss ein Host-Name benutzt werden und der Host-Name muss Ihrem DNS Server bekannt sein oder er muss in der Services-Datei Ihrer TCP/IP-Konfiguration definiert werden.

### SRVTRY - Anzahl der Verbindungsaufnahmeversuche

`SRVTRY=va lue` gibt für einen RPC-Server die Anzahl der Versuche an, eine Verbindung oder erneute Verbindung (REGISTER) zu einem EntireX Broker herzustellen, der nicht aktiv ist, und die Wartezeit zwischen zwei aufeinanderfolgenden Versuchen.

<b>Mögliche Werte</b>	Siehe <a href="#">SRVTRY-Subparameter-Syntax</a> .	
<b>Standard-Einstellung</b>	0,60	Keine Versuche.
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	

### SRVTRY-Subparameter-Syntax

Der Parameter SRVTRY wird wie folgt angegeben:

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

Oder:

```
SRVRTRY=attempts
```

 **Anmerkung:** Wenn nur ein Wert für *attempts* angegeben wird, können die Klammern weggelassen werden.

Dabei ist:

Syntax-Element	Wert	Erklärung
<i>attempts</i>	0 oder 1 - 2147483647	Anzahl der Connect/Reconnect-Versuche bei einem EntireX Broker, der nicht aktiv ist (EntireX Broker-Meldung 02150148).  <b>Anmerkung:</b>  1. Mit der Angabe für <i>attempts</i> können Sie einen Natural RPC Server starten, bevor der erforderliche EntireX Broker gestartet wurde, und einen Natural RPC Server starten, bevor der erforderliche EntireX Broker gestartet wurde, und einen EntireX Broker zeitweilig schließen, ohne implizit alle Natural RPC Server zu beenden.  2. Wenn nach Erreichen der angegebenen Anzahl der mit <i>attempts</i> angegebenen Versuche der EntireX Broker immer noch nicht aktiv ist oder wenn <i>attempts</i> auf Null gesetzt ist, dann wird der RPC Server beendet.
<i>wait-time</i>	0 oder 1 - 3600	Wartezeit zwischen zwei aufeinanderfolgenden Versuchen.

## Beispiele

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

oder

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

20 Versuche mit einer Wartezeit von 10 Sekunden zwischen zwei aufeinanderfolgenden Versuchen.

2. `RPC=(SRVRTRY=500)`

oder

NTRPC SRVTRY=500

500 Versuche mit einer Wartezeit von 60 Sekunden zwischen zwei aufeinanderfolgenden Versuchen.

 **Anmerkung:** Weitere Informationen siehe *Natural RPC (Remote Procedure Call)*-Dokumentation, und dort insbesondere *Besondere Aspekte bei Großrechner-Natural RPC-Servern mit Replicaten*.

### SRVTERM - Ereignis für Server-Beendigung

SRVTERM=*value* gibt das Ereignis an, bei dessen Auftreten ein Natural RPC Server automatisch beendet wird.

<b>Mögliche Werte</b>	NEVER	Ein Natural RPC Server wird niemals automatisch beendet.  <b>Anmerkung:</b> Um einen Natural RPC Server zu beenden siehe <i>Beenden eines Natural RPC-Servers</i> und <i>Anwendungsprogrammierschnittstelle USR2075N verwenden</i> (für den EntireX Broker Service) in der <i>Natural RPC (Remote Procedure Call)</i> -Dokumentation.
	TIMEOUT	Ein Natural RPC Server wird automatisch beendet, wenn die Wartezeit auf die nächste Client-Anforderung außerhalb einer RPC-Konversation überschritten wird.  <b>Anmerkung:</b> TIMEOUT sollte nur gesetzt werden, wenn Sie einen Attach Manager benutzen, um Natural RPC Server dynamisch bei Anforderung zu starten.
<b>Standard-Einstellung</b>	NEVER	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	

### SRVUSER - Benutzererkennung für RPC Server-Registrierung

SRVUSER=*value* gibt die Benutzererkennung an, die benötigt wird, um einen Natural RPC Server bei dem Knoten zu registrieren, der mit dem Schlüsselwort-Subparameter **SVRNODE** angegeben wird.

 **Anmerkung:** Im Falle eines EntireX Broker-Knotens wird SRVUSER außerdem zum Anmelden beim EntireX Broker benutzt. Ein Passwort wird entweder von Natural Security genommen (siehe '\*NSC' weiter unten) oder über die Anwendungsprogrammierschnittstelle USR2072N angegeben.

<b>Mögliche Werte</b>	<i>user-ID</i>	Gültige Benutzerkennung, 1 - 16 Zeichen.
	'*USER'	Wird SRVUSER auf '*USER' gesetzt, dann verwendet der Natural Server die aktuelle Natural-Benutzerkennung (siehe Systemvariable *USER) zum Anmelden bei dem Knoten.
	'*NSC'	Wenn SRVUSER auf '*NSC' gesetzt und wenn Natural Security installiert ist, dann verwendet der Natural Server die aktuelle Natural-Benutzerkennung (siehe Systemvariable *USER) und das in Natural Security zu dieser Benutzerkennung definierte Passwort zum Anmelden bei dem Knoten.  Falls Natural unter Natural SAF Security läuft, dürfen Sie '*NSC' nicht verwenden, weil das in Natural Security definierte Passwort normalerweise kein gültiges Passwort zum Anmelden beim EntireX Broker-Knoten ist. Siehe auch <i>EntireX Security auf der Server-Seite</i> in der <i>Natural RPC (Remote Procedure Call)</i> -Dokumentation.
<b>Standard-Einstellung</b>	<i>timestamp</i>	Wird die Benutzerkennung weggelassen, dann wird die Zeitmarke verwendet.
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	

### SRVWAIT - Wartezeit für RPC Server

SRVWAIT=*value* gibt die Anzahl an Sekunden an, die der Server auf eine Anforderung vom Natural RPC Client warten soll.

<b>Mögliche Werte</b>	0 oder 1 - 32767	Wartezeit in Sekunden.  <b>Anmerkung:</b>  1. Wenn die angegebene Zeit überschritten wird, wird der RPC Server durch den Knoten informiert, auf dem sich der PRC Server registriert hat. Der RPC Server schreibt eine entsprechende Meldung in die Natural RPC Server Trace-Datei und wartet weiterhin auf eine RPC Client-Anforderung.  2. Wenn zur Kommunikation mit dem Knoten TCP/IP verwendet wird, führt die Angabe eines Wertes größer als Null zur Vermeidung einer unbefristeten Wartezeit im TCP/IP, falls der Knoten aus irgendeinem Grund nicht reagieren kann.
	<b>Standard-Einstellung</b>	0  <b>Anmerkung:</b> Im Falle eines EntireX Broker-Knotens wird die Wartezeit auf den SERVER-NONACT-Wert der entsprechenden EntireX Broker-Attributdatei gesetzt.
<b>Dynamische Angabe</b>	ja	

<b>Angabe innerhalb der Session</b>	nein	
-------------------------------------	------	--

### TRACE - Trace Level für Natural RPC Server

Dieser Parameter aktiviert die RPC-Trace-Funktion und gibt den Trace Level an, der verwendet werden soll.

<b>Mögliche Werte</b>	0	Es erfolgt kein Trace.
	1	Der Trace erstreckt sich nur auf Meldungen (einschließlich Natural-Fehler).
	(1, E)	Im Fehlerfall erstreckt sich der Trace nur auf Meldungen.
	2	Der Trace erstreckt sich auf alle Meldungen und Daten vom/zum Client.
	(2, E)	Nur im Fehlerfall erfolgt ein Trace auf Meldungen und Daten vom/zum Client.
	3 - 9	Die Werte 3 - 9 werden ebenfalls akzeptiert. Diese Werte sind für zukünftige Verwendung vorgesehen und haben dieselbe Bedeutung wie TRACE=2.
<b>Standard-Einstellung</b>	0	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	

Weitere Informationen siehe *Server-Trace-Funktion benutzen in der Natural RPC (Remote Procedure Call)-Dokumentation*.

### TRANSP - Server-Transportprotokoll

Dieser Parameter gibt an, welches Server-Transport-Protokoll verwendet wird. Bei Angabe von ACI können Sie zusätzlich noch die Transportmethode angeben.



**Anmerkung:** Die Verwendung von TRANSP ist nicht mehr nötig, weil Sie jetzt den vollständigen Knotennamen mit dem Schlüsselwort-Subparameter **SRVNODE** angeben können. TRANSP wird nur noch aus Kompatibilitätsgründen unterstützt.

<b>Mögliche Werte</b>	ACI	ACI wird verwendet. Die Transportmethode wird durch den EntireX Broker definiert.
	(ACI, TCP)	ACI wird mit TCP/IP verwendet.
	(ACI, NET)	ACI wird mit Entire Network verwendet, d.h. unter Verwendung des Adabas-Protokolls.
	(ACI, TCP-NET)	Es wird versucht, ACI mit TCP/IP zu verwenden. Falls nicht verfügbar, wird ACI mit NET verwendet.

	(ACI, NET-TCP)	Es wird versucht, ACI mit NET zu verwenden. Falls nicht verfügbar, wird ACI mit TCP verwendet.
<b>Standard-Einstellung</b>	ACI	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	

## Beispiel für RPC-Parameter

Für den Client:

```
RPC=(RPCSIZE=80,MAXBUFF=30,AUTORPC=ON,DFS=(MYSERV,MYNODE,,ACI))
```

Für den Server:

```
RPC=(RPCSIZE=80,MAXBUFF=30,SRVNAME=MYSERV,SRVNODE=MYNODE,SERVER=ON)
```

## Beispiel für NTRPC-Makro

Für den Client:

```
NTRPC RPCSIZE=80, *
      MAXBUFF=30, *
      AUTORPC=ON, *
      DFS=(MYSERV,MYNODE1,,ACI), *
      RDS=((SRVX,NODEX),(SRVY,NODEY))
```

Für den Server:

```
NTRPC RPCSIZE=80, *
      MAXBUFF=30, *
      SRVNAME=MYSERV, *
      SRVNODE=MYNODE, *
      SERVER=ON
```



# 218

## RUNSIZE - Größe des Laufzeit-Puffers

---

Dieser Natural-Profilparameter gibt die Größe des Natural-Laufzeit-Puffers an.

<b>Mögliche Werte</b>	10 - 64	Puffergröße in KB.
<b>Standard-Einstellung</b>	16	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	



**Anmerkung:** Alternativ können Sie den äquivalenten Natural-Profilparameter [DS](#) oder das Parametermakro [NTDS](#) benutzen.

Der Natural-Laufzeit-Puffer enthält folgende Informationen:

- definierte Steplibs,
- die Datei-Umsetzungstabelle (Profilparameter [TF](#)),
- Protokollinformationen zum zuletzt benutzten Kommando,
- Umgebungs-Stack (für Benutzereinstellungen),
- Informationen über den zuletzt aufgetretenen Datenbankfehler,
- PF-Tastennamen, wenn im SAA-Modus gearbeitet wird,
- verschiedene interne Arbeitsbereiche und Steuerungsinformationen.

---

# 219

## SA - Terminal-Warnton

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

<b>Mögliche Werte</b>	ON	Der Warnton wird jedesmal ausgelöst, wenn Natural den Benutzer zu einer Eingabe auffordert..  <b>Anmerkung:</b> 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 ALARM-Option eines REINPUT-Statements ausgelöst werden.	
<b>Standard-Einstellung</b>	OFF		
<b>Dynamische Angabe</b>	ja		
<b>Angabe innerhalb der Session</b>	ja	Gültige Statements:	SET GLOBALS
		Gültiges Kommando:	GLOBALS
<b>Programmierschnittstelle (API)</b>	USR1005N	Siehe <i>SYSEXT Utility</i> in der Dokumentation <i>Debugger und Dienstprogramme</i> .	



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



# 220

## SB - Auswahlfeld

---

- Anmerkungen zur Syntax ..... 672
- Anmerkungen zur Laufzeitumgebung ..... 673
- Funktionen ..... 674
- Beschränkungen ..... 676

Auswahlfelder stehen in einem `INPUT`-Statement ausschließlich auf Großrechnern zur Verfügung. Für andere Plattformen können Auswahlfelder nur im Masken-Editor (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
<code>operand1</code>	C S A	A	ja	nein

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:

<b>Systemvariablen</b>	Zum Beispiel: *PROGRAM, *COM
<b>Benannte Konstanten (nur auf Großrechnern)</b>	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 Hilfenfenster 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.

# 221

## SCTAB - Scanner-Zeichentabelle

---

▪ SCTAB-Parameter-Syntax .....	678
▪ NTSCTAB-Makro-Syntax .....	679
▪ Beispiel für SCTAB-Parameter .....	680
▪ Beispiel für NTSCTAB-Makros .....	680

Mit diesem Natural-Profilparameter können Sie die Definitionen der im Konfigurationsmodul NATCONFIG enthaltenen Scanner-Zeichentabelle NTSCTAB überschreiben. SCTAB entspricht dem Makro NTSCTAB im Natural-Parametermodul.

<b>Mögliche Werte</b>	Siehe <i>SCTAB-Parameter-Syntax</i> weiter unten.	
<b>Standard-Einstellung</b>	Gemäß den Angaben im Makro NTSCTAB in NATCONFIG.	
<b>Dynamische Angabe</b>	ja	Dieser Parameter kann nur dynamisch angegeben werden. Im Natural-Parametermodul wird stattdessen das Makro NTSCTAB verwendet.
<b>Angabe innerhalb der Session</b>	nein	



#### Anmerkungen:

1. Die Tabelle NTSCTAB definiert die Eigenschaften von Zeichen, die in Maskendefinitionen für die MASK-Option und als Begrenzungszeichen in den Statements EXAMINE and SEPARATE verwendet werden.
2. Wenn der Profilparameter CP auf einen anderen Wert als OFF gesetzt ist, werden die mit SCTAB angegebenen Werte ignoriert. Siehe auch *Umsetzungstabellen (Translation Tables)* in der *Unicode- und Codepage-Unterstützung-Dokumentation*.

## SCTAB-Parameter-Syntax

Der Parameter SCTAB wird wie folgt angegeben:

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



**Anmerkung:** Sie können in der Liste der Werte mehr als ein Zeichen angeben. Dann müssen Sie die ganze Zeichenkette mit Zeichen/Attribut-Paaren in Klammern setzen.

Oder:

```
SCTAB=OFF
```

Dabei ist:

Syntax-Element / Wert	Erklärung	
<i>character</i>	Geben Sie ein Zeichen und danach seine(n) Attribut-Typ(en) an. Sie können das Zeichen entweder als das Ein-Byte-Zeichen selbst (in Hochkommas) oder als Hexidezimal-Darstellung dieses Zeichens angeben.	
<i>attribute-type</i>	Mögliche Attribut-Typen:	
	UPPER	Großbuchstaben, alphabetisch
	LOWER	Kleinbuchstaben, alphabetisch
	NUM	numerisch
	HEX	hexadezimal
	ALFANUM	alphanumerisch
	SPECIAL	Sonderzeichen
	NDELIM	Nicht-Begrenzungszeichen
OFF	SCTAB=OFF bewirkt, dass alle (statischen und dynamischen) Definitionen auf Werte zurückgesetzt werden, die im Makro NTSCATB im Modul NATCONFIG angegeben sind.	

## NTSCATB-Makro-Syntax

Das Makro NTSCATB wird wie folgt angegeben:

```
NTSCATB=character, attribute-type, attribute-type...
NTSCATB=character, attribute-type, attribute-type...
```



### Anmerkungen:

1. Erläuterung der Syntax-Elemente siehe [SCTAB-Parameter-Syntax](#).
2. Der Wert OFF kann nicht mit dem Makro NTSCATB, sondern nur dynamisch mit dem Profilparameter SCTAB angegeben werden.
3. Für jedes zu überschreibende Zeichen müssen Sie ein separates NTSCATB-Makro angeben, siehe [Beispiel für NTSCATB-Makros](#).

## Beispiel für SCTAB-Parameter

---

```
SCTAB=(5E,LOWER,NDELIM,'B',SPECIAL,7B,SPECIAL,'Ä',UPPER,NDELIM)
```

## Beispiel für NTSC TAB-Makros

---

```
NTSC TAB 5E,LOWER,NDELIM  
NTSC TAB 'B',SPECIAL  
NTSC TAB 7B,SPECIAL  
NTSC TAB 'Ä',UPPER,NDELIM
```

# 222

## SELUNIT - Aktivierung ausgewählter Natural-Merkmale

---

- Zur Verfügung stehende wählbare Natural-Merkmale ("Selectable Units") ..... 682
- SELUNIT-Parameter-Syntax ..... 683
- Beispiel für SELUNIT-Parameter ..... 683

Mit diesem Natural-Profilparameter können Sie neue oder geänderte Natural-Merkmale aktivieren, die als wählbare Natural-Merkmale („Selectable Units“) ausgeliefert werden. Dies bietet Ihnen die Möglichkeit, anstatt ein komplettes Natural Upgrade auszuführen, nur die Merkmale zu aktivieren, die in Ihrer Umgebung tatsächlich benötigt werden.

Weitere Informationen siehe *Selectable Units for New Natural Features* in der *Operations*-Dokumentation.

<b>Mögliche Werte</b>	OFF	Alle neuen und geänderten wählbaren Natural-Merkmale („Selectable Units“) werden deaktiviert; eventuell vorhandene Schlüsselwort-Subparameter-Einstellungen werden ausgeschaltet.
	$Un=\{ON OFF\}$	Eines oder mehrere wählbare Natural-Merkmale („Selectable Units“) werden mit dem Schlüsselwort-Subparameter $Un$ aktiviert (ON) bzw. deaktiviert (OFF). Siehe <a href="#">SELUNIT-Parameter-Syntax</a> .  Mit dem Systemkommando SHOWSU können Sie sich alle in Ihrer Umgebung zur Verfügung stehenden sowie alle aktivierten wählbaren Natural-Merkmale („Selectable Units“) auflisten lassen. Außerdem können Sie beim Systemkommando SYSPROD das Zeilenkommando bzw. Schlüsselwort SU benutzen, um zu prüfen, ob wählbare Natural-Merkmale („Selectable Units“) aktiv sind.
<b>Standard-Einstellung</b>	OFF	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	

## Zur Verfügung stehende wählbare Natural-Merkmale ("Selectable Units")

Folgende wählbare Natural-Merkmale („Selectable Units“) stehen unter z/OS zur Verfügung:

Natural-Merkmal („Selectable Unit“)	Merkmal/Funktion
U1	Komprimierung und Dekomprimierung von Natural Threads durch den Natural Roll Server während der Aus- und Einlagerungsoperationen (Roll In/Roll Out). Dies verringert die CPU-Nutzung durch den Natural-Nukleus und somit die CPU-Belastung durch das als Host fungierende TP-System (CICS oder IMS TM) oder den Natural-Server (Natural RPC Server oder Natural Development Server).  Die Komprimierungsroutinen laufen auf einem IBM System z Integrated Information Processor (zIIP) - falls vorhanden - und verbessern so die Gesamt-Performance von Natural.

## SELUNIT-Parameter-Syntax

---

Für den Parameter gilt folgende Syntax:

```
SELUNIT={OFF | ({Un={ON | OFF}},... ) }
```

Dabei ist:

*n* eine Zahl von 1 bis 24, die jeweils für ein bestimmtes Merkmal steht, das in den aktuellen *Natural Release Notes* für Großrechner angekündigt wird.

## Beispiel für SELUNIT-Parameter

---

```
SELUNIT=(U1=ON,U4=OFF)
```



# 223

## SENDER - Bestimmungsort für Bildschirmausgabe bei

### asynchroner Verarbeitung

---

Dieser Natural-Profilparameter gibt den Bestimmungsort an, an dem Ausgaben von einer asynchronen Anwendung angezeigt werden sollen.

<b>Mögliche Werte</b>	1 - 8 Zeichen	Ausgabe-Bestimmungsort, z.B. Drucker.
<b>Standard-Einstellung</b>	keine	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	



#### Anmerkungen:

1. Dieser Natural-Profilparameter gilt nur bei Natural unter CICS, Complete und IMS/TM.
2. Der angegebene Bestimmungsort gilt für Hardcopy-Ausgaben und primäre Reports.
3. Zusätzliche Reports werden an den Bestimmungsort gesendet, der mit dem Statement `DEFINE PRINTER` angegeben wird (so wie in einer asynchronen Online-Session).

#### Plattformspezifische Eigenschaften

Es gelten folgende plattformspezifische Eigenschaften:

Plattform	Kommentar
CICS	<p>Der Profilparameter <code>SENDER</code> gibt den Bestimmungsort für transiente CICS-Daten (TD) und das Terminal oder den Drucker für Ausgaben von asynchronen Sessions an. Falls der angegebene Bestimmungsort nicht existiert, wird die Session-Ausgabe an das angegebene Terminal oder den angegebenen Drucker gesendet. Existiert das angegebene Terminal oder der angegebene Drucker auch nicht, wird die Session abgebrochen.</p> <p>Das Standard-Terminal-Ausgabeformat für asynchrone Sessions ist ein 3270-Datenstrom. Wenn die <code>SENDER</code>-Terminalangabe kein 3270-Gerät ist, muss die Natural-Anwendung auf den richtigen</p>

Plattform	Kommentar
	<p>Terminaltyp umschalten, bevor das erste Ausgabe-Statement ausgeführt wird (zum Beispiel durch Angabe von <code>SET CONTROL 'T=PRNT'</code> für einen Drucker oder durch Starten mit dem Profilparameter <code>TTYPE=PRNT</code>).</p> <p>Wenn Sie alle Ausgaben an einen (Spool-)Bestimmungsort, z.B. <code>CSSL</code>, senden, muss die Natural-Anwendung auf Line-Modus schalten, z.B. durch Angabe von <code>SET CONTROL 'T=xxxx'</code> oder durch Starten mit dem Profilparameter <code>TTYPE=xxxx</code>, wobei für <code>xxxx</code> der Wert <code>BTCH</code> oder <code>ASYL</code> angegeben wird. In diesem Fall werden zwei weitere Profilparameter relevant: <code>EJ</code> und <code>INTENS</code>.</p> <ul style="list-style-type: none"> <li>■ Wenn Sie <code>EJ=ON</code> setzen, werden alle Line-Ausgaben mit einem vorangestellten ASA-Steuerzeichen weitergeleitet.</li> <li>■ Bei <code>EJ=OFF</code> gibt es kein vorangestelltes ASA-Steuerzeichen.</li> <li>■ <code>INTENS</code> sollte auf 1 gesetzt werden, insbesondere wenn Sie <code>EJ=OFF</code> gesetzt haben.</li> </ul> <p>Informationen zu weiteren CICS-spezifischen Funktionen siehe <i>Asynchronous Natural Processing under CICS</i> in der <i>TP Monitor Interfaces</i>-Dokumentation.</p>
Com-plete	Siehe <i>Asynchronous Natural Processing under Com-plete/SMARTS</i> in der TP-Monitor-Dokumentation.
IMS/TM	Der Profilparameter <code>SENDER</code> gibt den Standard-LTERM an. Dieser LTERM wird immer benutzt, wenn kein Drucker angegeben wurde. Sie sollten immer einen <code>SENDER</code> -Parameter im <code>NIIBOOT</code> -Modul angeben. Dies ist wichtig, wenn Natural versucht, beim Start einer Session eine Fehlermeldung auszugeben. Wird kein <code>SENDER</code> -Parameter angegeben, gibt es keinen gültigen LTERM, und <code>NATIMS</code> beendet die Session.



**Anmerkung:** Weitere Informationen siehe siehe Profilparameter `OUTDEST` und *Asynchronous Processing* in der *Operations*-Dokumentation.

# 224 SF - 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.

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



## Anmerkungen:

1. In einer Natural-Session kann der Profilparameter SF durch den Session-Parameter SF überschrieben werden.
2. Unter Natural Security: Die Einstellung dieses Parameters kann durch die *Session Parameters*-Option des Library-Profiles überschrieben werden.
3. Siehe auch *Column Spacing - SF Parameter and nX Notation* im Leitfaden zur Programmierung.



# 225

## SG - Vorzeichen-Stelle

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

<b>Mögliche Werte</b>	ON	Eine Vorzeichenstelle wird belegt.
	OFF	Keine Vorzeichenstelle wird belegt.  <b>Anmerkung:</b>  1. Wenn SG=OFF gesetzt ist, werden negative Werte ohne das Minuszeichen (-) ausgegeben. 2. SG=OFF verhindert nicht die Eingabe von negativen Werten in Eingabefeldern.
<b>Standard-Einstellung</b>	ON	
<b>Angabe innerhalb der Session</b>	ja	
<b>Gültige Statements</b>	DISPLAY FORMAT INPUT PRINT WRITE	
<b>Gültiges Kommando</b>	keines	



### Anmerkungen:

1. Wenn der EM-Parameter (Editiermodus) angegeben wird, überschreibt er den SG-Parameter.
2. Siehe auch *Parameter zur Beeinflussung der Ausgabe von Feldern im Leitfaden zur Programmierung*.

**Beispiel:**

```
FORMAT SG=OFF
```

# 226

## SHAPED - Steuerung der Zeichenumformung

---

Dieser Natural Profil- und Session-Parameter bestimmt, ob in ihren Grundformen („unshaped/ungeformt“) vorliegende Basiszeichen in ihre Darstellungsformen („shaped/geformt“) umgewandelt werden, bevor sie gespeichert werden.

Die Umformung der Zeichen ist notwendig, um Zeichen einer bidirektionalen Sprache (z.B. Arabisch) bei Benutzung eines Browsers oder beim Editieren von Texten mit dem NaturalONE-Quellcode-Editor (Source Editor) korrekt darzustellen.

<b>Mögliche Werte</b>	ON	Nicht umgewandelte Zeichen („unshaped“) werden in ihre Darstellungsformen („geformed/shaped“) umgewandelt.
	OFF	Nicht umgewandelte Zeichen („unshaped“) werden nicht in ihre Darstellungsformen („geformed/shaped“) umgewandelt.
<b>Standard-Einstellung</b>	OFF	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	ja	

Siehe auch *Unterstützung bidirektionaler Sprachen* in der *Unicode- und Codepage-Unterstützung-Dokumentation*.



# 227

## SKEY - Speicherschutzschlüssel für

### Programmausführung

---

Dieser Natural-Profilparameter gilt nur bei z/OS-Plattformen.

SKEY bestimmt den Speicherschutzschlüssel für das TP-Monitor-Interface (Com-plete oder CICS), unter dem Ntaural läuft.

<b>Mögliche Werte</b>	ON	Natural benutzt den Speicherschutzschlüssel des TP-Monitors, unter dem das Natural-Programm ausgeführt wird: <ul style="list-style-type: none"><li>■ Unter Com-plete: Natural benutzt den Speicherschutzschlüssel 8, um Speicher für die Programmausführung zu erhalten.</li><li>■ Unter CICS: Natural benutzt den CICS-Schlüssel auch dann, wenn EXECKEY(USER) in CICS definiert ist.</li></ul> <p>EXECKEY(USER) mit SKEY=ON hat nicht die gleiche Wirkung wie EXECKEY(CICS), insbesondere wenn die CICS-Transaktionsisolation aktiv ist.</p>
	OFF	Natural benutzt einen anderen Speicherschutzschlüssel als der TP-Monitor, unter dem das Natural-Programm ausgeführt wird.  Unter CICS benutzt Natural den in CICS definierten Ausführungsschlüssel (EXECKEY).  Wir empfehlen diese Einstellung, damit Sie die Vorteile der CICS-Speicherschutzseinrichtung maximal nutzen können.
<b>Standard-Einstellung</b>	OFF	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	



**Anmerkungen:**

1. Einzelheiten zu Speicherschutzschlüsseln siehe Complete-Dokumentation.
2. Einzelheiten zur CICS-Schlüsselausführung siehe entsprechende IBM-Dokumentation.

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.

<b>Mögliche Werte</b>	20 - 250	<p><b>Im Batch-Betrieb:</b></p> <p>Die Anzahl der in jeder Zeile in den Dateien CMSYNIN und CMOBJIN zu verarbeitenden Zeichen.</p> <p><b>Anmerkung:</b> Einzelheiten zu diesen Dateien siehe betriebssystemspezifische Teile des Abschnitts <i>Natural in Batch Mode</i> in der <i>Operations</i>-Dokumentation.</p> <p><b>Im Online-Betrieb:</b></p> <p>Die Anzahl der zu interpretierenden Zeichen, wenn Sie den Natural-Programm-Editor im EDT-Modus benutzen (der mit dem Systemkommando EDT aktiviert wurde).</p>
<b>Standard-Einstellung</b>	72	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	ja	
<b>Gültige Statements</b>	keine	
<b>Gültiges Kommando</b>	GLOBALS	
<b>Application programming interface</b>	USR1005N	Siehe <i>SYSEXT Utility</i> in der Dokumentation <i>Debugger und Dienstprogramme</i> .



#### Anmerkungen:

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

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

# 229

## SLOCK - Source-Sperrung

Dieser Natural-Profilparameter gibt an, wie zeitgleiche Änderung an Natural Source-Objekten zu behandeln sind (siehe auch *Sperrmechanismen für Source-Objekte* in der *Editoren-Dokumentation*).

<b>Mögliche Werte</b>	PRE	<p>Aktiviert die Sperrung von Source-Objekten, die entweder lokal oder in einer SPoD-Umgebung oder mit Natural ISPF oder in gemischten Umgebungen editiert werden.</p> <p><b>Anmerkung:</b></p> <ol style="list-style-type: none"> <li>1. Dies ist die empfohlene Einstellung für das Arbeiten in gemischten Umgebungen.</li> <li>2. Um ein Source Member gegen zeitgleiche Aktualisierungen zu sperren, wird ein spezifischer Datensatz in die Natural-Systemdatei FUSER oder FNAT (je nachdem, wo sich das zu editierende Source Member befindet) geschrieben.</li> </ol>
	SPOD	<p>Die Sperrung von Source-Objekten erfolgt nur in einer Remote-Entwicklungsumgebung auf der Basis von Natural Single Point of Development (SPoD).</p> <p><b>Anmerkung:</b></p> <ol style="list-style-type: none"> <li>1. Diese Einstellung sorgt für Kompatibilität mit früheren Natural-Versionen, die die Sperrung unter SPoD unterstützten.</li> <li>2. Um ein Source Member gegen zeitgleiche Aktualisierungen zu sperren, wird ein spezifischer Datensatz in die als Development Server File bezeichnete Natural-Systemdatei (entspricht dem FDIC) geschrieben.</li> <li>3. Für ein DDM wird der gesperrte Datensatz in die Natural-Systemdatei FDIC geschrieben.</li> </ol>
	POST	<p>Wenn SLOCK=POST gesetzt wird, kann das Source-Objekt, das momentan editiert wird, in den Editor-Arbeitsbereich gelesen und von mehreren Benutzern geändert</p>

		<p>werden. Es kann aber nur der Benutzer, der eine Änderung zuerst speichert, das Source-Objekt aktualisieren.</p> <p><b>Anmerkung:</b></p> <ol style="list-style-type: none"> <li>1. Dies geschieht durch den Vergleich der Zeitmarke des in der Datenbank gespeicherten Source-Objekts mit der Zeitmarke des Source-Objekts beim Einlesen in den Editor-Arbeitsbereich.</li> <li>2. Alle anderen Benutzer erhalten eine entsprechende Fehlermeldung, wenn sie die Source zu speichern versuchen.</li> <li>3. Dieses Konzept ist nicht kompatibel mit dem SPoD-Sperrkonzept früherer Natural-Versionen.</li> </ol>
	OFF	Deaktiviert alle Sperrmechanismen.
<b>Standard-Einstellung</b>	SPOD	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	

# 230

## SM - Programmierung im Structured Mode

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

<b>Mögliche Werte</b>	ON	Die Programmierung darf nur in Structured Mode-Syntax erfolgen.
	OFF	Die Programmierung kann entweder in Structured Mode oder in Reporting Mode erfolgen.
<b>Standard-Einstellung</b>	OFF	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	ja	
<b>Gültige Statements</b>	keine	
<b>Gültiges Kommando</b>	GLOBALS	
<b>Programmierschnittstelle (API)</b>	USR1005N	Siehe <i>SYSEXT Utility</i> in der Dokumentation <i>Debugger und Dienstprogramme</i> .



### Anmerkungen:

1. Falls mit dem Profilparameter `SM` festgelegt wird, dass im Structured Mode programmiert werden muss `SM=ON`, wird ein Versuch, diese Einstellung während der Session per Systemkommando `GLOBALS` und Session-Parameter `SM` zu ändern mit einer Fehlermeldung zurückgewiesen (Reporting mode not permitted).
2. In einer Natural-Session kann der Profilparameter `SM=OFF` durch den Session-Parameter `SM=ON` überschrieben werden.
3. 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).
4. Falls Natural Security installiert ist, besteht die Möglichkeit, dass dieser Parameter im Security-Profil einer Library deaktiviert ist; ist dies der Fall, so gilt für die Library unabänderlich Structured Mode.



# 231

## SORT - Steuerung des Sortierprogramms

---

▪ SORT-Parameter-Syntax .....	702
▪ NTSORT-Makro-Syntax .....	702
▪ Schlüsselwort-Subparameter .....	703
▪ Beispiel für SORT-Parameter .....	706
▪ Beispiele für NTSORT-Macros .....	707

Dieser Natural-Profilparameter dient zur Steuerung des Sortierprogramms bei SORT-Statements. Es entspricht dem Makro `NTSORT` im Natural-Parametermodul.

<b>Mögliche Werte</b>	Siehe <a href="#">SORT Parameter-Syntax</a> und <a href="#">Schlüsselwort-Subparameter</a> .	
<b>Standard-Einstellung</b>	Siehe <a href="#">Schlüsselwort-Subparameter</a> .	
<b>Dynamische Angabe</b>	ja	Dieser Parameter kann nur dynamisch angegeben werden. Im Natural-Parametermodul wird stattdessen das Makro <code>NTSORT</code> verwendet.
<b>Angabe innerhalb der Session</b>	nein	



**Anmerkungen:**

1. Mit den Schlüsselwort-Subparametern von `SORT` oder `NTSORT` können Sie verschiedene Optionen angeben, die die Handhabung des Sortierprogramms steuern, wenn ein `SORT`-Statement ausgeführt wird.
2. Das zu verwendende Sortierprogramm kann entweder das Natural-interne Programm (Standard-Einstellung für alle Umgebungen) oder ein externes Sortierprogramm sein. Die Art der Sortierung ist abhängig von der Einstellung des Schlüsselwort-Subparameters `EXT`.

## SORT-Parameter-Syntax

Der Profilparameter `SORT` wird wie folgt angegeben:

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

Namen und Werte der Schlüsselwort-Subparameter siehe [Schlüsselwort-Subparameter](#).

## NTSORT-Makro-Syntax

Das Makro `NTSORT` wird wie folgt angegeben:

```
NTSORT EXT=value, *
      EXTEOJ=value, *
      EXTNAME=value, *
      EXTOPT=value,value,..., *
      STORAGE=value, *
      WRKSIZE=value
```

Namen und Werte der Schlüsselwort-Subparameter siehe [Schlüsselwort-Subparameter](#).

## Schlüsselwort-Subparameter

EXT | EXTEOJ | EXTNAME | EXTOPT | STORAGE | WRKSIZE

### EXT - Verwendung eines externen Sortierprogramms

EXT=*value* gibt an, ob ein externes Sortierprogramm verwendet werden soll oder nicht.

Wert	Erklärung
ON	Ein externes Sortierprogramm wird verwendet. Dies ist nur in Batch-Modus-Umgebungen einschließlich IMS/BMP und TSO möglich.
OFF	Das Natural-Sortierprogramm wird verwendet. Dies ist der Standardwert.

### EXTEOJ - Maßnahme im Fehlerfall

EXTEOJ=*value* gibt die Maßnahme an, die durchgeführt werden soll, wenn bei der Ausführung des externen Sortierprogramms ein Fehler festgestellt wird.

Wert	Erklärung
ON	Falls ein Fehler festgestellt wird, wird die SORT-Verarbeitung beendet. Die Einstellung ON erfordert, dass das verwendete Sortierprogramm in der Lage ist, einen Return Code 16 von den beiden SORT-Exit-Routinen E15 und E35 festzustellen.
OFF	Falls ein Fehler festgestellt wird, hält Natural weitere Aufrufe an das Sortierprogramm zurück und ignoriert jeden Datensatz, der an die SORT-Exit-Routine E35 übergeben wird. Dies ist die Standard-Einstellung.

### EXTNAME - Name des externen Sortierprogramms

EXTNAME=*value* gibt den Namen (1 bis 8 Zeichen) des zu verwendenden externen Sortierprogramms an. Der Standard-Name ist SORT.

Wert	Erklärung
1 bis 8 Zeichen.	Name des externen Sortierprogramms.
SORT	Dies ist der Standardwert.

## EXTOPT - Zusätzliche Optionen für externes Sortierprogramm

`EXTOPT=(value,value,...)` gibt zusätzliche Optionen für das externe Sortierprogramm an.

Natural generiert die nötigen Feld- und Format-Parameter und übergibt sie an das externe Sortierprogramm. Mit `EXTOPT` können Sie zusätzliche Parameter angeben, die an das externe Sortierprogramm übergeben werden sollen. Sie können nur Parameter angeben, die Bestandteil der Steueranweisungssyntax Ihres externen Sortierprogramms sind.

Sie können bis zu zwei Optionszeichenketten angeben, die mit einem Schrägstrich (/) abgegrenzt werden. Die erste Optionszeichenkette wird an die `SORT`-Steueranweisung angehängt, die zweite Optionszeichenkette wird verwendet, um eine `OPTION`-Steueranweisung zu erstellen. Sie können die Optionszeichenkette vor bzw. nach dem Schrägstrich weglassen. Wenn Sie die Optionszeichenkette nach dem Schrägstrich weglassen, wird überhaupt keine `OPTION`-Steueranweisung erzeugt.

Sie müssen die gesamte Optionszeichenkette in Hochkommas ('...') stellen. Aus Kompatibilitätsgründen besteht immer noch die Möglichkeit, die Optionszeichenkette stattdessen in Klammern zu setzen.

Aus Kompatibilitätsgründen wird eine einzelne Zeichenkette ohne voran- oder nachgestellten Schrägstrich abweichend behandelt. Bei z/OS werden die Optionen an die `SORT`-Steueranweisung angehängt:

### EXTOPT-Beispiel:

Die zusätzlichen Parameter können wie folgt angegeben werden:

```
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/'
```

## STORAGE - Typ des Speichermediums

`STORAGE=value` gibt den Typ des Speichermediums an, das vom Natural-internen Sortierprogramm benutzt werden soll.

Wert	Erklärung
MAIN	Nur der verbleibende <code>WRKSIZE</code> -Speicher wird benutzt; es steht kein sonstiger Zwischenspeicher zur Verfügung.  Dies ist der Standardwert.
BP	Der Sortier-Buffer Pool wird als Zwischenspeicher benutzt. Siehe <a href="#">Hinweise</a> weiter unten.

Wert	Erklärung
SD	SD-Dateien werden als Zwischenspeicher benutzt. Dieser Wert wird nur unter Com-plete berücksichtigt.
SMARTS	Das SMARTS Portable File System wird benutzt.



**Anmerkung:** Generell versucht das SORT-Statement zunächst, den restlichen Speicher in der WRKSIZE zu benutzen. Falls dieser Speicher für die Anzahl der Datensätze nicht ausreicht, versucht das SORT-Statement zusätzlich Zwischenspeicher für die Verarbeitung der Datensätze zu benutzen.

### Hinweise zu SD-Dateien unter Com-plete/SMARTS

- Die für die Sortierverarbeitung benutzten SD-Dateien werden alle temporäre SD-Dateien für eine Stack-Ebene zugeordnet. Die Namenssyntax der Sortier-SD-Dateien lautet daher wie folgt:

```
&&STsnnn
```

Dabei ist:

Syntax-Element	Beschreibung
&&	Indikator für eine temporäre SD-Datei.
ST	Standard-Präfix der für die Sortierverarbeitung benutzten Sortier-SD-Datei.
s	Stack-Ebene.
nnn	Folgenummer innerhalb eines einzelnen Sortierlaufs.

- SMARTS-Arbeitsdateien befinden sich im SMARTS Portable File System. Der Pfad muss mit der SMARTS-Umgebungsvariablen \$NAT\_WORK\_ROOT angegeben werden. Für eine zur Sortierverarbeitung benutzte SMARTS-Arbeitsdatei wird ein spezielles Verzeichnis mit dem Namen SORT hinzugefügt, und für jeden Benutzer wird ein getrenntes, mit der Benutzerkennung benanntes Verzeichnis angelegt. Das resultierende Verzeichnis ist dann \$NAT\_WORK\_ROOT/SORT/userid. Der Name der zur Sortierverarbeitung benutzten Arbeitsdatei entspricht dem Namen der temporären SD-Datei unter Com-plete.

### Hinweis zur Benutzung des Sortier-Buffer Pool

Die Verwendung eines Sortier-Buffer Pool ist nur dann sinnvoll, wenn Sie den WRKSIZE-Wert nicht mehr weiter vergrößern können, um die Sortierdatensätze zu speichern. Dies gilt typischerweise für Online-Umgebungen, in denen die Speicher-Thread-Größe begrenzt ist. In allen anderen Fällen brauchen Sie nur einen ausreichend großen WRKSIZE-Wert anzugeben.

Wenn Sie einen Sortier-Buffer Pool benutzen wollen, setzen Sie den Schlüsselwort-Subparameter STORAGE=BP, um anzugeben, dass ein Sortier-Buffer Pool als zusätzlicher Speicher zu dem mit WRKSIZE definierten Puffer benutzt werden soll. Außerdem müssen Sie mit dem Profilparameter BPI bzw. dem Makro NTBPI einen Buffer Pool vom Typ TYPE=Sort mit dem NAME=name in Natural

anmelden. Beispiel: `BPI=(TYPE=SORT,NAME=XYZ)`. Wenn mit dem Subparameter `NAME` des Profilparameters `BPI` ein Name angegeben wird, dann bezieht sich diese Angabe auf einen globalen Sortier-Buffer Pool; ein lokaler Sortier-Buffer Pool kann dagegen mit `NAME=' '` (leer) angegeben werden.

### WRKSIZE - Größe des vom Sortierprogramm benutzten Arbeitspuffers

`WRKSIZE=value` gibt die Größe des vom Sortierprogramm benutzten Arbeitspuffers an.

Wert	Erklärung
0 oder	Größe des Arbeitspuffers (in KB).
10 - 2097151	Bei <code>WRKSIZE=0</code> können keine Sortieroperationen durchgeführt werden.
10	Dies ist der Standardwert.



**Anmerkung:** Der mit `WRKSIZE` angegebene Arbeitspuffer dient zur Aufnahme interner Sortiersteuerdaten. Der restliche Speicher wird benutzt, um die Datensätze zu sammeln und zu sortieren. Die Größe der Sortiersteuerdaten ist von verschiedenen Faktoren abhängig (`WRKSIZE` selbst, Datensatzlänge, Anzahl der Sortierschlüssel, deren Größe und Format) und kann daher nicht formelhaft berechnet werden.

## Beispiel für SORT-Parameter

---

### Beispiel 1:

```
SORT=(EXT=OFF,WRKSIZE=1024)
```

Für die Sortierverarbeitung wird das Natural-interne Sortierprogramm und ein Arbeitspufferspeicher von 1 MB benutzt.

### Beispiel 2:

```
SORT=(EXT=ON,EXTOPT='/EQUALS')
```

Für die Sortierverarbeitung wird ein externes Sortierprogramm mit einer [Steueranweisung](#) `OPTION EQUALS` benutzt.

## Beispiele für NTSORT-Macros

---

```
NTSORT EXT=OFF, *  
        WRKSIZE=1024
```

```
NTSORT EXT=ON, *  
        EXTOPT=' /EQUALS '
```

Im ersten Beispiel wird für die Sortierverarbeitung das Natural-interne Sortierprogramm und ein Arbeitspufferspeicher von 1 MB benutzt. Im zweiten Beispiel wird für die Sortierverarbeitung ein externes Sortierprogramm mit einer **Steueranweisung** `OPTION EQUALS` benutzt.



# 232

## SOSI - Shift-Out/Shift-In-Codes für

### Doppel-Byte-Zeichensätze

---

▪ SOSI-Parameter-Syntax .....	710
▪ Positionsgebundene Subparameter .....	710
▪ Konvertierung von logischen Shift-Out/Shift-In Zeichen .....	711
▪ Automatische Anpassung von Umsetzungstabellen .....	711
▪ Kompatibilität des SOSI-Profilparameters mit den veralteten SO- und SI-Profilparametern .....	712
▪ SOSI-Parameter-Beispiel .....	712

Dieser Natural-Profilparameter ist für asiatische Länder relevant, die Doppel-Byte-Zeichensätze (DBCS) verwenden.

Wenn der Profilparameter CP auf eine Multi-Byte Codepage (MBCS) gesetzt wird, werden die logischen Shift-In- und Shift-Out-Zeichen mit der Codepage geliefert, darum wird dann der Profilparameter SOSI ignoriert.

<b>Mögliche Werte</b>	Siehe <i>Positionsgebundene Subparameter</i> weiter unten.	
<b>Standard-Einstellung</b>	keine	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	

## SOSI-Parameter-Syntax

---

Der Profilparameter SOSI wird wie folgt angegeben:

```
SOSI=(logical-shift-out,[physical-shift-out],logical-shift-in,[physical-shift-in],[SO/SI-display-length]
```

Der Shift-Out-Code dient dazu, den Punkt anzugeben, an dem der Code der Zeichendarstellung vom Normal-Modus (Einzel-Byte) in den Doppel-Byte-Modus geschaltet wird.

Der Shift-In-Code dient dazu, den Punkt anzugeben, an dem der Code der Zeichendarstellung vom Doppel-Byte-Modus zurück in den Normal-Modus (Einzel-Byte) geschaltet wird.

## Positionsgebundene Subparameter

---

Folgende positionsgebundene Subparameter sind vorhanden:

Subparameter	Erklärung
<i>logical-shift-out</i>	Das logische Shift-Out-Zeichen muss ein einzelnes Zeichen sein. Geben Sie die Hexadezimal-Darstellung des logischen Shift-Out-Zeichens an.  Üblicherweise wird für IBM-Hardware der Wert 0E benutzt.
<i>physical-shift-out</i>	Der Wert für das physische Shift-Out-Zeichen muss entsprechend der verwendeten Hardware gewählt werden.  Die Länge für das physische Shift-Out-Zeichen kann 1 oder 2 Bytes sein. Geben Sie die Hexadezimal-Darstellung des physischen Shift-Out-Zeichens an.  Der Standardwert ist das logische Shift-Out-Zeichen.

Subparameter	Erklärung
<i>logical-shift-in</i>	Das logische Shift-In-Zeichen muss ein einzelnes Zeichen sein. Geben Sie die Hexadezimal-Darstellung des logischen Shift-Out-Zeichens an.  Üblicherweise wird für IBM-Hardware der Wert 0F benutzt.
<i>physical-shift-in</i>	Der Wert für das physische Shift-In-Zeichen muss entsprechend der verwendeten Hardware gewählt werden.  Die Länge für das physische Shift-In-Zeichen kann 1 oder 2 Bytes sein. Geben Sie die Hexadezimal-Darstellung des physischen Shift-In-Zeichens an.  Der Standardwert ist das logische Shift-In-Zeichen.
<i>SO/SI-display-length</i>	Die Anzahl an Bytes, die vom physischen Shift-Out-/Shift-In-Zeichen auf dem Bildschirm belegt wird.  Mögliche Werte sind 0 und 1. Der Standardwert ist 1.  Für IBM-Hardware muss der Wert 1 benutzt werden.

## Konvertierung von logischen Shift-Out/Shift-In Zeichen

Logische Shift-Out/Shift-In Zeichen werden in das entsprechende physische Shift-Out/Shift-In Zeichen umgesetzt, bevor die Daten zum Bildschirm übertragen werden.

Physische Shift-Out/Shift-In Zeichen werden in das entsprechende logische Shift-Out/Shift-In Zeichen umgesetzt, bevor die am Bildschirm eingegebenen Daten an die Natural-Anwendung übertragen werden.

## Automatische Anpassung von Umsetzungstabellen

Bei abgeschalteter Codepage-Unterstützung (d.h., der Profilparameter **CP** ist auf **OFF** gesetzt) werden die Einträge für die logischen Shift-Out/Shift-In-Zeichen in den Umsetzungstabellen aktualisiert, die von den folgenden Makros und Profilparametern zur Verfügung gestellt werden:

Tabelle	Makro	Profilparameter
Standardmäßig verwendete (primäre) Ausgabe-Umsetzungstabelle	<b>NTTAB</b>	<b>TAB</b>
Alternative (sekundäre) Ausgabe-Umsetzungstabelle	<b>NTTAB1</b>	<b>TAB1</b>
Sekundäre Eingabe-Umsetzungstabelle, die benutzt wird, wenn der Session-Parameter <b>PM</b> auf <b>C</b> gesetzt ist.	<b>NTTAB2</b>	<b>TAB2</b>
SYS*-Ausgabe-Umsetzungstabelle	<b>NTTABL</b>	<b>TABL</b>

Wenn die Zeichen, in die die logischen Shift-Out/Shift-In-Zeichen umgesetzt werden sollen, beim Start von Natural noch ihren Standardwert ("?" = X'6F') in der entsprechenden Umsetzungstabelle haben (d.h., sie sind nicht durch eines der oben aufgeführten Makros bzw. Profilparameter geändert worden), dann werden sie so aktualisiert, dass logische Shift-Out/Shift-In-Zeichen nicht für die Eingabe und Ausgabe umgesetzt werden.

Ausführliche Informationen zu den Umsetzungstabellen siehe *Translation Tables* in der *Operations*-Dokumentation.

## Kompatibilität des SOSI-Profilparameters mit den veralteten SO- und SI-Profilparametern

---

Der Subparameter *logical-shift-out* entspricht dem Profilparameter *S0* und der Subparameter *logical-shift-in* entspricht dem Profilparameter *SI*.

Die veralteten Profilparameter *S0* und *SI* sind immer noch gültig, dürfen aber nur sich gegenseitig ausschließend mit dem Profilparameter *SOSI* benutzt werden. Wir empfehlen Ihnen dringend, den Profilparameter *SOSI* anstelle der Profilparameter *S0* und *SI* zu benutzen.

Die Angabe *S0=xx, SI=yy* ist gleichbedeutend mit der Angabe *SOSI=(xx, xx, yy, yy, 1)*.

## SOSI-Parameter-Beispiel

---

Für IBM-Hardware sollten Sie *SOSI=(0E, 0E, 0F, 0F, 1)* benutzen, was gleichbedeutend ist mit *SOSI=(0E, , 0F, , 1)*.

# 233

## SRETAIN - Source-Format beibehalten

Dieser Natural-Profilparameter gibt das Kodierformat für neue und existierende Natural-Source-Programme an, wenn diese gespeichert werden.

<b>Mögliche Werte</b>	ON	<p>Die ursprüngliche Codepage einer existierenden Natural-Source wird beibehalten.</p> <p>Wenn eine existierende Natural-Source ohne Codepage-Informationen gespeichert wird, erhält sie keine Codepage-Information.</p> <p>Wenn einen neue Natural-Source angelegt wird, wird sie in dem Standard-Codepage-Format gespeichert, das mit mit dem Profilparameter <b>CP</b> definiert ist.</p>
	OFF	Natural-Source-Programme werden im Standard-Codepage-Format gespeichert.
	(ON, EXCEPTNEW)	<p>Die ursprüngliche Codepage einer existierenden Natural-Source wird beibehalten.</p> <p>Wenn eine existierende Natural-Source ohne Codepage-Information gespeichert wird, erhält sie keine Codepage-Informationen.</p> <p>Wenn einen neue Natural-Source angelegt wird, wird sie ohne Codepage-Informationen gespeichert.</p> <p>(ON, EXCEPTNEW) sichert die Kompatibilität mit existierenden Anwendungen, die mit früheren Natural-Versionen erstellt wurden, bei denen es noch keine Codepage-Unterstützung gab.</p> <p><b>Anmerkung:</b> Der Wert (ON, EXCEPTNEW) wird nur auf Großrechnern unterstützt.</p>
<b>Standard-Einstellung</b>	ON	
<b>Dynamische Angabe</b>	ja	

<b>Angabe innerhalb der Session</b>	nein	
-------------------------------------	------	--



**Anmerkungen:**

1. Für die Codepage einer Natural-Source, die mit `SAVE` oder `STOW` gespeichert wird, hängt die resultierende Kodierung von den Einstellungen der Profilparameter `SRETAIN` and `CP` ab. Siehe *Codepage-Unterstützung bei Editoren, Systemkommandos und Utilities* in der *Unicode- und Codepage-Unterstützung-Dokumentation*.
2. Siehe auch *Natural-Profilparameter und Parameter-Makros* in der *Unicode- und Codepage-Unterstützung-Dokumentation*.

# 234

## SSIZE - Größe des Software AG Editor-Arbeitsbereichs

Dieser Natural-Profilparameter bestimmt die Größe des Puffers, der vom Software AG Editor benutzt wird.

<b>Mögliche Werte</b>	40 - 512	Puffergröße in KB.
	0	<b>Anmerkung:</b> Wenn SSIZE=0 angegeben wird oder wenn der angeforderte Speicher nicht verfügbar ist, kann der Software AG Editor nicht benutzt werden.
<b>Standard-Einstellung</b>	64	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	



### Anmerkungen:

1. Alternativ können Sie den äquivalenten Natural-Profilparameter **DS** oder das Parametermakro **NTDS** benutzen.
2. Wenn Sie eine Editor-Arbeitsdatei mit einer Datensatzlänge von mehr als 4 KB (Standardwert) definiert haben, sollten Sie einen SSIZE-Wert > 64 KB benutzen. Es gibt zwei Arbeitsdatei-Datensatzpuffer, die innerhalb von SSIZE zugeordnet werden. Deshalb sollten Sie zweimal Ihre Arbeitsdatei-Datensatzpuffergröße minus 4 KB zu Ihrem SSIZE hinzufügen. Beispiel: Ihre Editor-Arbeitsdatei hat eine Datensatzlänge von 10 KB. Dann verwenden Sie mindestens SSIZE=76 (das heißt  $64+2*(10-4)$ ).
3. Weitere Informationen zur Software AG Editor-Arbeitsdatei siehe *Operating the Software AG Editor, Editor Work File* in der *Operations*-Dokumentation.



# 235

## STACK - Daten/Kommandos auf den Natural Stack

### stellen

---

Dieser Natural-Profilparameter dient dazu, Daten oder Kommandos auf den Natural Stack zu stellen.



#### Anmerkung:

<b>Mögliche Werte</b>	Beliebige Zeichenkette	Siehe weiter unten.
<b>Standard-Einstellung</b>	HELLO	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	



#### Anmerkungen:

1. Wenn `STACK` benutzt wird, muss ein Doppelpunkt (`:`) bei den Profil- (oder Session-)Parametern `DC`, `HI`, `IA`, `ID` und `STACKD` angegeben werden.
2. Der Natural-Stack kann eine Folge von Natural-Kommandos und/oder benutzerdefinierten Kommandos zusammen mit ihren Daten zur Ausführung zu Beginn der Natural-Session enthalten.
3. Der Kommando-Stack wird verarbeitet, bevor der Benutzer am Bildschirm zur Eingabe aufgefordert wird (TP-Modus) oder Daten von den Dateien `CMSYNIN/CMOBJIN` gelesen werden; siehe *Natural in Batch Mode* in der *Operations*-Dokumentation.
4. Wenn während der Stack-Verarbeitung ein `INPUT`-Statement angetroffen wird, wird der entsprechende Eingabebildschirm nur dann erzeugt, wenn die erforderlichen Eingabedaten nicht schon beim Anlegen des Stack mit dem Kommando mitgeliefert wurden.
5. Auf jedes Systemkommando oder benutzerdefinierte Kommando können optional Daten folgen, die benutzt werden, um die Anforderung von Informationen zu erfüllen, die während der Kommandoausführung benötigt werden. Die Zeichenkette, die als Daten mit dem

STACK-Parameter mitgegeben werden, müssen in Klammern gesetzt werden. Wenn es sich bei dem Kommando um ein benutzerdefiniertes Kommando handelt, d.h. um den Namen eines Benutzerprogramms, erfüllen die mitgelieferten Daten die Datenerfordernisse von INPUT-Statements im Benutzerprogramm.

### Konventionen

- Mehrere Einstellungen für ein INPUT-Statement werden durch ein Komma (,) voneinander getrennt.
- Daten für mehrere INPUT-Statements werden durch einen Doppelpunkt (:) voneinander getrennt.
- Kommandos werden durch das Stack-Abgrenzungszeichen voneinander getrennt, dass mit dem Profilparameter `STACKD` definiert ist. Die Standard-Einstellung ist ein Semikolon (;).

### Beispiele:

```
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. ↵
```

Da einige Kommandos (z.B. `GLOBALS`) keine Parameter durch `INPUT` lesen, sollte statt eines Doppelpunkts (:) ein Leerzeichen benutzt werden, um ein Kommando vom ersten Parameterdatenbestandteil abzugrenzen.

```
STACK='LOGON SYSTEM'
```

Da der Makro-Assembler keine eingeschlossenen Leerzeichen innerhalb von Klammern zulässt, muss die Zeichenkette in Hochkommas gesetzt werden, wenn sie als statischer Parameter verwendet werden soll.

Dieser Natural-Profilparameter gibt das Zeichen an, das als Kommando-Abgrenzungszeichen für den STACK-Parameter und für die Kommandoingabe unter dem Natural Development Server (Produktcode: NDV) in einer Natural Single Point of Development-Umgebung verwendet werden soll.

<b>Mögliche Werte</b>	Beliebiges Zeichen	Zeichen, das als Kommando-Abgrenzungszeichen verwendet werden soll.  <b>Anmerkung:</b> Das Zeichen darf nicht identisch mit dem Zeichen sein, das mit den Profil-/Session-Parametern <b>ID</b> (Input-Delimiterzeichen), <b>DC</b> (Dezimalstellenzeichen) oder <b>IA</b> (Input-Zuweisungszeichen) angegeben ist.
<b>Standard-Einstellung</b>	; (Semikolon)	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	



#### Anmerkungen:

1. Um zu vermeiden, dass der für den STACK-Parameter angegebene Wert oder die als Kommandoingabe unter dem NDV zu übergebenden Daten nicht wie beabsichtigt interpretiert werden, sollte der Wert des STACK-Parameters auf ein Zeichen gesetzt werden, das nicht in den übergebenen Daten enthalten ist, falls die Daten den Standardwert des Stack-Abgrenzungszeichen (siehe Beispiel oben) enthalten. Die STACK-Parameter-Einstellung sollte auf ein Zeichen geändert werden, das nicht mit dem Standardwert übereinstimmt, wenn der ID-Parameter auf ein Semikolon (;) gesetzt worden ist. Aus Gründen der Abwärtskompatibilität gilt diese Einschränkung nicht bei STACKD=; (Standard-Einstellung).
2. Das angegebene Zeichen kann in Hochkommas gesetzt werden.
3. Wenn das Input-Zuweisungszeichen ein Komma sein soll, muss es als ID=' , ' angegeben werden, weil das Komma zum Abtrennen einzelner Parameter dient.

### Beispiel:

```
STACKD='/',ID=';' STACK=(DUMP IOB;+100/FIN)
```

Um zu vermeiden, dass das Semikolon nach `DUMP IOB` als Befehlsabgrenzungszeichen interpretiert wird, wird `STACKD` auf `'/'` gesetzt.

# 237

## STEPLIB - Zusätzliche Steplib Library

---

Dieser Natural-Profilparameter gibt den Namen einer zusätzlichen Natural Steplib (verkettete Library) an, die benutzt werden soll.

<b>Mögliche Werte</b>	1 - 8 Zeichen	Steplib-Name.
<b>Standard-Einstellung</b>	SYSTEM	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	
<b>Anwendungsprogrammierschnittstelle</b>	USR1005N	Siehe <i>SYSEXT Utility</i> in der Dokumentation <i>Debugger und Dienstprogramme</i> .



**Anmerkung:** Weitere Informationen siehe *Steplib Libraries* und *Suchreihenfolge bei der Objekt-Ausführung* in der Dokumentation *Natural benutzen*.



# 238

## SUBSID - Subsystemkennung unter z/OS

---

Dieser Natural-Profilparameter gibt das zu verwendende Natural-Subsystem an.

<b>Mögliche Werte</b>	1 - 4 Zeichen	Natural-Subsystem. <b>Anmerkung:</b> Wenn Sie weniger als 4 Zeichen angeben, werden Leerzeichen angehängt, um so eine 4-Byte-Einstellung zu erhalten.
<b>Standard-Einstellung</b>	NAT8	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	



### Anmerkungen:

1. Dieser Natural-Profilparameter steht nur unter z/OS zur Verfügung.
2. Für die Zwecke des Natural CICS Interface (siehe `ROLLSRV`, `SIPSERV`, `SUBSID`) wird der Natural-Profilparameter `SUBSID` ignoriert, wenn er in einer Parameterkette mittels Profilparameter `SYS` oder `PROFILE` in einem alternativen Parametermodul (angegeben mit dem Profilparameter `PARM`) angegeben wird.
3. Informationen zum Natural-Subsystem siehe *Natural Subsystem under z/OS* in der *Operations-Dokumentation*.



# 239

## SYNERR - Übergabe von Syntaxfehlern

---

Dieser Natural-Profilparameter gibt an, ob Syntaxfehler an das Fehlertransaktionsprogramm übergeben werden oder nicht.

<b>Mögliche Werte</b>	ON	Syntaxfehler werden an das Fehlertransaktionsprogramm übergeben.
	OFF	Es erfolgt keine Syntaxfehlerübergabe an das Fehlertransaktionsprogramm.
<b>Standard-Einstellung</b>	OFF	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	
<b>Anwendungsprogrammierschnittstelle</b>	USR4007N	Siehe <i>SYSEXT Utility</i> in der Dokumentation <i>Debugger und Dienstprogramme</i> .



### Anmerkungen:

1. Das Fehlertransaktionsprogramm wird entweder mit dem Profilparameter [ETA](#) oder innerhalb des Natural Security-Library-Profiles definiert.
2. Weitere Informationen siehe *Verwendung eines Fehlertransaktionsprogramms* im Leitfaden zur Programmierung.



# 240

## SYS - Satz dynamischer Profilparameter definieren und aktivieren

---

- SYS-Parameter-Syntax ..... 728
- NTSYS-Makro-Syntax ..... 729
- Beispiele für NTSYS-Makros ..... 730

Mit diesem Natural-Profilparameter können Sie einen Satz dynamischer Profilparameter aktivieren, der im Natural-Parametermodul vordefiniert wird. Damit können Sie die wiederholte Angabe langer Folgen von Profilparametern beim Natural-Session-Start vermeiden.

<b>Mögliche Werte</b>	Siehe <a href="#">SYS-Parameter-Syntax</a> .	
<b>Standard-Einstellung</b>	keine	
<b>Dynamische Angabe</b>	ja	Dieser Parameter kann nur dynamisch angegeben werden.
<b>Angabe innerhalb der Session</b>	nein	



**Anmerkungen:**

1. Der angegebene Parametersatz muss im zurzeit aktiven Natural-Parametermodul, z.B. in einem alternativen Parametermodul, definiert werden, wenn er mit dem [PARM](#)-Parameter vor dem [SYS](#)-Parameter angegeben wird.
2. Der Parametersatz wird exakt so anstelle von [SYS](#) im Parameter-Zeichensatz ausgewertet, als ob Sie stattdessen den definierten Parametersatz an dieser Stelle angegeben hätten.
3. Als Alternative zum Profilparameter [SYS](#) können Sie den Profilparameter [PROFILE](#) benutzen, der einen ähnlichen Funktionsumfang bietet.
4. Im Natural-Parametermodul benutzen Sie [NTSYS](#)-Makros, um Sätze dynamischer Profilparameter vorzudefinieren. Zur Identifizierung geben Sie dem Parametersatz einen eindeutigen Satznamen.

Dieses Dokument behandelt folgende Themen:

## SYS-Parameter-Syntax

---

Die Syntax des Parameters [SYS](#) lautet wie folgt:

```
SYS=set-name
```

Dabei ist:

Syntax-Element	Erläuterung
<i>set-name</i>	Der Satzname, definiert mit einem <a href="#">NTSYS</a> -Makro im Natural-Parametermodul. Der Satzname kennzeichnet den nachfolgenden Parametersatz. Er kann 1 bis 8 Zeichen lang sein und muss mit einem alphabetischen Zeichen beginnen.

## NTSYS-Makro-Syntax

Das Makro NTSYS wird für jeden Parametersatz wie folgt angegeben:

```
NTSYS set-name, 'parameter-string'
```

Wenn der Parametersatz mehr als 255 Zeichen enthält, müssen Sie die Makro-Zeile(n) umbrechen oder mehrere Parameterzeichenketten definieren:

```
NTSYS set-name, 'parameter-string', *
      'parameter-string', *
      'parameter-string'
```

Um mehrere Parametersätze zu definieren, müssen Sie für jeden Satz ein eigenes NTSYS-Makro benutzen:

```
NTSYS set-name1, 'parameter-string'
NTSYS set-name2, 'parameter-string'
```

Dabei ist:

Syntax-Element	Erläuterung
<i>set-name</i>	Der Satzname kennzeichnet den nachfolgenden Parametersatz. Er kann 1 bis 8 Zeichen lang sein und muss mit einem alphabetischen Zeichen beginnen.
<i>parameter-string</i>	Nach dem Satznamen ( <i>set-name</i> ) geben Sie einzelne Profilparameter und ihre Werte an.



### Anmerkungen:

1. Der gesamte Parametersatz, den Sie mit einem NTSYS-Makro angeben, muss eine gültige Kette dynamischer Parameter darstellen. Die Parameterkette wird beim Assemblieren des Natural-Parametermoduls nicht auf Gültigkeit geprüft.
2. Alle Parameterketten eines NTSYS-Makros werden zu einem Parametersatz verkettet.
3. Ein Hochkomma in einer Teilzeichenkette wird durch zwei Hochkommas dargestellt.

## Beispiele für NTSYS-Makros

---

```
NTSYS SET1, 'FUSER=(,50),LC=ON,NC=ON,ULANG=2,          *  
           CMPO=(TQMARK=OFF),STACK=(LOGON ULIB1)'  
NTSYS SET2, 'FUSER=(,51),ULANG=4,WH=ON,KC=ON,          *  
           STACK=(LOGON ULIB2)'
```

# 241 SYSCIP - Adabas-Standard-Chiffrierschlüssel für

## Natural-Systemdateien

---

Dieser Natural-Profilparameter definiert einen Adabas-Standard-Chiffrierschlüssel für den Zugriff auf Natural-Systemdateien (FNAT, FUSER, FDIC, FSEC, FSPPOOL, FPROF, FREG), die mit der Ciphred-Option geladen worden sind.

<b>Mögliche Werte</b>	1 - 8 Zeichen	Der Adabas-Standard-Chiffrierschlüssel für den Zugriff auf Natural-Systemdateien.  <b>Anmerkung:</b>  1. Der mit dem Parameter SYSCIP angegebene Chiffrierschlüssel gilt für alle Natural-Systemdateien, für die keine eigenen Chiffrierschlüssel angegeben sind.  2. Wenn die Natural-Systemdateien nicht verschlüsselt sind, setzen Sie SYSCIP auf Leerzeichen.
	Leerzeichen	
<b>Standard-Einstellung</b>	Leerzeichen	
<b>Dynamische Angabe</b>	ja	<b>Anmerkung:</b> Wenn Sie den Parameter SYSCIP dynamisch in Verbindung mit einem der individuellen Systemdatei-Parametern FNAT, FUSER, FDIC, FSEC, FSPPOOL, FPROF und FREG angeben, müssen Sie den SYSCIP-Parameter <i>vor</i> einem der individuellen Systemdate-Parameter angeben.
<b>Angabe innerhalb der Session</b>	nein	



### Anmerkungen:

1. Dieser Natural-Profilparameter gilt nur bei Adabas-Datenbanken.
2. Chiffrierschlüssel für einzelne Systemdateien können mit dem Subparameter *cipher-key* der Profilparameter FNAT, FUSER, FDIC, FSEC, FSPPOOL, FPROF und FREG angegeben werden.



# 242

## SYSPSW - Adabas-Standard-Passwort für

### Natural-Systemdateien

---

Dieser Natural-Profilparameter gibt ein Adabas-Standard-Passwort für den Zugriff auf Natural-Systemdateien (FNAT, FUSER, FDIC, FSEC, FPROF, FSPPOOL, FREG) an, die mit einem Passwort geschützt worden sind.

<b>Mögliche Werte</b>	1 - 8 Zeichen	Das Adabas-Standard-Passwort für passwortgeschützte Natural-Systemdateien.  <b>Anmerkung:</b>  1. Wenn eine Natural-Systemdatei passwortgeschützt ist, muss ein Passwort angegeben werden, welches das Aktualisieren der Datei gestattet.  2. Das mit dem Parameter SYSPSW angegebene Passwort gilt für alle Natural-Systemdateien, für die keinen individuellen Passwörter angegeben sind.  3. Wenn der Profilparameter OPRB angegeben wird, dann wird das mit SYSPSW angegebene Passwort für den ersten Adabas-Open-Aufruf benutzt und muss den Zugriff und/oder die Aktualisierung aller in OPRB angegebenen Dateien wie erforderlich gestatten.
	Leerzeichen	<b>Anmerkung:</b> Wenn die Natural-Systemdateien nicht passwortgeschützt sind, setzen Sie SYSPSW auf Leerzeichen.
<b>Standard-Einstellung</b>	Leerzeichen	
<b>Dynamische Angabe</b>	ja	<b>Anmerkung:</b> Wenn Sie den Profilparameter SYSPSW dynamisch in Verbindung mit einem der individuellen Systemdatei-Profilparametern <b>FNAT</b> , <b>FUSER</b> , <b>FDIC</b> , <b>FSEC</b> , <b>FSPPOOL</b> , <b>FPROF</b> und <b>FREG</b> angeben, müssen Sie den Profilparameter SYSPSW <i>vor</i> einem der individuellen Systemdatei-Parameter angeben.

Angabe innerhalb der Session	nein	
------------------------------	------	--



**Anmerkungen:**

1. Dieser Natural-Profilparameter gilt nur bei Adabas-Datenbanken.
2. Passwörter für einzelne Systemdateien können mit dem Subparameter *password* der Profilparameter **FNAT**, **FUSER**, **FDIC**, **FSEC**, **FSPool**, **FPROF** und **FREG** angegeben werden.

# 243

## TAB - Standard-Ausgabezeichen-Umsetzung

---

- TAB-Parameter-Syntax ..... 736
- NTTAB-Makro-Syntax ..... 737
- Beispiel für TAB-Parameter ..... 737
- Beispiel für NTTAB-Makro ..... 737

Mit diesem Natural-Profilparameter können Sie die Definitionen in der Umsetzungstabelle `NTTAB` überschreiben, die im Konfigurationsmodul `NATCONFIG` enthalten sind. `TAB` entspricht dem Makro `NTTAB` im Natural-Parametermodul.

<b>Mögliche Werte</b>	Siehe <i>TAB Parameter-Syntax</i> weiter unten.	
<b>Standard-Einstellung</b>	Gemäß den Angaben im Makro <code>NTTAB</code> in <code>NATCONFIG</code> .	
<b>Dynamische Angabe</b>	ja	Dieser Parameter kann nur dynamisch angegeben werden. Im Natural-Parametermodul wird stattdessen das Makro <code>NTTAB</code> benutzt.
<b>Angabe innerhalb der Session</b>	nein	



**Anmerkungen:**

1. Die Tabelle `NTTAB` ist die Standard-Ausgabezeichen-Umsetzungstabelle.
2. Wenn der Profilparameter `CP` auf einen anderen Wert als `OFF` gesetzt ist, werden die mit `TAB` angegebenen Werte ignoriert. Siehe auch *Umsetzungstabellen (Translation Tables)* in der *Unicode- und Codepage-Unterstützung-Dokumentation*.

## TAB-Parameter-Syntax

---

Der Profilparameter `TAB` wird wie folgt angegeben:

```
TAB=(a1,a2,b1,b2,c1,c2,...)
```

Sie geben Zeichenpaare an, wobei das erste Zeichen eines Paares das umzusetzende Zeichen und das zweite Zeichen eines Paares das Zeichen ist, in welches das erste Zeichen umgesetzt werden soll.

Sie können jedes Zeichen entweder als das 1-Byte-Zeichen selbst (in Hochkommas gesetzt) oder als Hexadezimaldarstellung dieses Zeichens angeben.

Oder:

```
TAB=OFF
```

Bei `TAB=OFF` werden alle (statischen und dynamischen) Definitionen auf die Werte zurückgesetzt, die im Makro `NTTAB` im Modul `NATCONFIG` angegeben sind.

## NTTAB-Makro-Syntax

---

Das Makro `NTTAB` wird wie folgt angegeben:

```
NTTAB a1,a2,b1,b2,c1,c2,...
```



### Anmerkungen:

1. Erklärung der Syntax-Elemente siehe [TAB-Parameter-Syntax](#).
2. Der Wert `OFF` kann nicht mit dem Makro `NTTAB`, sondern nur dynamisch mit dem Profilparameter `TAB` angegeben werden.

## Beispiel für TAB-Parameter

---

Beim Parameter `TAB` müssen Sie die gesamte Zeichenkette mit Zeichenpaaren in Klammern setzen, zum Beispiel:

```
TAB=(5E,'Ä','ö',78,FF,00,'ü','Ü')
```

## Beispiel für NTTAB-Makro

---

```
NTTAB 5E,'Ä','ö',78,FF,00,'ü','Ü'
```

In diesem Beispiel wird das durch `H'5E'` dargestellte Zeichen in `'Ä'` umgesetzt, `'ö'` in das durch `H'78'` dargestellte Zeichen, das durch `H'FF'` dargestellte Zeichen in das durch `H'00'` dargestellte Zeichen und `'ü'` in `'Ü'`.



# 244

## TAB1 - Alternative Ausgabezeichen-Umsetzung

---

▪ TAB1-Parameter-Syntax .....	740
▪ NTTAB1-Makro-Syntax .....	741
▪ Beispiel für TAB1-Parameter .....	741
▪ Beispiel für NTTAB1-Makro .....	741

Mit diesem Natural-Profilparameter können Sie die Definitionen in der Umsetzungstabelle `NTTAB1` überschreiben, die im Konfigurationsmodul `NATCONFIG` enthalten ist. `TAB1` entspricht dem Makro `NTTAB1` im Natural-Parametermodul.

<b>Mögliche Werte</b>	Siehe <i>TAB1-Parameter-Syntax</i> weiter unten.	
<b>Standard-Einstellung</b>	Gemäß den Angaben im Makro <code>NTTAB1</code> in <code>NATCONFIG</code> .	
<b>Dynamische Angabe</b>	ja	Dieser Parameter kann nur dynamisch angegeben werden. Im Natural-Parametermodul wird stattdessen das Makro <code>NTTAB1</code> benutzt.
<b>Angabe innerhalb der Session</b>	nein	



**Anmerkung:** Die Tabelle `NTTAB1` in `NATCONFIG` ist die alternative Ausgabezeichen-Umsetzungstabelle für den sekundären Zeichensatz, der benutzt wird, wenn der Profil-/Session-Parameter `PM` auf `=C` gesetzt ist.

## TAB1-Parameter-Syntax

---

Der Profilparameter `TAB1` wird wie folgt angegeben:

```
TAB1=(a1,a2,b1,b2,c1,c2,...)
```

Sie geben Zeichenpaare an, wobei das erste Zeichen eines Paares das umzusetzende Zeichen und das zweite Zeichen eines Paares das Zeichen ist, in welches das erste Zeichen umgesetzt werden soll.

Sie können jedes Zeichen entweder als das 1-Byte-Zeichen selbst (in Hochkommas gesetzt) oder als 2-Byte-Hexadezimaldarstellung dieses Zeichens angeben.

Oder:

```
TAB1=OFF
```

Bei `TAB1=OFF` werden alle (statischen und dynamischen) Definitionen auf die Werte zurückgesetzt, die im Makro `NTTAB1` im Modul `NATCONFIG` angegeben sind.

## NTTAB1-Makro-Syntax

---

The NTTAB1 wird wie folgt angegeben:

```
NTTAB1 a1,a2,b1,b2,c1,c2,...
```



### Anmerkungen:

1. Erklärung der Syntax-Elemente siehe [TAB1-Parameter-Syntax](#).
2. Der Wert OFF kann nicht mit dem Makro NTTAB1, sondern nur dynamisch mit dem Profilparameter TAB1 angegeben werden.

## Beispiel für TAB1-Parameter

---

Beim Parameter TAB1 müssen Sie die gesamte Zeichenkette mit Zeichenpaaren in Klammern setzen, zum Beispiel:

```
TAB1=(5E,'Ä','ö',78,FF,00,'ü','Ü')
```

## Beispiel für NTTAB1-Makro

---

```
NTTAB1 5E,'Ä','ö',78,FF,00,'ü','Ü'
```

In diesem Beispiel wird das durch H'5E' dargestellte Zeichen in 'Ä' umgesetzt, 'ö' in das durch H'78' dargestellte Zeichen, das durch H'FF' dargestellte Zeichen in das durch H'00' dargestellte Zeichen und 'ü' in 'Ü'.



# 245

## TAB2 - Alternative Eingabezeichen-Umsetzung

---

- TAB2-Parameter-Syntax ..... 744
- NTTAB2-Makro-Syntax ..... 745
- Beispiel für TAB2-Parameter ..... 745
- Beispiel für NTTAB2-Makro ..... 745

Mit diesem Natural-Profilparameter können Sie die Definitionen in der Umsetzungstabelle `NTTAB2` überschreiben, die im Konfigurationsmodul `NATCONFIG` enthalten ist. `TAB2` entspricht dem Makro `NTTAB2` im Natural-Parametermodul.

<b>Mögliche Werte</b>	Siehe <i>TAB2 Parameter-Syntax</i> weiter unten.	
<b>Standard-Einstellung</b>	Gemäß den Angaben im Makro <code>NTTAB2</code> in <code>NATCONFIG</code> .	
<b>Dynamische Angabe</b>	ja	Dieser Parameter kann nur dynamisch angegeben werden. Im Natural-Parametermodul wird stattdessen das Makro <code>NTTAB2</code> benutzt.
<b>Angabe innerhalb der Session</b>	nein	



**Anmerkung:** Die Tabelle `NTTAB2` ist die alternative Eingabezeichen-Umsetzungstabelle für den sekundären Zeichensatz, der benutzt wird, wenn der Profil-/Session-Parameter `PM` auf `=C` gesetzt ist.

## TAB2-Parameter-Syntax

---

Der Profilparameter `TAB2` wird wie folgt angegeben:

```
TAB2=(a1,a2,b1,b2,c1,c2,...)
```

Sie geben Zeichenpaare an, wobei das erste Zeichen eines Paares das umzusetzende Zeichen und das zweite Zeichen eines Paares das Zeichen ist, in welches das erste Zeichen umgesetzt werden soll.

Sie können jedes Zeichen entweder als das 1-Byte-Zeichen selbst (in Hochkommas gesetzt) oder als 2-Byte-Hexadezimaldarstellung dieses Zeichens angeben.

Oder:

```
TAB2=OFF
```

Bei `TAB2=OFF` werden alle (statischen und dynamischen) Definitionen auf die Werte zurückgesetzt, die im Makro `NTTAB2` im Modul `NATCONFIG` angegeben sind.

## NTTAB2-Makro-Syntax

---

Das Makro NTTAB2 wird wie folgt angegeben:

```
NTTAB2 a1,a2,b1,b2,c1,c2,...
```



### Anmerkungen:

1. Erklärung der Syntax-Elemente siehe [TAB2-Parameter-Syntax](#).
2. Der Wert OFF kann nicht mit dem Makro NTTAB2, sondern nur dynamisch mit dem Profilparameter TAB2 angegeben werden.

## Beispiel für TAB2-Parameter

---

Beim Parameter TAB2 müssen Sie die gesamte Zeichenkette mit Zeichenpaaren in Klammern setzen, zum Beispiel:

```
TAB2=(5E,'Ä','ö',78,FF,00,'ü','Ü')
```

## Beispiel für NTTAB2-Makro

---

```
NTTAB2 5E,'Ä','ö',78,FF,00,'ü','Ü'
```

In diesem Beispiel wird das durch H'5E' dargestellte Zeichen in 'Ä' umgesetzt, 'ö' in das durch H'78' dargestellte Zeichen, das durch H'FF' dargestellte Zeichen in das durch H'00' dargestellte Zeichen und 'ü' in 'Ü'.



# 246

## TABA1 - EBCDIC/ASCII-Zeichenumsetzung

---

▪ TABA1-Parameter-Syntax .....	748
▪ NTTABA1-Makro-Syntax .....	748
▪ Beispiel für TABA1-Parameter .....	749
▪ Beispiel für NTTABA1-Makro .....	749

Mit diesem Natural-Profilparameter können Sie die Definitionen in der EBCDIC/ASCII-Zeichenumsetzungstabelle NTTABA1 überschreiben, die im Konfigurationsmodul NATCONFIG enthalten ist. TABA1 entspricht dem Makro NTTABA1 im Natural-Parametermodul.

<b>Mögliche Werte</b>	Siehe <i>TABA1 Parameter-Syntax</i> weiter unten.	
<b>Standard-Einstellung</b>	Gemäß den Angaben im Makro NTTABA1 in NATCONFIG.	
<b>Dynamische Angabe</b>	ja	Dieser Parameter kann nur dynamisch angegeben werden. Im Natural-Parametermodul wird stattdessen das Makro NTTABA1 benutzt.
<b>Angabe innerhalb der Session</b>	nein	

## TABA1-Parameter-Syntax

---

Der Profilparameter TABA1 wird wie folgt angegeben:

```
TABA1=(a1,a2,b1,b2,c1,c2,...)
```

Sie geben Zeichenpaare an, wobei das erste Zeichen eines Paares das umzusetzende Zeichen und das zweite Zeichen eines Paares das Zeichen ist, in welches das erste Zeichen umgesetzt werden soll.

Sie können jedes Zeichen entweder als das 1-Byte-Zeichen selbst (in Hochkommas gesetzt) oder als 2-Byte-Hexadezimaldarstellung dieses Zeichens angeben.

Oder:

```
TABA1=OFF
```

Bei TABA1=OFF werden alle (statischen und dynamischen) Definitionen auf die Werte zurückgesetzt, die im Makro NTTABA1 im Modul NATCONFIG angegeben sind.

## NTTABA1-Makro-Syntax

---

Das Makro NTTABA1 wird wie folgt angegeben:

```
NTTABA1 a1,a2,b1,b2,c1,c2,...
```



#### Anmerkungen:

1. Erklärung der Syntax-Elemente siehe [TABA1-Parameter-Syntax](#).
2. Der Wert OFF kann nicht mit dem Makro NTTABA1, sondern nur dynamisch mit dem Profilparameter TABA1 angegeben werden.

## Beispiel für TABA1-Parameter

---

Beim Parameter TABA1 müssen Sie die gesamte Zeichenkette mit Zeichenpaaren in Klammern setzen, zum Beispiel:

```
TABA1=(5E,'Ä','ö',78,FF,00,'ü','Ü')
```

## Beispiel für NTTABA1-Makro

---

```
NTTABA1 5E,'Ä','ö',78,FF,00,'ü','Ü'
```

In diesem Beispiel wird das durch H'5E' dargestellte Zeichen in 'Ä' umgesetzt, 'ö' in das durch H'78' dargestellte Zeichen, das durch H'FF' dargestellte Zeichen in das durch H'00' dargestellte Zeichen und 'ü' in 'Ü'.



# 247

## TABA2 - ASCII/EBCDIC-Zeichenumsetzung

---

▪ TABA2-Parameter-Syntax .....	752
▪ NTTABA2-Makro-Syntax .....	752
▪ Beispiel für NTTABA2-Makro .....	753
▪ Beispiel für TABA2-Parameter .....	753

Mit diesem Natural-Profilparameter können Sie die Definitionen in der ASCII/EBCDIC-Zeichenumsetzungstabelle NTTABA2 überschreiben, die im Konfigurationsmodul NATCONFIG enthalten ist. TABA2 entspricht dem Makro NTTABA2 im Natural-Parametermodul.

<b>Mögliche Werte</b>	Siehe <a href="#">TABA2 Parameter-Syntax</a> weiter unten.	
<b>Standard-Einstellung</b>	Gemäß den Angaben im Makro NTTABA2 in NATCONFIG.	
<b>Dynamische Angabe</b>	ja	Dieser Parameter kann nur dynamisch angegeben werden. Im Natural-Parametermodul wird stattdessen das Makro NTTABA2 benutzt.
<b>Angabe innerhalb der Session</b>	nein	

## TABA2-Parameter-Syntax

Der Profilparameter TABA2 wird wie folgt angegeben:

$$\text{TABA2} = \left\{ \begin{array}{l} (a1,a2,b1,b2,c1,c2,\dots) \\ \text{OFF} \end{array} \right\}$$

Sie geben Zeichenpaare an, wobei das erste Zeichen eines Paares das umzusetzende Zeichen und das zweite Zeichen eines Paares das Zeichen ist, in welches das erste Zeichen umgesetzt werden soll.

Sie können jedes Zeichen entweder als das 1-Byte-Zeichen selbst (in Hochkommas gesetzt) oder als 2-Byte-Hexadezimaldarstellung dieses Zeichens angeben.

Bei TABA2=OFF werden alle (statischen und dynamischen) Definitionen auf die Werte zurückgesetzt, die im Makro NTTABA2 im Modul NATCONFIG angegeben sind.

## NTTABA2-Makro-Syntax

Das Makro NTTABA2 wird wie folgt angegeben:

```
NTTABA2 a1,a2,b1,b2,c1,c2,...
```



### Anmerkungen:

1. Erklärung der Syntax-Elemente siehe [TABA2-Parameter-Syntax](#).
2. Der Wert OFF kann nicht mit dem Makro NTTABA2, sondern nur dynamisch mit dem Profilparameter TABA2 angegeben werden.

## Beispiel für NTTABA2-Makro

---

```
NTTABA2 5E,'Ä','ö',78,FF,00,'ü','Ü'
```

In diesem Beispiel wird das durch H'5E' dargestellte Zeichen in 'Ä' umgesetzt, 'ö' in das durch H'78' dargestellte Zeichen, das durch H'FF' dargestellte Zeichen in das durch H'00' dargestellte Zeichen und 'ü' in 'Ü'.

## Beispiel für TABA2-Parameter

---

Beim Parameter TABA2 müssen Sie die gesamte Zeichenkette mit Zeichenpaaren in Klammern setzen, zum Beispiel:

```
TABA2=(5E,'Ä','ö',78,FF,00,'ü','Ü')
```



# 248

## TABL - SYS-Library-Ausgabe-Zeichenumsetzung

---

▪ TABL-Parameter-Syntax .....	756
▪ NTTABL-Makro-Syntax .....	757
▪ Beispiel für TABL-Parameter .....	757
▪ Beispiel für NTTABL-Makro .....	757

Mit diesem Natural-Profilparameter können Sie die Definitionen in der Umsetzungstabelle `NTTABL` überschreiben, die im Konfigurationsmodul `NATCONFIG` enthalten ist. `TABL` entspricht dem Makro `NTTABL` im Natural-Parametermodul.

<b>Mögliche Werte</b>	Siehe <i>TABL Parameter-Syntax</i> weiter unten.	
<b>Standard-Einstellung</b>	Gemäß den Angaben im Makro <code>NTTABL</code> in <code>NATCONFIG</code> .	
<b>Dynamische Angabe</b>	ja	Dieser Parameter kann nur dynamisch angegeben werden. Im Natural-Parametermodul wird stattdessen das Makro <code>NTTABL</code> benutzt.
<b>Angabe innerhalb der Session</b>	nein	



**Anmerkung:** Die Tabelle `NTTABL` wird benutzt, um die Ausgabe der Programme umzusetzen, die sich in Libraries befinden, deren Namen mit `SYS...` beginnen.

## TABL-Parameter-Syntax

---

Der Profilparameter `TABL` wird wie folgt angegeben:

```
TABL=(a1,a2,b1,b2,c1,c2,...)
```

Sie geben Zeichenpaare an, wobei das erste Zeichen eines Paares das umzusetzende Zeichen und das zweite Zeichen eines Paares das Zeichen ist, in welches das erste Zeichen umgesetzt werden soll.

Sie können jedes Zeichen entweder als das 1-Byte-Zeichen selbst (in Hochkommas gesetzt) oder als 2-Byte-Hexadezimaldarstellung dieses Zeichens angeben.

Oder:

```
TABL=OFF
```

Bei `TABL=OFF` werden alle (statischen und dynamischen) Definitionen auf die Werte zurückgesetzt, die im Makro `NTTABL` im Modul `NATCONFIG` angegeben sind.

## NTTABL-Makro-Syntax

---

Das Makro `NTTABL` wird wie folgt angegeben:

```
NTTABL a1,a2,b1,b2,c1,c2,...
```



### Anmerkungen:

1. Erklärung der Syntax-Elemente siehe [TABL-Parameter-Syntax](#).
2. Der Wert `OFF` kann nicht mit dem Makro `NTTABL`, sondern nur dynamisch mit dem Profilparameter `TABL` angegeben werden.

## Beispiel für TABL-Parameter

---

Beim Parameter `TABL` müssen Sie die gesamte Zeichenkette mit Zeichenpaaren in Klammern setzen, zum Beispiel:

```
TABL=(5E,'Ä','ö',78,FF,00,'ü','Ü')
```

## Beispiel für NTTABL-Makro

---

```
NTTABL 5E,'Ä','ö',78,FF,00,'ü','Ü'
```

In diesem Beispiel wird das durch `H'5E'` dargestellte Zeichen in `'Ä'` umgesetzt, `'ö'` in das durch `H'78'` dargestellte Zeichen, das durch `H'FF'` dargestellte Zeichen in das durch `H'00'` dargestellte Zeichen und `'ü'` in `'Ü'`.



# 249

## TC - Nachgezogene Zeichen

Die mit dem Session-Parameter TC 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.

<b>Mögliche Werte</b>	beliebige(s) Zeichen	Sie können eine Zeichenkette mit bis zu 10 Zeichen definieren.
<b>Standard-Einstellung</b>	Keine	
<b>Angabe innerhalb der Session</b>	ja	
<b>Gültige Statements</b>	FORMAT	Parameter kann dynamisch mit dem FORMAT-Statement angegeben werden.
	DISPLAY	Parameter kann auf Statement- und/oder Element-Ebene angegeben werden.
<b>Gültiges Kommando</b>	keines	



### Anmerkungen:

1. 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.
2. Der Parameter TC kann auch für Felder des Formats U benutzt werden. Weitere Informationen zum Unicode-Format entnehmen Sie dem Abschnitt *Unicode- und Codepage-Unterstützung in der Natural-Programmiersprache, Session-Parameter, Abschnitt EMU, ICU, LCU, TCU versus EM, IC, LC, TC*.
3. Siehe auch *Parameter zur Beeinflussung der Ausgabe von Feldern im Leitfaden zur Programmierung*.

**Beispiele:**

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

# 250

## TCU - Nachgezogene Zeichen (Unicode)

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.

<b>Mögliche Werte</b>	beliebige(s) Zeichen	Sie können eine Zeichenkette mit bis zu 10 Zeichen definieren.
<b>Standard-Einstellung</b>	Keine	
<b>Angabe innerhalb der Session</b>	ja	
<b>Gültige Statements</b>	FORMAT	Parameter kann dynamisch mit dem FORMAT-Statement angegeben werden.
	DISPLAY	Parameter kann auf Statement- und/oder Element-Ebene angegeben werden.
<b>Gültiges Kommando</b>	keines	



### Anmerkungen:

1. 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.
2. Der Session-Parameter TCU ist identisch mit dem Session-Parameter TC. Der Unterschied ist, dass die nachgezogenen Zeichen (Trailing Characters) immer im Unicode-Format gespeichert werden. Dadurch können Sie die nachgestellten Zeichen mit gemischten Zeichen aus unterschiedlichen Codepages eingeben. Es ist sichergestellt, dass immer das korrekte Zeichen angezeigt wird, unabhängig von der installierten System-Codepage.

Siehe auch:

- *Parameter zur Beeinflussung der Ausgabe von Feldern im Leitfaden zur Programmierung*

- *Unicode- und Codepage-Unterstützung in der Natural-Programmiersprache, Session-Parameter, Abschnitt EMU, ICU, LCU, TCU im Vergleich zu EM, IC, LC, TC.*

# 251

## TD - Zeitdifferenz

Dieser Natural-Profilparameter gibt eine Zeitdifferenz an, die sich auf die Natural-Zeit-/Datumseinstellung auswirken soll, um sicherzustellen, dass anstelle der Zeit-/Datumswerte des Rechenzentrums die aktuellen lokalen Zeit-/Datumswerte verwendet werden.

Mögliche Werte	AUTO	<p>Während der Session-Initialisierung vergleicht Natural die physische Maschinenzeit (Store Clock) und die logische Maschinenzeit (Systemumgebung) und benutzt die Differenz zwischen den beiden als Einstellung für den TD-Parameter.</p> <p><b>Anmerkung:</b></p> <ol style="list-style-type: none"><li>1. Damit eine Zeitänderung für Natural wirksam wird (z.B. Wechsel nach Sommerzeit oder zurück zur Winterzeit), genügt es deshalb, die logische Maschinenzeit zurückzusetzen.</li><li>2. Dies betrifft nur diejenigen Sessions, die nach der Zeitänderung gestartet wurden.</li></ol>
	+/- hh (+/- hh, mm) (+/- hh, mm, ss)	<p>Stunden, Minuten und Sekunden von (-23, 59, 59) bis (+23, 59, 59). Ein Plus-Zeichen (optional) oder Minus-Zeichen gibt an, ob der TD-Wert addiert oder subtrahiert werden soll.</p> <p><b>Anmerkung:</b> Die angegebene Zeit wird zu der oder von der physischen Maschinenzeit addiert oder subtrahiert, um die Uhrzeit und das Datum zu setzen, das von Natural verwendet werden soll.</p>
	1 - 32 Zeichen	<p>Name der zu verwendenden Zeitzone.</p> <p><b>Anmerkung:</b></p>

		<ol style="list-style-type: none"> <li>1. Diese Zeitzone muss als gültige Zeitzone im Makro NTTZ des NATCONFIG-Moduls definiert sein; siehe <i>Configuration Tables - Module NATCONFIG</i>.</li> <li>2. Ändert sich die Zeit gemäß der DST-Definition für die Zeitzone im NATCONFIG, wird dies während einer laufenden Session berücksichtigt.</li> </ol>
<b>Standard-Einstellung</b>	0	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	
<b>Anwendungsprogrammierschnittstelle</b>	USR1005N	Siehe <i>SYSEXT Utility</i> in der Dokumentation <i>Debugger und Dienstprogramme</i> .



**Anmerkung:** Dieser Parameter gilt in einer Umgebung, in der entfernt angeordnete Knoten (Remote Nodes) in einem Rechnernetzwerk benutzt werden.

### Beispiele

```

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)
    
```

# 252

## TF - Umsetzung der Datenbankkennung/Dateinummer

---

▪ TF-Parameter-Syntax .....	767
▪ NTTF-Makro-Syntax .....	767
▪ Beispiel für TF-Parameter .....	768
▪ Beispiel für NTTF-Makro .....	768

Dieser Natural-Profilparameter kann benutzt werden, um während der Ausführung einer Anwendung die Datenbankkennung (DBID)/Dateinummer (FNR) in eine andere Datenbankkennung/Dateinummer umzusetzen. Er entspricht dem Makro `NTTF` im Natural-Parametermodul.

<b>Mögliche Werte</b>	Siehe <i>TF-Parameter-Syntax</i> .	
<b>Standard-Einstellung</b>	keine	
<b>Dynamische Angabe</b>	ja	Dieser Parameter kann nur dynamisch angegeben werden. Im Natural-Parametermodul wird stattdessen das Makro <code>NTTF</code> benutzt.
<b>Angabe innerhalb der Session</b>	nein	
<b>Anwendungsprogrammierschnittstelle</b>	USR1034N USR2005N *	Siehe <i>SYSEXT Utility</i> in der Dokumentation <i>Debugger und Dienstprogramme</i> .  * Empfohlen.

**Anmerkungen:**

1. Dieser Parameter gilt nur für Benutzerdateien. Er gilt nicht für Systemdateien.
2. Dieser Parameter ist relevant, wenn eine Anwendung in einer Produktionsumgebung entwickelt wird. Er ermöglicht es Ihnen, eine Anwendung in einer Test-Datenbank zu entwickeln und dann die fertiggestellte Anwendung in die Produktionsdatenbank zu übertragen, ohne die Anwendung verändern oder neu katalogisieren zu müssen. Die Natural-Objekte werden mit Produktionsdatenbankkennung und -dateinummer (DBID/FNR) katalogisiert, aber immer dann, wenn ein Datenbankzugriff ausgeführt wird, wird die Produktions-DBID/FNR gemäß den Angaben im Parameter `TF` in die Test-DBID/FNR umgesetzt; d.h., es wird die Testdatenbank benutzt. Das bedeutet, dass der Test in der aktuellen Produktionsumgebung, jedoch ohne Produktionsdaten stattfinden kann.
3. Der Parameter `TF` bzw. das Makro `NTTF` kann mehrmals angegeben werden, so dass verschiedene Kombinationen von Dateinummern angegeben werden können.

Falls mehrere `TF`-Parameter angegeben werden, gilt Folgendes:

- Die Liste der `TF`-Definitionen wird in der Reihenfolge durchsucht, in der sie angegeben sind. Verwendet wird der erste Eintrag, der exakt bezüglich DBID und FNR (kein \*-Platzhalterzeichen) übereinstimmt.
- Wird keine Übereinstimmung gefunden, werden die `TF`-Definitionen ein zweites Mal durchsucht. Dieses Mal wird der erste passende \*-Platzhalterzeichen-Eintrag (entweder DBID oder FNR ist ein \*-Platzhalterzeichen) verwendet.

## TF-Parameter-Syntax

Der Profilparameter TF wird wie folgt angegeben:

```
TF=(production-dbid,production-fnr,test-dbid,test-fnr)
```

Dabei ist:

Syntax-Element	Wert	Erklärung
<i>production-dbid</i>	0 - 254 oder 256 - 65535 oder *	Datenbankkennung der Produktionsdatenbank. Ein Stern (*) kann als Platzhalterzeichen für alle Datenbankkennungen benutzt werden.  <b>Anmerkung:</b> Die Datenbankkennung 255 ist für logische Systemdateien für Software AG-Produkte reserviert, siehe Profilparameter <a href="#">LFILE</a> .
<i>production-fnr</i>	1 - 65535 oder *	Dateinummer der Produktionsdatenbank. Ein Stern (*) kann als Platzhalterzeichen für alle Dateinummern benutzt werden.
<i>test-dbid</i>	0 - 254 oder 256 - 65535 oder *	Datenbankkennung der Testdatenbank. Ein Stern (*) kann als Platzhalterzeichen benutzt werden, dann bleibt die Datenbankkennung unverändert.
<i>test-fnr</i>	1 - 65535 oder *	Dateinummer der Testdatenbank. Ein Stern (*) kann als Platzhalterzeichen benutzt werden, dann bleibt die Dateinummer unverändert.

## NTTF-Makro-Syntax

Das Makro NTTF wird wie folgt angegeben:

```
NTTF production-dbid,production-fnr,test-dbid,test-fnr
```



**Anmerkung:** Erklärung der Syntax-Elemente und der möglichen Werte siehe [TF-Parameter-Syntax](#).

## Beispiel für TF-Parameter

---

TF=(777,39,17,88),TF=(251,\*,9,\*)

## Beispiel für NTTF-Makro

---

Äquivalente Angabe im Natural-Parametermodul:

NTTF 777,39,17,88  
NTTF 251,\*,9,\*

# 253

## THPINIT - Name des vorinitialisierten

### Speicher-Thread-Modells

---

Dieser Natural-Profilparameter kann benutzt werden, um den Namen eines vorinitialisierten Speicher-Thread-Modells zu definieren und somit die Verarbeitungsleistung bei der Initialisierung der Natural-Session zu verbessern.

<b>Mögliche Werte</b>	1 - 8 Zeichen	Der Name eines vorinitialisierten Speicher-Thread-Modells, das bei der Initialisierung der Natural-Session benutzt werden soll.
<b>Standard-Einstellung</b>	' '	Bei der Initialisierung der Natural-Session wird kein vorinitialisiertes Speicher-Thread-Modell benutzt.
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	

Bei der statischen Initialisierung der Natural-Session erfolgt hauptsächlich eine Auswertung dynamischer Profilparameter und eine Pufferzuordnung/-Initialisierung, die CPU-intensiv sind und bei vielen Sitzungen oft auf genau die gleiche Weise wiederholt werden. Der Profilparameter THPINIT wird benutzt, um den Natural-Speicher-Thread-Inhalt nach der statischen Initialisierung der Sitzung zu speichern und diesen Inhalt als Speicher-Thread-Modell für andere Sitzungen wiederzuverwenden. Alle darauf folgenden Sitzungen mit derselben THPINIT-Definition überspringen dann die statische Initialisierung und laufen mit dem definierten vorinitialisierten Speicher-Thread-Modell.

THPINIT gilt nur bei Thread-Umgebungen. Wird der Profilparameter THSIZE benutzt, um einen Thread für Batch oder TSO zu definieren, kann THPINIT auch bei diesen Umgebungen benutzt werden.

Wird der Natural Roll Server benutzt, steht ein vorinitialisiertes Speicher-Thread-Modell nur bei der Umgebung (zum Beispiel CICS, IMS TM, Batch oder TSO) zur Verfügung, in der das Modell erzeugt wurde. Wird der Natural Roll Server nicht benutzt, wird ein vorinitialisiertes Speicher-Thread-Modell gespeichert und steht nur in der aktuellen Region zur Verfügung.

**Was bei der Benutzung des Profilparameters THPINIT zu bedenken ist:**

- Wenn ein vorinitialisiertes Speicher-Thread-Modell benutzt wird, werden nur die Profilparameter THPINIT und NUCNAME ausgewertet. Dies gilt sowohl bei statischen (Natural-Parametermodul) als auch bei dynamischen (beim Start der Session) Parameterdefinitionen. Somit werden keine Warnungen oder Fehlermeldungen zurückgegeben, falls irgendwelche Parameter ungültig oder fehlerhaft sind.
- Bei einer Session, die mit einem vorinitialisierten Speicher-Thread-Modell läuft, erfolgt keine Profilparameter-Protokollierung (PLOG=ON), und zwar auch dann nicht, wenn bei der Original-Session der Profilparameter PLOG angegeben und gespeichert war.
- Ein mit dem Profilparameter PROFILE (auf AUTO, PROGRAM oder TERMINAL gesetzt) angegebener generischer Wert wird als Wert bei der Session, für die das vorinitialisierte Speicher-Thread-Modell gespeichert wurde, zugewiesen, jedoch nicht als Wert für die aktuelle Session.
- Bei einer Session, die ein vorinitialisiertes Speicher-Thread-Modell benutzt, können statische und dynamische Parameterdefinitionen, die für die Original-Session (in der das Modell gespeichert wurde) gesetzt wurden, unter Verwendung der Natural User Exits USR4004N und USR8203N angezeigt werden.
- Ein gespeichertes vorinitialisiertes Speicher-Thread-Modell kann nicht geändert oder ersetzt werden. Die Utility-Funktion SYSTP R kann benutzt werden, um gespeicherte vorinitialisierte Thread-Session-Aufzeichnungen vom Natural Roll Server (falls vorhanden) anzuzeigen oder zu löschen. Das Präfix der Benutzerkennung für den Natural Roll Server ist \$THP.
- THPINIT-Definitionen in Parameterprofilen (Profilparameter PROFILE oder SYS) und alternative Natural-Parametermodule (Profilparameter PARM) werden ignoriert. THPINIT wird nur in der Anfangsphase der Session-Initialisierung ausgewertet, bevor Parameterprofile und alternative Parametermodule ausgewertet werden.
- Die physische Größe des Terminal-Bildschirms muss die gleiche sein wie die Bildschirmgröße bei dem vorinitialisierten Speicher-Thread-Modell.
- Die aktuelle Speicher-Thread-Größe muss größer oder gleich der Thread-Größe bei dem vorinitialisierten Speicher-Thread-Modell sein.

**Beispiel:**

```
THPINIT=MYTHREAD
```

# 254

## THSEPCH - 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.

<b>Mögliche Werte</b>	beliebiges Zeichen	Zur Laufzeit wird das dynamische Tausender-Trennzeichen durch dieses Zeichen ersetzt.  <b>Anmerkung:</b>  1. 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.  2. 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=' '' '.	
<b>Standard-Einstellung</b>	, (Komma)	<b>Anmerkung:</b> Standardmäßig wird als Tausender-Trennzeichen ein Komma verwendet.	
<b>Dynamische Angabe</b>	ja		
<b>Angabe innerhalb der Session</b>	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.*

# 255

## THSIZE - Größe des Speicher-Thread

---

Dieser Natural-Profilparameter kann benutzt werden, um die Größe des Speicher-Thread unter z/OS Batch oder TSO festzulegen, um so die Verarbeitungsleistung zu verbessern.

<b>Mögliche Werte</b>	256 - 2097151 oder 0	Thread-Größe in KB.
<b>Standard-Einstellung</b>	0	Es wird kein Thread verwendet.
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	



### Anmerkungen:

1. Dieser Natural-Profilparameter hat keinen Einfluss auf andere Umgebungen, in denen Threads schon verwendet werden, z.B. in Server-Umgebungen.
2. Der Speicher-Thread wird beim Session-Start zugewiesen. Natural versucht, alle Speicheranforderungen (GETMAIN/FREEMAIN) zu erfüllen, anstatt sie an das Betriebssystem weiterzuleiten. Der Thread muss groß genug sein, um alle Puffer mit fester Größe aufnehmen zu können. Wenn der Thread voll ist, können Puffer mit veränderbarer Größe außerhalb des Thread zugewiesen werden, vorausgesetzt die Einstellung des Profilparameters **OVSIZE** gestattet dies.
3. Der Profilparameter **THSIZE** wird nicht beachtet, wenn er in einer Parameterkette angegeben wird, die durch einen Profilparameter **SYS** oder **PROFILE** aktiviert wird, oder in einem alternativen Parametermodul (gemäß Angabe im Profilparameter **PARM**).
4. Wenn der Natural zIIP Enabler installiert und aktiv ist (nur bei z/OS Batch und TSO), kann bei Wahl einer geeigneten Einstellung des Profilparameters **THSIZE** durch Verringerung der Anzahl an physischen GETMAIN-Anweisungen die Anzahl der Umschaltungen in den TCB-Modus verringert werden. Das gilt ebenso, wenn der Profilparameter **WPSIZE** benutzt wird.
5. Hinsichtlich der Bedeutung des Profilparameters **THSIZE** im Zusammenhang mit Natural for zIIP siehe Abschnitt *Thread Size Specification - THSIZE Parameter* im Abschnitt *Tuning zIIP Usage* in der *Natural for zIIP* -Dokumentation.

**Beispiel:**

```
THSIZE=5000
```

Dieser Natural-Profilparameter steuert die IBM 3270-Terminal-Modellnummer für Online-Umgebungen, zum Beispiel unter IMS TM.

<b>Mögliche Werte</b>	0	Die Bildschirmgröße wird durch das umgebungsabhängige Treibermodul festgelegt. Wenn möglich, erhält es die Bildschirmgrößeninformation von seinem Subsystem. Sonst werden die Definitionen des Standard-Modells 2 verwendet, z.B. unter IMS TM oder beim Natural Web I/O Interface.
	2	Die Bildschirmgröße beträgt 24 Zeilen und 80 Spalten.
	3	Die Bildschirmgröße beträgt 32 Zeilen und 80 Spalten.
	4	Die Bildschirmgröße beträgt 43 Zeilen und 80 Spalten.
	5	Die Bildschirmgröße beträgt 27 Zeilen und 132 Spalten.
	( <i>lines, cols</i> )	Diese Syntax ist nur bei NWO-Server-Terminals zulässig. Die Anzahl der Zeilen ( <i>lines</i> ) kann 24 bis 250 sein, und die Anzahl der Spalten ( <i>cols</i> ) kann 80 bis 250 betragen.
<b>Standard-Einstellung</b>	0	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	



#### Anmerkungen:

1. Dieser Natural-Profilparameter ist nur für IBM-Großrechner oder für das Natural Web I/O Interface vorgesehen.
2. Er dient dazu, die Anzahl der Zeilen und Spalten auf dem Terminal-Bildschirm festzulegen.
3. Wenn Ihre TMODEL-Angabe inkompatibel mit der physischen Bildschirmgröße ist, können die Ausgabedaten unrichtig angezeigt werden oder es können Hardware-Fehler auftreten.
- 4.

5. TMODEL kann außerdem unter dem Natural Development Server (NDV) und für den Natural Web I/O Interface (NWO) Server in jeder Betriebssystemumgebung zum Definieren der Terminal-Bildschirmgröße für das Natural Web I/O Interface benutzt werden. Weitere Informationen siehe Natural Development Server-Dokumentation, *Configuring the Natural Development Server* bzw. *Natural Web I/O Interface*-Dokumentation.
6. Die Bildschirmgröße hat direkten Einfluß auf den Speicher, der für die von Natural verwendeten Terminal-E/A-Puffer erforderlich ist.
7. Unter CICS wird dieser Parameter bei terminalgeführten Sessions ignoriert, weil bei diesen die Terminal-Bildschirmgröße durch die CICS-Terminal-Steuertabelle definiert wird.

# 257 TPF (für interne Verwendung)

---

Dieser Parameter ist für die interne Nutzung durch Natural reserviert.

 **Vorsicht:** Ändern Sie seine Einstellung nicht.



# 258

## TQ - Translate Quotation Marks

---

Dieser Parameter ist durch den Subparameter TQMARK des Profilparameters **CMPO** ersetzt worden.



# 259

## TRACE - Komponenten für den Trace angeben

---

▪ TRACE-Parameter-Syntax .....	782
▪ NTTRACE-Makro-Syntax .....	783
▪ Beispiel für TRACE-Parameter .....	783
▪ Beispiel für NTTRACE-Makro .....	783

Mit diesem Profilparameter können Sie die Komponenten definieren, für die Trace-Daten geschrieben werden sollen. TRACE entspricht dem Makro NTTRACE im Natural-Parametermodul.



**Vorsicht:** Benutzen Sie diesen Parameter nicht ohne vorherige Rücksprache mit dem Software AG Support.

<b>Mögliche Werte</b>	Siehe <i>TRACE-Parameter-Syntax</i> .	
<b>Standard-Einstellung</b>	keine	
<b>Dynamische Angabe</b>	ja	Dieser Parameter kann nur dynamisch angegeben werden. Im Natural-Parametermodul wird stattdessen das Makro NTTRACE benutzt.
<b>Angabe innerhalb der Session</b>	nein	



#### Anmerkungen:

1. Dieser Natural-Profilparameter ist in erster Linie für Debugging-Zwecke zur Software AG-internen Verwendung vorgesehen. Er aktiviert nicht die Trace-Aufzeichnung.
2. Die Trace-Aufzeichnung kann wie folgt aktiviert werden: Mit den Profilparametern ITRACE (interner Trace) und ETRACE (externer Trace) oder während der Session mit den entsprechenden Terminalkommandos %TRI und %TRE.
3. Die Listen mit Einstellungen von mehreren TRACE-Parameterangaben werden nicht verkettet; d.h., ein TRACE-Parameter überschreibt die Angaben von zuvor angegebenen TRACE-Parametern und vorhandene NTTRACE-Makro-Definitionen.

## TRACE-Parameter-Syntax

Der Parameter TRACE wird wie folgt angegeben:

```
TRACE=(trace-id,trace-id,...)
```

Dabei ist:

Syntax Element	Value	Explanation
<i>trace-id</i>	1-8 bytes pro Kennung.	Die Trace-Kennungen <i>trace-ID</i> definieren die Namen der Natural-Komponenten, für die ein Trace erfolgen soll. Die Namen der Komponenten müssen in Großbuchstaben angegeben werden.

## NTTRACE-Makro-Syntax

---

Das Makro NTTRACE wird wie folgt angegeben:

```
NTTRACE trace-id,trace-id,...
```



### Anmerkungen:

1. Erklärung der Syntax-Elemente siehe [TRACE-Parameter-Syntax](#).
2. Mehrfache Angaben des NTTRACE-Makros werden zu einer Trace-Liste verkettet.

## Beispiel für TRACE-Parameter

---

```
TRACE=(NATGETM,NATFREM,DYNPARMS)
```

Diese Angabe definiert, dass Trace-Daten für die Natural-Nukleus-Komponenten „Speicherbeschaffung“, „Speicherfreigabe“ und „dynamische Parameterauswertung“ geschrieben werden sollen.

## Beispiel für NTTRACE-Makro

---

```
NTTRACE NATGETM,NATFREM,DYNPARMS
```

Diese Angabe definiert, dass Trace-Daten für die Natural-Nukleus-Komponenten „Speicherbeschaffung“, „Speicherfreigabe“ und „dynamische Parameterauswertung“ geschrieben werden sollen.



# 260 TS - Umsetzung von System-Library-Programmausgaben

Dieser Natural Profil- und Session-Parameter dient dazu, Ausgaben von Natural-System-Libraries (d.h. Libraries, 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 oder fernöstlichen Ländern) erforderlich.



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

<b>Mögliche Werte</b>	ON	Die Ausgabe wird umgesetzt.  <b>Anmerkung:</b> Bei <code>TS=ON</code> werden der Profilparameter <code>LC=OFF</code> und der Session-Parameter <code>AD=T</code> , die beide Eingaben in Großbuchstaben konvertieren, ignoriert, weil sie eine unerwünschte Zeichen-Konvertierung für spezifische Zeichensätze verursachen würden
	OFF	Die Ausgabe wird nicht umgesetzt.
<b>Standard-Einstellung</b>	OFF	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	ja	
<b>Gültige Statements</b>	SET GLOBALS	
<b>Gültiges Kommando</b>	GLOBALS	
<b>Programmierschnittstelle (API)</b>	USR1005N	Siehe <i>SYSEXT Utility</i> in der Dokumentation <i>Debugger und Dienstprogramme</i> .



**Anmerkungen:**

1. 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). Wird der Text in der lokalen Sprache (zum Beispiel Hebräisch) angezeigt, dann wird er nicht in Großbuchstaben umgesetzt. Die Umsetzung von Meldungen und Warnungen ist nicht abhängig von der Library, aus der das Programm ausgeführt wird.
2. Innerhalb einer Natural-Session kann der Profilparameter TS durch den Session-Parameter TS überschrieben werden.
3. Die Umsetzungstabelle kann mit dem `NTTABL`-Makro oder dem betreffenden dynamischen Profilparameter `TABL` geändert werden.

### **Unterstützung von TS=ON bei Natural under IMS/TM-Meldungen**

Alle Natural under 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 under z/OS
- 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 - Größe des Puffers für Adabas Text Retrieval

---

Dieser Natural-Profilparameter gibt die Größe des für Adabas Text Retrieval zu benutzenden Puffers an.

<b>Mögliche Werte</b>	1-2097151	Puffergröße in KB. <b>Anmerkung:</b> Falls der angeforderte Speicher nicht verfügbar ist, kann Adabas Text Retrieval nicht benutzt werden.
	0	Adabas Text Retrieval wird nicht benutzt.
<b>Standard-Einstellung</b>	0	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	



**Anmerkung:** Alternativ können Sie den äquivalenten Natural-Profilparameter [DS](#) oder das Parametermakro [NTDS](#) benutzen.



# 262

## TSOP - Parameter für Natural TSO Interface

---

■ TSOP-Parameter-Syntax .....	790
■ NTTSOP-Makro-Syntax .....	790
■ Schlüsselwort-Subparameter .....	791
■ Beispiel für TSOP-Parameter .....	795
■ Beispiel für NTTSOP-Makro .....	795

Die Parameter für das Natural TSO Interface können als Subparameter des Profilparameters [TSOP](#) oder des Makros [NTTSOP](#) angegeben werden.

<b>Mögliche Werte</b>	Siehe <a href="#">TSOP-Parameter-Syntax</a> .	
<b>Standard-Einstellung</b>	Siehe <a href="#">Schlüsselwort-Subparameter</a> .	
<b>Dynamische Angabe</b>	ja	Der Parameter TSOP kann nur dynamisch angegeben werden. Im Natural-Parametermodul wird das Makro <a href="#">NTTSOP</a> verwendet.
<b>Angabe innerhalb der Session</b>	nein	

## TSOP-Parameter-Syntax

---

Der Profilparameter TSOP wird wie folgt angegeben:

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

Informationen zu Namen und Werten von Schlüsselwort-Subparametern siehe [Schlüsselwort-Subparameter](#).

## NTTSOP-Makro-Syntax

---

Das Makro NTTSOP macro wird wie folgt angegeben:

```
NTTSOP ABEXIT=value, *
      ALTSCRN=value, *
      LBPNAME=value, *
      LEHDLR=value, *
      NDBFSRV=value, *
      PA2=value, *
      SUBPOOL=value, *
      SWAPKEY=value, *
      TIOBSZ=value
```

Siehe [Schlüsselwort-Subparameter](#).

## Schlüsselwort-Subparameter

ABEXIT | ALTSCRN | LBPNAME | LEHDLR | NDBFSRV | PA2 | SUBPOOL | SWAPKEY | TIOBSZ

### ABEXIT - Abbruch-Verarbeitung

ABEXIT=*value* gibt die Art der Abbruch-Verarbeitung innerhalb von Natural an.

Wert	Erklärung
ESTAE	Natural fängt alle Abbrüche ab und gibt die entsprechenden Fehlermeldungen aus. Dies ist der Standardwert.
SPIE	Nur Programmprüfungen (SOCX-Abbrüche) werden abgefangen.
OFF	Natural fängt überhaupt keine Abbrüche bzw. Programmprüfungen ab.



#### Anmerkungen:

1. Die Einstellung ABEXIT=OFF entspricht dem Profilparameter DU=FORCE.
2. wird nicht empfohlen, weil einige Funktionen, die das Abfangen von Abbrüchen benötigen, dann nicht mehr funktionieren. Durch Benutzung des Profilparameters MT wird anstelle des Fehlers NAT0953 ein Abbruch U0322 bewirkt, wenn die CPU-Zeitgrenze erreicht wird.

### ALTSCRN – Session-Bildschirm-Modus

ALTSCRN=*value* gibt an, ob die alternative Bildschirmgröße verwendet werden soll.

Wert	Erklärung
ON	Die alternative Bildschirmgröße soll verwendet werden. Dies ist der Standardwert.
OFF	Die Standard-Bildschirmgröße soll verwendet werden.



#### Anmerkungen:

1. Aufgrund der VTAM LOGMODE-Definition gibt es zwei Bildschirm-Höhen-/Breiten-Einstellungen: Standard und Alternativ. Üblicherweise kommt die Standardgröße mit 24 Zeilen und 80 Spalten zur Anwendung. Die alternative Bildschirmgröße ist vom 3270-Terminal-Modell abhängig (2, 3, 4 oder 5).
2. Die Bildschirmgröße kann mit dem Natural-Profilparameter TMODEL überschrieben werden.

## LBPNAME – Gemeinsame Nutzung Lokaler Buffer Pools

LBPNAME=*value* steuert die gemeinsame Nutzung von lokalen Buffer Pools, wenn mehrere Natural-Sessions innerhalb desselben TSO-Region laufen. Dieser Subparameter definiert den Namen der gemeinsam genutzten lokalen Buffer Pool-Umgebung und dient zur Standortfestlegung des gemeinsam genutzten lokalen Buffer Pool.

Wert	Erklärung
1 - 8 Zeichen oder ' ' (leer)	Name der gemeinsam genutzten lokalen Buffer Pool-Umgebung.
' ' (leer)	Die lokalen Buffer Pools werden nicht gemeinsam genutzt.  Dies ist der Standardwert.



**Anmerkung:** Wenn mehrere Natural-Sessions in einer z/OS-Batch- oder TSO-Region gleichzeitig laufen, dann ordnet jede Session Speicher für einen separaten Buffer Pool zu. Außer für den Natural-z/OS-Batch-Modus-Server werden die lokalen Buffer Pools standardmäßig nicht gemeinsam genutzt, das heißt, wenn die verschiedenen Sessions dieselben Natural-Objekte benutzen, müssen diese separat jeweils einmal für jede Session geladen werden. Wenn *name* angegeben wird, nutzen alle Natural-Sessions gemeinsam denselben lokalen Buffer Pool.

## LEHDLR – Benutzung einer LE-Fehlerbehandlungsroutine für aufrufende LE-Subprogramme

LEHDLR=*value* gibt an, ob Natural eine LE-Fehlerbehandlungsroutine für den Aufruf von LE-Subprogramme benutzt.

Wert	Erklärung
ON	Während des Aufrufs von LE-Subprogrammen wird von Natural eine LE-Fehlerbehandlungsroutine aktiviert. Das bedeutet, dass Natural die Kontrolle erhält, wenn bei der Ausführung eines LE-Subprogramms ein nicht behandelter LE-Fehler auftritt, und dann diesen Fehler (durch Ausgabe der Fehlermeldung NAT0954) behandeln kann.  Dies ist der Standardwert.
OFF	Während des Aufrufs von LE-Subprogrammen wird von Natural keine LE-Fehlerbehandlungsroutine aktiviert. Das bedeutet, dass die LE-Enklave beendet wird und damit die Natural-Session verlorenght, wenn bei der Ausführung eines LE-Subprogramms ein nicht behandelter LE-Fehler auftritt.



### Anmerkungen:

1. Informationen zu LE-Laufzeitoptionen siehe Beschreibung des Source-Moduls NATLEOPT in der *Installation for z/OS-Dokumentation*.

2. Informationen zum Betrieb von Natural mit der IBM Language Environment siehe *Natural Execution - Miscellaneous Topics, LE Subprograms* in der *Operations*-Dokumentation.

### NDBFSRV - Natural for Db2 File Server

NDBFSRV=*value* gibt an, ob der Natural for Db2 File Server benutzt werden soll.

Wert	Erklärung
OFF	Der Natural for Db2 File Server wird nicht benutzt. Dies ist der Standardwert.
ON	Der Natural for Db2 File Server wird bei jeder Terminal-Ein-/Ausgabe aufgerufen.

### PA2 - Verhalten der PA2-Taste

PA2=*value* gibt an, wie die PA2-Taste funktionieren soll.

Wert	Erklärung
ON	Der Wert der PA2-Taste wird an die Natural-Anwendung übergeben.
OFF	Der Wert der PA2-Taste wird verwendet, um den Terminal-Bildschirminhalt erneut anzuzeigen und wird nicht an die Natural-Anwendung übergeben. Dies ist der Standardwert.



**Anmerkung:** Die PA2-Subparameterangabe hat keine Bedeutung, wenn die PA2-Taste mit dem Subparameter **SWAPKEY** als Bildschirm-Umschalttaste definiert wird.

### SUBPOOL – Speicher-Subpool für GETMAIN-Anforderungen

SUBPOOL=*value* gibt den Speicher-Subpool für GETMAIN-Anforderungen an.

Wert	Erklärung
1 - 127	Subpool-Nummer.
0	Dies ist der Standardwert.



#### Anmerkungen:

1. Der Subparameter SUBPOOL wird nur in dem Natural-Parametermodul berücksichtigt, welches mit dem TSO-Treiber verlinkt ist, jedoch nicht in einem alternativen Parametermodul, das durch Angabe von **PARM=** aktiviert wird.
2. Da der Subparameter SUBPOOL nur während der Session-Initialisierung ausgewertet wird, kann er nicht als dynamischer Subparameter angegeben werden.

## SWAPKEY – Unterstützung der TSO/ISPF-Bildschirm-Umschaltung

SWAPKEY=*value* definiert die TSO/ISPF-Bildschirm-Umschalttaste für Natural, die bei den meisten Masken des Software AG-Produkts Natural ISPF (Integrated Structured Programming Facility) der Taste PF9 zugewiesen ist.

Wert	Erklärung
PF1 - PF24 PA1 - PA3	Definiert die PF- oder PA-Taste, die zum Umschalten auf die nächste TSO/ISPF-Session benutzt werden soll. Gültige Tasten sind PF1 bis PF24 und PA1 bis PA3.
OFF	Standardmäßig ist keine Umschalttaste definiert; das heißt, es wird keine Unterstützung der Bildschirm-Umschaltung generiert.



### Anmerkungen:

1. Die angegebene Taste kann nicht von Natural-Anwendungen verwendet werden. Üblicherweise ist das Kommando SWAP in TSO/ISPF auf den meisten TSO/ISPF-Masken der Taste PF9 zugewiesen.
2. Zur Unterstützung der Bildschirm-Umschaltung muss das TSO/ISPF Interface-Modul ISPLINK aus der Ladebibliothek im Link-Schritt für NATTSO enthalten sein. Sie können ISPLINK im Link-Schritt für NATTSO inkludieren. Wenn das Modul nicht verlinkt ist, versucht Natural, es dynamisch zu laden, wenn die mit dem Subparameter SWAPKEY definierte Taste erstmals innerhalb der Session benutzt wird. Wenn ISPLINK nicht in der Ladebibliothek enthalten ist, verhält sich Natural so, als ob SWAPKEY auf OFF gesetzt ist.

## TIOBSZ – Größe des Terminal-Ein-/Ausgabe-Puffers

TIOBSZ=*value* gibt die Größe des Terminal-Ein-/Ausgabe-Puffers an. Der Puffer wird unterhalb der 16-MB-Grenze zugeordnet.

Wert	Erklärung
4 - 32	Größe des Terminal-Ein-/Ausgabe-Puffers in KB.
8	Dies ist der Standardwert.



**Anmerkung:** Die Zuordnung erfolgt unterhalb der 16 MB-Grenze.

## Beispiel für TSOP-Parameter

---

```
TSOP=(LBPNAME=NATTEST1,TIOBSZ=12,SWAPKEY=PF9)
```

## Beispiel für NTTSOP-Makro

---

```
NTTSOP LBPNAME=NATTEST1,TIOBSZ=12,SWAPKEY=PF9
```



# 263

## TTYTYPE - Terminaltyp

---

Mit diesem Natural-Profilparameter können Sie in TP-Umgebungen, in denen diese Information nicht automatisch geliefert wird, den verwendeten Terminaltyp angeben, so dass Natural die passende Konverter-Routine für Attributfolgen aktivieren kann, um diesen Terminaltyp zu betreiben.

<b>Mögliche Werte</b>	1 - 4 Zeichen	Die mit dem TTYTYPE-Parameter angegebene Einstellung muss als gültiger Terminal-Gerätetyp im Makro NTDVCE des NATCONFIG-Moduls definiert sein, siehe <i>Configuration Tables - Module NATCONFIG</i> .
<b>Standard-Einstellung</b>	3270	Standard-Einstellung unter z/OS.
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	ja	Der Parameter TTYTYPE hat dieselbe Funktion wie das Terminalkommando %T=.



**Anmerkung:** Wenn Sie den Parameter TTYTYPE benutzen, ist es nicht mehr nötig, beim Session-Start ein Programm auszuführen, das ein Statement `SET CONTROL 'T=...` enthält, um den Terminaltyp einzustellen.



# 264 UC - 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.

<b>Mögliche Werte</b>	beliebiges Zeichen OFF	Siehe auch die folgende Anmerkung.
<b>Standard-Einstellung</b>	-	Bindestrich (-)
<b>Angabe innerhalb der Session</b>	ja	
<b>Gültige Statements</b>	<code>DISPLAY</code> <code>FORMAT</code> <code>WRITE TITLE</code> <code>WRITE TRAILER</code>	
<b>Gültiges Kommando</b>	keines	



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

<code>UC=</code>	Statt einer Unterstreichung wird eine Leerzeile ausgegeben.
<code>UC=OFF</code>	Die Feldwerte werden unmittelbar unter der Überschrift, ohne Leerzeile dazwischen, ausgegeben. Sie können <code>UC=OFF</code> nur auf Statement-Ebene eines <code>DISPLAY</code> -Statements angeben; in diesem Fall können Sie für einzelne Felder in dem Statement keine anderen <code>UC</code> -Angaben machen.

**Beispiele:**

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



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

# 265 UCONMAX - Maximale Anzahl gleichzeitiger Sessions für einen Benutzer

---

Dieser Natural-Profilparameter dient dazu, die maximal mögliche Anzahl gleichzeitiger Sessions für einen Benutzer anzugeben.

Die Natural-Systemdatei `FREG` wird verwendet, um die mit einem Wert größer als Null gestarteten Sessions für den Natural-Profilparameter `UCONMAX` zu registrieren.

<b>Mögliche Werte</b>	1 - 32767	Maximale Anzahl gleichzeitiger Sessions für einen Benutzer.
	0	UCONMAX=0 gibt an, dass keine Einschränkung wirksam sein soll.
<b>Standard-Einstellung</b>	0	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	



## Anmerkungen:

1. Wird die angegebene Anzahl gleichzeitiger Session überschritten, wird die Natural-Session abgelehnt und der Benutzer erhält eine entsprechende Natural-Fehlermeldung.
2. Dieser Natural-Profilparameter ist wirksam für Server, welche den Natural Development Server und das *Natural Web I/O Interface* unterstützen.



# 266

## UDB - Benutzer-Datenbankkennung

Dieser Natural-Profilparameter gibt die Datenbankkennung (DBID) an, die für den Datenbankzugriff verwendet werden soll.

<b>Mögliche Werte</b>	0 - 65535, außer 255	Gültige Datenbankkennung. <b>Anmerkung:</b> Die Datenbankkennung 255 ist für logische Systemdateien für Software AG-Produkte reserviert, siehe Profilparameter <a href="#">LFILE</a> .
<b>Standard-Einstellung</b>	Die Datenbankkennung, die für FUSER gültig ist.	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	
<b>Anwendungsprogrammierschnittstelle</b>	USR1005N USR1040N *	Siehe <i>SYSEXT Utility</i> in der Dokumentation <i>Debugger und Dienstprogramme</i> .  * empfohlen



### Anmerkungen:

1. Die Datenbankkennung 0 und die mit dem Parameter UDB ausgewählten Datenbanken müssen vom selben Typ sein (z.B.: ADA/ADA, SQL/SQL oder XML/XML).
2. Wenn in der verwendeten DDM keine Datenbankkennung angegeben ist, bestimmt die mit dem UDB-Parameter angegebene Datenbankkennung, auf welche Datenbank zugegriffen werden soll. Somit ist es möglich, verschiedene Benutzerumgebungen zu haben, ohne mehrere FUSER-Dateien zu benötigen.
3. Wenn im DDM keine Datenbankkennung angegeben ist und wenn der Profilparameter UDB nicht angegeben ist, dann wird die Datenbankkennung benutzt, die für die Systemdatei FUSER gilt.



# 267

## ULANG - Benutzersprache

Dieser Natural-Profilparameter dient zur Angabe der Sprache, die für Datumseditiermasken, Systemmeldungen, Benutzermeldungen, Hilfetexte, Helprountinen und mehrsprachigen Masken verwendet werden soll. Die Einstellung wird verwendet, um die Natural-Systemvariable \*LANGUAGE zu setzen.

<b>Mögliche Werte</b>	1 - 60	Natural-Sprachcode. <b>Anmerkung:</b>  1. Zum Beispiel: 1 steht für Englisch, 2 für Deutsch, 3 für Französisch.  2. Eine ausführliche Liste der Sprachcodes finden Sie in der Dokumentation der Systemvariablen *LANGUAGE.
<b>Standard-Einstellung</b>	1	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	
<b>Anwendungsprogrammierschnittstelle</b>	USR1005N	Siehe <i>SYSEXT Utility</i> in der Dokumentation <i>Debugger und Dienstprogramme</i> .



### Anmerkungen:

1. Siehe auch den Hinweis zur sprachcodebezogenen Anpassung des Profilparameters [CP](#), wenn dieser auf ON gesetzt ist.
2. Innerhalb der Session kann der Sprachcode mit dem Terminalkommando %L= angegeben werden.
3. Siehe auch:
  - *Configuration Tables - Module NATCONFIG* in der *Operations*-Dokumentation bezüglich zusätzlicher Informationen über Sprachkennzeichen und mögliche Werte.

- *Bildschirmgestaltung, Kenntnisabhängige Benutzeroberflächen (Expertenmodus) im Leitfaden zur Programmierung.*

# 268 UNIO (für interne Verwendung)

---

Dieser Parameter ist für die interne Nutzung durch Natural reserviert.



**Vorsicht:** Ändern Sie seine Einstellung nicht.



# 269 USER - Verwendung von Profilparameterketten und

## Modulen einschränken

---

▪ USER-Parameter-Syntax .....	810
▪ NTUSER-Makro-Syntax .....	811
▪ Beispiel für USER-Parameter .....	812
▪ Beispiel für NTUSER-Makro .....	812

Dieser Profilparameter kann verwendet werden, um die Benutzung von dynamischen Parameterketten, die in einem SYSPARM-Profil, einem NTSYS-Makro oder Parameter-Dataset (CMPRMIN) angegeben sind, einzuschränken oder um ein alternatives Natural-Parametermodul einzuschränken.

<b>Mögliche Werte</b>	Siehe <i>USER-Parameter-Syntax</i> .	
<b>Standard-Einstellung</b>	keine	
<b>Dynamische Angabe</b>	ja	Dieser Parameter kann nur dynamisch angegeben werden.  Um die Verwendung eines alternativen Natural-Parametermoduls einzuschränken, wird das entsprechende Makro NTUSER benutzt.
<b>Angabe innerhalb der Session</b>	nein	



**Anmerkungen:**

1. Der Parameter USER gilt nur für die Kette mit dynamischen Parametern, die *nach* ihm angegeben ist.
2. Wenn die dynamischen Profilparameter ausgewertet werden und wenn dabei ein USER-Parameter vorgefunden wird, prüft Natural, ob die aktuelle Benutzerkennung (d.h. die aktuelle Einstellung der Systemvariablen \*INIT-USER) in der Liste der Benutzerkennungen enthalten ist, die mit dem USER-Parameter angegeben sind. Falls sie nicht darin enthalten ist, erhält der Benutzer eine entsprechende Fehlermeldung, und die Verarbeitung der dynamischen Profilparameter wird unmittelbar darauf beendet.

## USER-Parameter-Syntax

Der Parameter USER wird wie folgt angegeben:

```
USER=(user-id,user-id,...),profile-parameter-string
```

Dabei ist:

Syntax-Element	Erklärung
<i>user-id</i>	Benutzerkennungen der Benutzer an, denen es gestattet wird, die anschließend als Zeichenkette angegebenen Profilparameter zu benutzen.
<i>profile-parameter-string</i>	Kette mit Profilparametern.

➤ **Um die Benutzung eines SYSPARM-Profiles einzuschränken:**

- Geben Sie den **USER**-Parameter als ersten Parameter in dem Profil an. Die nachfolgende Profilparameterkette, d.h. das gesamte Profil, kann dann nur von dem Benutzer benutzt werden, der im **USER**-Parameter angegeben ist.

➤ **Um die Benutzung einer Parameterkette einzuschränken, die mit einem NTSYS-Makro oder in einem CMPRMIN-Dataset definiert ist:**

- Geben Sie den **USER**-Parameter als ersten Parameter in der Parameterkette an.

## NTUSER-Makro-Syntax

Das Makro NTUSER wird im Natural-Parametermodul wie folgt angegeben:

```
NTUSER user-id,user-id,user-id,...,profile-parameter-string
NTUSER user-id,user-id,...,profile-parameter-string
```



**Anmerkungen:**

1. Erklärung der Syntax-Elemente siehe [USER-Parameter-Syntax](#).
2. Das Makro NTUSER gilt für das Parametermodul, in dem es angegeben wird. Das zum umgebungsabhängigen Natural-Nukleus gelinkte Standard-Natural-Parametermodul kann nicht eingeschränkt werden.

➤ **Um die Benutzung eines alternativen Parametermoduls einzuschränken:**

- Geben Sie das Makro NTUSER in dem alternativen Parametermodul an.



**Anmerkung:** Wenn ein alternatives Parametermodul benutzt werden soll, lädt Natural das mit dem PARM-Parameter angegebene alternative Parametermodul und prüft, ob die aktuelle Benutzererkennung (d.h. die aktuelle Einstellung der Systemvariablen \*INIT-USER) in der Liste der Benutzerkennungen enthalten ist, die mit dem NTUSER-Makro in dem alternativen Parametermodul angegeben wird. Falls sie nicht darin enthalten ist, erhält der Benutzer eine entsprechende Fehlermeldung, und das alternative Parametermodul wird übergangen.

## Beispiel für USER-Parameter

---

```
USER=(ADMIN1,ADMIN2),FNAT=(12,177,SECPASSW,74832055)
```

## Beispiel für NTUSER-Makro

---

Das folgende Beispiel zeigt, wie ein Natural-Parameternakro geschützt wird:

```
NTPRM ...  
...  
NTUSER ADMIN1,ADMIN2
```

# 270 USERBUF (für interne Verwendung)

---

Dieser Parameter ist für die interne Nutzung durch Natural reserviert.

 **Vorsicht:** Ändern Sie seine Einstellung nicht.



# 271

## UTAB1 - Umsetzung von Klein- in Großschreibung

---

■ UTAB1-Parameter-Syntax .....	816
■ NTUTAB1-Makro-Syntax .....	817
■ Beispiel für UTAB1-Parameter .....	817
■ Beispiel für NTUTAB1-Makro .....	817

Mit diesem Natural-Profilparameter können Sie die Definitionen in der Umsetzungstabelle NTUTAB1 überschreiben, die im Konfigurationsmodul NATCONFIG enthalten ist.

<b>Mögliche Werte</b>	Siehe <a href="#">UTAB1 Parameter-Syntax</a> weiter unten.	
<b>Standard-Einstellung</b>	Gemäß den Angaben im Makro NTUTAB1 in NATCONFIG.	
<b>Dynamische Angabe</b>	ja	Dieser Parameter kann nur dynamisch angegeben werden. Im Natural-Parametermodul wird stattdessen das Makro NTUTAB1 benutzt.
<b>Angabe innerhalb der Session</b>	nein	



**Anmerkungen:**

1. UTAB1 entspricht dem Makro [NTUTAB1](#) im Natural-Parametermodul.
2. Die Tabelle NTUTAB1 wird benutzt, um Zeichen in Kleinschreibung in Großschreibung umzusetzen.
3. Wenn der Profilparameter [CP](#) auf einen anderen Wert als OFF gesetzt ist, werden die mit UTAB1 angegebenen Werte ignoriert. Siehe auch *Umsetzungstabellen (Translation Tables)* in der *Unicode- und Codepage-Unterstützung-Dokumentation*.

## UTAB1-Parameter-Syntax

---

Der Parameter UTAB1 wird wie folgt angegeben:

```
UTAB1=(a1,a2,b1,b2,c1,c2,...)
```

Sie geben Zeichenpaare an, wobei das erste Zeichen eines Paares das umzusetzende Zeichen in Kleinschreibung und das zweite Zeichen eines Paares das Zeichen in Großschreibung ist, in welches das erste Zeichen umgesetzt werden soll.

Sie können jedes Zeichen entweder als das 1-Byte-Zeichen selbst (in Hochkommas gesetzt) oder als 2-Byte-Hexadezimaldarstellung dieses Zeichens angeben.

Beim Parameter UTAB1 müssen Sie die gesamte Zeichenkette mit Zeichenpaaren in Klammern setzen. Siehe [Beispiel für UTAB1-Parameter](#).

Oder:

```
UTAB1=OFF
```

Bei `UTAB1=OFF` werden alle (statischen und dynamischen) Definitionen auf die Werte zurückgesetzt, die im Makro `NTUTAB1` im Modul `NATCONFIG` angegeben sind.

## NTUTAB1-Makro-Syntax

Das Makro `NTUTAB1` wird wie folgt angegeben:

```
NTUTAB1 a1,a2,b1,b2,c1,c2,...
```



### Anmerkungen:

1. Erklärung der Syntax-Elemente siehe [UTAB1-Parameter-Syntax](#). Beispiel siehe [Beispiel für NTUTAB1-Makro](#).
2. Der Wert `OFF` kann nicht mit dem Makro `NTUTAB1`, sondern nur dynamisch mit dem Profilparameter `UTAB1` angegeben werden.

## Beispiel für UTAB1-Parameter

```
UTAB1=(5E,'Ä','ö',78,FF,00,'ü','Ü')
```

In diesem Beispiel wird das durch `H'5E'` dargestellte Zeichen in `'Ä'` umgesetzt, `'ö'` in das durch `H'78'` dargestellte Zeichen, das durch `H'FF'` dargestellte Zeichen in das durch `H'00'` dargestellte Zeichen und `'ü'` in `'Ü'`.

## Beispiel für NTUTAB1-Makro

```
NTUTAB1 5E,'Ä','ö',78,FF,00,'ü','Ü'
```

In diesem Beispiel wird das durch `H'5E'` dargestellte Zeichen in `'Ä'` umgesetzt, `'ö'` in das durch `H'78'` dargestellte Zeichen, das durch `H'FF'` dargestellte Zeichen in das durch `H'00'` dargestellte Zeichen und `'ü'` in `'Ü'`.



# 272

## UTAB2 - Umsetzung von Groß- in Kleinschreibung

---

■ UTAB2-Parameter-Syntax .....	820
■ NTUTAB2-Makro-Syntax .....	821
■ Beispiel für UTAB2-Parameter .....	821
■ Beispiel für NTUTAB2-Makro .....	821

Mit diesem Natural-Profilparameter können Sie die Definitionen in der Umsetzungstabelle NTUTAB2 überschreiben, die im Konfigurationsmodul NATCONFIG enthalten ist.

<b>Mögliche Werte</b>	Siehe <a href="#">UTAB2 Parameter-Syntax</a> weiter unten.	
<b>Standard-Einstellung</b>	Gemäß den Angaben im Makro NTUTAB2 in NATCONFIG.	
<b>Dynamische Angabe</b>	ja	Dieser Parameter kann nur dynamisch angegeben werden. Im Natural-Parametermodul wird stattdessen das Makro NTUTAB2 benutzt.
<b>Angabe innerhalb der Session</b>	nein	



**Anmerkungen:**

1. UTAB2 entspricht dem Makro [NTUTAB2](#) im Natural-Parametermodul.
2. Die Tabelle NTUTAB2 wird benutzt, um Zeichen in Großschreibung in Kleinschreibung umzusetzen.
3. Wenn der Profilparameter [CP](#) auf einen anderen Wert als OFF gesetzt ist, werden die mit UTAB2 angegebenen Werte ignoriert. Siehe auch *Umsetzungstabellen (Translation Tables)* in der *Unicode- und Codepage-Unterstützung-Dokumentation*.

## UTAB2-Parameter-Syntax

---

Der Parameter UTAB2 wird wie folgt angegeben:

```
UTAB2=(a1,a2,b1,b2,c1,c2,...)
```

Sie geben Zeichenpaare an, wobei das erste Zeichen eines Paares das umzusetzende Zeichen in Großschreibung und das zweite Zeichen eines Paares das Zeichen in Kleinschreibung ist, in welches das erste Zeichen umgesetzt werden soll.

Sie können jedes Zeichen entweder als das 1-Byte-Zeichen selbst (in Hochkommas gesetzt) oder als 2-Byte-Hexadezimaldarstellung dieses Zeichens angeben.

Beim Parameter UTAB2 müssen Sie die gesamte Zeichenkette mit Zeichenpaaren in Klammern setzen. Siehe [Beispiel für UTAB2-Parameter](#).

Oder:

UTAB2=OFF

Bei `UTAB2=OFF` werden alle (statischen und dynamischen) Definitionen auf die Werte zurückgesetzt, die im Makro `NTUTAB2` im Modul `NATCONFIG` angegeben sind.

## NTUTAB2-Makro-Syntax

---

Das Makro `NTUTAB2` wird wie folgt angegeben:

```
NTUTAB2 a1,a2,b1,b2,c1,c2,...
```



### Anmerkungen:

1. Erklärung der Syntax-Elemente siehe [UTAB2-Parameter-Syntax](#). Beispiel siehe [Beispiel für NTUTAB2-Makro](#).
2. Der Wert `OFF` kann nicht mit dem Makro `NTUTAB2`, sondern nur dynamisch mit dem Profilparameter `UTAB2` angegeben werden.

## Beispiel für UTAB2-Parameter

---

```
UTAB1=(5E,'Ä','ö',78,FF,00,'ü','Ü')
```

In diesem Beispiel wird das durch `H'5E'` dargestellte Zeichen in `'Ä'` umgesetzt, `'ö'` in das durch `H'78'` dargestellte Zeichen, das durch `H'FF'` dargestellte Zeichen in das durch `H'00'` dargestellte Zeichen und `'ü'` in `'Ü'`.

## Beispiel für NTUTAB2-Makro

---

```
NTUTAB2 5E,'Ä','ö',78,FF,00,'ü','Ü'
```

In diesem Beispiel wird das durch `H'5E'` dargestellte Zeichen in `'Ä'` umgesetzt, `'ö'` in das durch `H'78'` dargestellte Zeichen, das durch `H'FF'` dargestellte Zeichen in das durch `H'00'` dargestellte Zeichen und `'ü'` in `'Ü'`.



# 273

## VSAM - Parameter für Natural for VSAM

---

■ VSAM-Parameter-Syntax .....	824
■ NTVSAM-Makro-Syntax .....	824
■ NTVEXIT-Makro-Syntax .....	825
■ NTVLSR-Makro-Syntax .....	825
■ NTVTVSD-Makro-Syntax .....	825
■ Schlüsselwort-Subparameter .....	826
■ Beispiel für VSAM-Parameter .....	841
■ Beispiel für NTVSAM-Makro .....	841

Die Parameter für Natural for VSAM können als Subparameter des dynamischen Profilparameters [VSAM](#) oder des Makros [NTVSAM](#) und der Makros [NTVEXIT](#), [NTVLSR](#) und [NTVTVSD](#) im Natural-Parametermodul angegeben werden.

<b>Mögliche Werte</b>	Siehe <a href="#">VSAM-Parameter-Syntax</a> .	
<b>Standard-Einstellung</b>	Siehe Standardwerte der <a href="#">Schlüsselwort-Subparameter</a> .	
<b>Dynamische Angabe</b>	ja	Der Parameter VSAM kann nur dynamisch angegeben werden. Im Natural-Parametermodul werden die Makros <a href="#">NTVSAM</a> , <a href="#">NTVEXIT</a> , <a href="#">NTVLSR</a> und <a href="#">NTVTVSD</a> verwendet.
<b>Angabe innerhalb der Session</b>	nein	

## VSAM-Parameter-Syntax

Der Profilparameter VSAM wird wie folgt angegeben:

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

Siehe [Schlüsselwort-Subparameter](#).

## NTVSAM-Makro-Syntax

Das Makro NTVSAM wird wie folgt angegeben:

```
NTVSAM BTSUPP=value, *
      CLSUPP=value, *
      DDMCHK=value, *
      DDSWITE=value, *
      DFBE=value, *
      DFBN=value, *
      ENADIS=value, *
      ENAUNE=value, *
      ETSUPP=value, *
      FORMAT=(value1,value2), *
      KEYLGH=value, *
      MF=value, *
      OPSUPP=value, *
      PATH=value, *
      PSIGNF=value, *
      RETRY=(value1,value2), *
      RLS=value, *
      ROLLSIZ=value, *
      SFILE=value, *
```

TAFE= <i>value</i> ,	*
TAFN= <i>value</i> ,	*
TIMEOUT= <i>value</i> ,	*
TSAE= <i>value</i> ,	*
TVS= <i>value</i> ,	*
UPDL= <i>value</i>	

Siehe [Schlüsselwort-Subparameter](#).



**Anmerkung:** Die Schlüsselwort-Subparameter EXIT, LSR und TVSD stehen im Makro NTVSAM nicht zur Verfügung. Darum werden im Natural-Parametermodul stattdessen die Makros [NTVSAM](#), [NTVEXIT](#), [NTVLSR](#) und [NTVTVSD](#) verwendet.

## NTVEXIT-Makro-Syntax

Das Makro NTVEXIT wird wie folgt angegeben:

```
NTVEXIT file-name,exit-name,workarea-size
```

Einzelheiten siehe [EXIT - Datei-User Exits](#).

## NTVLSR-Makro-Syntax

Das Makro NTVLSR wird wie folgt angegeben:

```
NTVLSR file-name,subpool
```

Einzelheiten siehe [LSR - Subpools für gemeinsam genutzte lokale Ressourcen](#).

## NTVTVSD-Makro-Syntax

Das Makro NTVTVSD wird wie folgt angegeben:

```
NTVTVSD file-name,option
```

Einzelheiten siehe [TVSD - Aktivierung der DFSMS Transactional VSAM Services](#).

## Schlüsselwort-Subparameter

BTSUPP | CLSUPP | DDMCHK | DDSWITE | DFBE | DFBN | ENADIS | ENAUNE | ETSUPP | EXIT | FORMAT | KEYLGH | LSR | MF | OPSUPP | PATH | PSIGNF | RETRY | RLS | ROLLSIZ | SFILE | TAFE | TAFN | TIMEOUT | TSAE | TVS | TVSD | UPDL

### BTSUPP - Unterstützung von BACKOUT TRANSACTION-Statements

BTSUPP=*value* gibt an, ob BACKOUT TRANSACTION-Statements ausgeführt werden oder nicht.

Wert	Erklärung
ON	Jedes BACKOUT TRANSACTION-Statement wird ausgeführt und in ein entsprechendes ROLLBACK-Kommando umgesetzt.  Dies ist der Standardwert.
OFF	BACKOUT TRANSACTION-Statements werden nicht beachtet.



**Anmerkung:** Dieser Subparameter gilt nur in TP- und DFSMStvs-Umgebungen, in denen VSAM-Logging unterstützt wird.

### CLSUPP - Unterstützung eines CLOSE-Aufrufs beim Beenden der Session

CLSUPP=*value* gibt an, ob ein CLOSE-Aufruf beim Beenden der Session ausgeführt wird oder nicht.

Wert	Erklärung
ON	Jeder CLOSE-Aufruf wird ausgeführt und in ein entsprechendes SYNCPOINT-Kommando umgesetzt.  Dies ist der Standardwert.
OFF	Jeder CLOSE-Aufruf wird nicht beachtet.



**Anmerkung:** Wenn ein CLOSE-Aufruf ausgeführt wird, erzwingt Natural for VSAM ein END TRANSACTION nur in TP- und DFSMStvs-Umgebungen, in denen VSAM-Logging unterstützt wird.

## DDMCHK - Unterstützung der DDM-Integrität

Bei `DDMCHK=value` wird geprüft, ob sich das Datei-Layout und folglich auch die DDM verändert hat.

Wert	Erklärung
ON	Die DDM-Prüfung ist eingeschaltet.
OFF	Die DDM-Prüfung ist ausgeschaltet. Dies ist der Standardwert.



**Anmerkung:** Die Prüfung erfolgt nach jeder Programmbeendigung auf der NEXT-Ebene über den Natural Buffer Pool. Der Subparameter `DDMCHK` ist nur in Entwicklungsumgebungen relevant, in denen DDMs geändert werden. Um die Performance zu verbessern, sollten Sie diese Prüfung in Produktionsumgebungen ausschalten.

## DDSWITE - Maximale Anzahl der Einträge im DD/DLBL-Namensumschaltpuffer

`DDSWITE=value` gibt die maximale Anzahl der Einträge im DD/DLBL-Namensumschaltpuffer an.

Wert	Erklärung
0 bis <code>TAFE=value</code> .	Maximale Anzahl der Einträge; d.h., 0 oder ein Wert in einem Bereich, der durch den im Parameter <code>TAFE</code> angegebenen Maximalwert vorgegeben ist.
0	Dies ist der Standardwert.



**Anmerkung:** Weitere Informationen zum Umschalten von DD-Namen siehe Anwendungsprogrammierschnittstelle `USR1047N` in der Dokumentation *Debugger und Dienstprogramme*.

## DFBE - Anzahl der dekodierten Format-Puffer-Einträge

`DFBE=value` gibt die Anfangszahl der Einträge der dekodierten Format-Puffer an.

Wert	Erklärung
1 - 1000	Durchschnittliche Anzahl Felder.
10	Dies ist der Standardwert.



**Anmerkungen:**

1. Für jedes aktive Natural-Ein-/Ausgabe-Statement (`FIND`, `READ`, `UPDATE`, `STORE`) wird in dieser Tabelle ein Eintrag zugeordnet.
2. Wenn Sie den Wert von `DFBE` oder `DFBN` erhöhen, müssen Sie berücksichtigen, dass die zugeordnete Speicherbereichsgröße nicht durch Addieren, sondern durch Multiplizieren dieser Werte berechnet wird.

## DFBN - Anzahl der Felder in dekodierten Format-Puffer-Einträgen

DFBN=*value* gibt die durchschnittliche Anzahl der in einem Eintrag der dekodierten Form-Puffer-Tabelle enthaltenen Felder an.

Wert	Erklärung
1 - 1000	Durchschnittliche Anzahl Felder.
50	Dies ist der Standardwert.



### Anmerkungen:

1. Für jedes Natural-Ein-/Ausgabe-Statement (FIND, READ, UPDATE, STORE) wird ein Eintrag erstellt.
2. Wenn Sie den Wert von DFBN oder DFBE erhöhen, müssen Sie berücksichtigen, dass die zugeordnete Speicherbereichsgröße nicht durch Addieren, sondern durch Multiplizieren dieser Werte berechnet wird.

## ENADIS - Ausgeschaltete Felder einschalten

ENADIS=*value* dient dazu, ausgeschaltete Dateien einzuschalten.

Wert	Erklärung
ON	Für alle ausgeschalteten Dateien, auf die während der Session zugegriffen wird, wird ein EXEC CICS SET ENABLED-Kommando ausgeführt.
OFF	Alle ausgeschalteten Dateien bleiben ausgeschaltet. Dies ist der Standardwert.



### Anmerkungen:

1. Dieser Subparameter gilt nur in CICS-Umgebungen und wird nur beim ersten Dateizugriff berücksichtigt, der in der aktuellen Natural Session ausgeführt wird.
2. Falls der Subparameter auf OFF gesetzt ist und die Dateien nicht eingeschaltet wurden, muss auf den ersten Dateizugriff die Fehlermeldung NAT3516 folgen.

## ENAUNE - Nicht eingeschaltete Dateien einschalten

ENAUNE=*value* dient dazu, nicht eingeschaltete Dateien einzuschalten.

Wert	Erklärung
ON	Für alle nicht eingeschalteten Dateien, auf die während der Session zugegriffen wird, wird ein EXEC CICS SET ENABLED-Kommando ausgeführt.  For all „unenabled“ files accessed during the session, an EXEC CICS SET ENABLED command is executed.
OFF	Alle nicht eingeschalteten Dateien bleiben nicht eingeschaltet.  Dies ist der Standardwert.



### Anmerkungen:

1. Dieser Subparameter gilt nur in CICS-Umgebungen und wird nur beim ersten Dateizugriff berücksichtigt, der in der aktuellen Natural Session ausgeführt wird.
2. Falls der Subparameter auf OFF gesetzt ist und die Datei nicht eingeschaltet wurde, muss auf den ersten Dateizugriff die Fehlermeldung NAT3539 folgen.

## ETSUPP - Unterstützung von END TRANSACTION-Statements

ETSUPP=*value* gibt an, ob END TRANSACTION-Statements unterstützt werden oder nicht.

Wert	Erklärung
ON	Jedes END TRANSACTION-Statement wird ausgeführt und in ein entsprechendes SYNCPOINT-Kommando umgesetzt.  Dies ist der Standardwert.
OFF	END TRANSACTION-Statements werden nicht beachtet.



**Anmerkung:** Dieser Subparameter gilt nur in TP- und DFSMStvs-Umgebungen, in denen VSAM-Logging unterstützt wird.

## EXIT - Datei-User Exits

Natural for VSAM bietet die Möglichkeit, einen oder mehrere User Exits zu definieren. Zu jeder VSA-Datei, auf die zugegriffen werden soll, kann ein User Exit definiert werden. Die Definition eines User Exit erfolgt mit dem Subparameter `EXIT` des Profilparameters `VSAM` bzw. mit dem Makro `NTVEXIT` im Natural-Parametermodul.

Der Subparameter `EXIT` wird wie folgt angegeben:

```
EXIT=(dd-name,exit-name,workarea-size)
```

Das Makro `NTVEXIT` wird wie folgt angegeben:

```
NTVEXIT dd-name,exit-name,workarea-size
```

Dabei ist:

Wert	Erklärung
<i>dd-name</i>	DD/DLBL/FCT-Name der VSAM-Datei, auf die zugegriffen werden soll. <b>Anmerkung:</b> Es gibt keinen Standardwert.
<i>exit-name</i>	Name des User Exit. <b>Anmerkung:</b> Es gibt keinen Standardwert.
<i>workarea-size</i>	Optional kann die Größe des User Exit-Arbeitsbereichs (in Bytes) angegeben werden. <b>Anmerkung:</b> <ol style="list-style-type: none"> <li>1. Eine Mindestgröße von 72 Bytes ist erforderlich; diese entspricht der Größe des gespeicherten IBM-Standard-Register-Bereichs, d.h., 18 Vollwörter.</li> <li>2. Der Maximalwert beträgt 1024 Bytes.</li> <li>3. Der Standardwert ist 72.</li> </ol>



**Anmerkung:** Alle User Exits müssen entweder mit dem Natural-Parametermodul verlinkt oder mit Hilfe von CSTATIC- oder RCA-Techniken definiert werden. Für jeden Datei-User Exit ist eine separate Definition mit dem Subparameter `EXIT` bzw. dem Makro `NTVEXIT` erforderlich.

## User Exit Linkage-Konventionen

Bei der Übergabe der Kontrolle an den und von dem User Exit werden IBM-Standard Linkage Register-Notationen verwendet.

Register	Verwendung
R1	Adresszeiger auf die Parameteradressenliste.  Die Parameteradressenliste liefert die Adressen des Datensatzes, des LRECL, der aktuellen Funktion und des Arbeitsbereichs.
R3	Adresszeiger auf die VSAM Control Area (VCA).
R12	Adresszeiger auf den Natural Basic Control Block (BB).
R13	Adresse der 18-Wort-Save Area.
R14	Rückgabe-Adresse.
R15	Eintrag-Adresse/Rückgabeschlüssel.  Ein Rückgabeschlüssel von 0 steht für normale Rückgabe der Kontrolle. In allen übrigen Fällen wird eine Natural-Fehlermeldung zurückgegeben.

Die aktuelle Funktion (siehe oben, Register R1) zeigt an, wie die Kontrolle an den User Exit übergeben worden ist. Die Übergabe der Kontrolle kann entweder vor oder nach einem Natural-Aufruf an VSAM erfolgen (siehe auch Feld DCRREQCD im mitgelieferten Makro NVMDCR).

- Bei den Statements STORE und UPDATE wird die Kontrolle vor dem Aufruf übergeben.
- Bei den Statements FIND, GET und READ wird die Kontrolle nach dem Aufruf übergeben.

### Beispiel-User Exit

Ein Beispiel-User Exit NVSEX01 ist auf dem Installationsband vorhanden.

### FORMAT - Unterstützung der Datensatzformatierung bei STORE- und UPDATE-Statements

Dieser Abschnitt beschreibt die neue (erweiterte) Syntax des Subparameters FORMAT und zusätzlich die **alte Subparameter-Syntax**, die aus Kompatibilitätsgründen noch unterstützt wird.

#### Neue (erweiterte) Subparameter-Syntax:

FORMAT=(*value1*, *value2*) ermöglicht die Unterstützung der Formatierung von VSAM-Datensätzen, die in einem STORE- oder UPDATE-Statement referenziert werden. Datensatzfelder, die nicht referenziert werden und deshalb binäre Nullen enthalten, werden in ein Format umgesetzt, das dem in der relevanten DDM definierten Feldtyp und der dort definierten Datensatzlänge entspricht.

Wert	Erklärung
<i>value1</i> ON	VSAM-Datensätze werden gemäß den entsprechenden DDM-Definitionen formatiert.  Dies ist der Standardwert.
OFF	VSAM-Datensätze werden <i>nicht</i> formatiert, und Felder, die nicht referenziert werden, enthalten binäre Nullen.

Wert		Erklärung
value2	KEYS	Alle VSAM-Schlüssel werden gemäß dem DDM-Feldtyp formatiert. Dies ist der Standardwert.
	NOKEYS	Alle VSAM-Schlüssel werden <i>nicht</i> formatiert.

 **Anmerkung:** Natural for VSAM-Systemdatei-Datensätze werden immer formatiert; dies kann nicht geändert werden.

Sinnvolle Kombinationsmöglichkeiten sind:

FORMAT=(ON , NOKEYS)	Alle Felder, mit Ausnahme der Schlüssel, werden formatiert.
FORMAT=(OFF , KEYS)	Nur die Schlüssel werden formatiert.

### Alte Subparameter-Syntax:

Die alte Subparameter-Syntax wird aus Kompatibilitätsgründen noch unterstützt.

FORMAT=value ermöglicht die Unterstützung der Formatierung von VSAM-Datensätzen, die in einem STORE- oder UPDATE-Statement referenziert werden.

Wert	Wert
ON	VSAM-Datensätze werden gemäß den entsprechenden DDM-Definitionen formatiert. Dies ist der Standardwert.
OFF	VSAM-Datensätze werden nicht formatiert, und Felder, die nicht referenziert werden, enthalten binäre Nullen.

 **Anmerkungen:**

1. Natural for VSAM-Systemdatei-Datensätze werden immer formatiert; dies kann nicht geändert werden.
2. Datensatzfelder, die nicht referenziert werden und deshalb binäre Nullen enthalten, werden in ein Format umgesetzt, das dem in der relevanten DDM definierten Feldtyp und der dort definierten Datensatzlänge entspricht.

## KEYLGH - Länge der in E/A-Statements verwendeten VSAM-Schlüssel

KEYLGH=*value* gibt die Länge von in Ein-/Ausgabe-Statements verwendeten VSAM-Schlüsseln an.

Wert	Erklärung
1 - 255	Länge (in Bytes) von in Ein-/Ausgabe-Statements verwendeten VSAM-Schlüsseln.
32	Dies ist der Standardwert.



### Anmerkungen:

1. Für eine VSAM-Datei beträgt die maximale Schlüssellänge 255 Bytes.
2. Der Wert dieses Parameters wird benutzt, um die Größe der TSA-Tabelle (Table of Sequential Access) zu berechnen.
3. Wenn Sie VSAM-Systemdateien benutzen, geben Sie mindestens folgende Werte an: 87 Bytes für die Systemdateien FNAT, FUSER, FDIC und FSPool bzw. 126 Bytes für die Systemdatei FSEC und für Natural ISPF-Systemdateien.

## LSR - Subpools für gemeinsam genutzte lokale Ressourcen

Dieser Subparameter ist nur erforderlich, wenn VSAM-Dateien als gemeinsam genutzte lokale Ressourcen (Local Shared Resources) benutzt werden.



**Anmerkung:** Diese Methode bewirkt eine erheblichen Performance-Verbesserung bei TSO und bei Batch-Verarbeitungsläufen und gleichzeitig zu einer Verringerung der VSAM-Ein-/Ausgabe-Rate. Die Definition der Verwendung eines gemeinsam genutzten lokalen Ressourcen-Subpool pro Datei erfolgt mit dem Subparameter LSR im Profilparameter VSAM bzw. im Makro NTVLSR im Natural-Parametermodul.

Der Subparameter LSR wird wie folgt angegeben:

```
LSR=(dd-name,subpool-number)
```

Das Makro NTVLSR wird wie folgt angegeben:

```
NTVLSR dd-name,subpool-number
```

Dabei ist:

Wert	Erklärung
<i>dd-name</i>	DD/DLBL/FCT-Name der VSAM-Datei, auf die zugegriffen werden soll. Es gibt keinen Standardwert.
<i>subpool-number</i>	Subpool-Nummer (ID) zwischen 0 und 255 bei z/OS; siehe auch die relevante IBM VSAM-Dokumentation. Es gibt keinen Standardwert.

Bis zu 200 logische Dateien sind möglich. Für jede Datei ist eine separate Definition mit dem Subparameter LSR bzw. dem Makro NTVLSR erforderlich.

Wenn ERROR=YES im Modul NVSMISC gesetzt ist, müssen alle mit dem Subparameter LSR bzw. dem Makro NTVLSR definierten Dateien zur Laufzeit per JCL definiert werden, andernfalls wird eine entsprechende Fehlermeldung bei der Natural-Initialisierung zurückgegeben.

Wenn Sie mit dem Subparameter LSR bzw. dem Makro NTVLSR Basisgruppen (Cluster) definiert haben, die Pfadeinträge enthalten, müssen alle Pfade auch mit dem Subparameter LSR bzw. dem Makro NTVLSR definiert werden.

Für Umgebungen ohne Pfade gilt: Wenn im VSAM-Katalog die Upgrade-Option aktiv ist und wenn mit dem Subparameter LSR bzw. dem Makro NTVLSR eine VSAM-Datei definiert wird und diese Datei Referenzen auf einen alternativen Index (AIX) enthält, müssen alle AIX-Dateien auch mit dem Subparameter LSR bzw. dem Makro NTVLSR definiert werden.

Natural for VSAM benutzt die entsprechenden VSAM-Kataloginformationen über die beteiligten Dateien, berechnet automatisch die optimale Pool-Größe und erstellt dann separate Subpools für Daten- und Index-Bestandteile.

Im Batch-Modus unter z/OS ordnet Natural for VSAM die Pools als ESO Hiperspace zu, wenn die folgenden Bedingungen erfüllt werden:

- Alle Größen im VSAM-Katalog sind mit mindestens als 4 KB oder einem Vielfachen dieses Wertes angegeben (dies gilt sowohl für Daten- als auch für Index-Bestandteile).
- Die Library, aus der Natural for VSAM geladen wurde, ist eine AFP-zugelassene Library.

Diese Bedingung ist nötig, um den Adressbereich als nicht verlagerbar (*non-swapable*) zu definieren, was eine Voraussetzung für ESO Hiperspaces ist.

### OPSUPP - Unterstützung dynamischer Open-Aufrufe

OPSUPP=*value* schaltet die Unterstützung mehrerer Open-Aufrufe innerhalb einer Session ein bzw. aus.

Wert	Erklärung
ON	Mehrere unterschiedliche Open-Aufrufe werden durch Aufrufen der Anwendungsprogrammierschnittstelle <code>USR2008N</code> unterstützt.
OFF	Mehrere unterschiedliche Open-Aufrufe innerhalb einer Session werden nicht unterstützt. Dies ist der Standardwert.

 **Anmerkung:** Informationen zu Anwendungsprogrammierschnittstellen siehe *SYSEXT Utility* in der Dokumentation *Debugger und Dienstprogramme*.

### MF – Multi-Fetch-Faktor

`MF=value` dient dazu, den Multi-Fetch-Faktor für VSAM-Ein-/Ausgaben über Natural Statements anzugeben.

Wert	Erklärung
1 - 28000	Multi-Fetch-Faktor, der bei Natural-Ein-/Ausgabe-Statements verwendet wird.
0	Dies ist der Standardwert.

 **Anmerkungen:**

1. Der Multi-Fetch-Faktor kann nur bei Batch-Verarbeitung angewendet werden.
2. Der Multi-Fetch-Faktor kann nur bei L1-, L2-, L3- und L9-Aufrufen angewendet werden.
3. Der maximale Multi-Fetch-Faktor für eine VSAM-Datei ist 28000.

### PATH - Unterstützung der Pfad-Verarbeitung

`PATH=value` dient dazu, einen Sekundärschlüssel als Pfad oder als native AIX-Datei zu handhaben.

Wert	Erklärung
ON	Alle in einer DDM definierten Sekundärschlüssel werden als Pfade für AIX-Dateien behandelt.
OFF	Alle Sekundärschlüssel werden als AIX-Dateien behandelt.
CHECK	Natural for VSAM prüft, ob die Sekundärschlüssel im VSAM-Katalog als Pfade oder als AIX definiert sind. Dies ist der Standardwert.

 **Anmerkung:** Falls Sie VSAM-Systemdateien `FSEC` und/oder `FSPool` verwenden, dürfen Sie `ON` nicht angeben; geben Sie entweder `OFF` oder `CHECK` an.

### PSIGNF - Unterstützung der Compiler-Option PSIGNF

PSIGNF=*value* dient dazu, die interne Darstellung positiver Vorzeichen gepackter Zahlen zu handhaben.

Wert	Erklärung
ON	Natural for VSAM unterstützt die Compiler-Option PSIGN für ein Natural-Objekt; die entsprechende DDM-Beschreibung im Feld ZONES wird nicht beachtet.
OFF	Natural for VSAM benutzt die DDM-Beschreibung im Feld ZONES. Dies ist der Standardwert.

### RETRY - Unterstützung des RETRY-Statement bei einer ON ERROR-Klausel

RETRY=(*value1*, *value2*) dient dazu, das RETRY-Statement bei den folgenden Natural for VSAM-Fehlermeldungen zu unterstützen:

- NAT3541 File :1:, control interval/record held by another user
- NAT3520 Held VSAM record modified by another user

Der erste Wert des Subparameters RETRY gilt für die Meldung NAT3541, der zweite für die Meldung NAT3520.

Dabei ist:

Wert	Erklärung
( <i>value1</i> , <i>value2</i> )	Jeder der beiden Werte kann entweder ON oder OFF sein.
(OFF, OFF)	Dies ist der Standardwert.

### RLS - Unterstützung des Record Level Sharing

RLS=*value* dient zur Unterstützung des Record Level Sharing (RLS) unter z/OS, DFSMS Version 1.6 oder höher.

Wert	Erklärung
ON	Alle Dateien werden im RLS-Modus geöffnet.
OFF	Alle Dateien werden im Nicht-RLS-Modus geöffnet (NSR, LSR). Dies ist der Standardwert.
CHECK	Alle Dateien werden überprüft, ob sie als SMS-verwaltete Datasets mit RLS-Optionen definiert sind; falls ja, wird die Datei im RLS-Modus geöffnet; falls nein, im Nicht-RLS-Modus.

**Anmerkungen:**

1. Dieser Subparameter gilt nur bei z/OS.
2. Wenn `TVS=ON` gesetzt ist (siehe Subparameter `TVS` weiter unten) und im Makro `NTVTVSD` (siehe oben) keine VSAM-Datei definiert worden ist, setzen Sie `RLS=CHECK`, um zu prüfen, ob die entsprechende VSAM-Datei als wiederherstellbarer Dataset definiert worden ist.

**ROLLSIZ - Größe des Bereichs für Session-Status-Informationen**

`ROLLSIZ=value` gibt die Größe des von Natural zum Speichern von internen Session-Status-Informationen benutzten Bereichs an, wenn eine Natural-Transaktion wegen des Endes eines TP-Monitor Tasks beendet wird.

Wert	Erklärung
0 oder 1 - 10000	Größe des Bereichs (in Bytes).
550	Dies ist der Standardwert.



**Anmerkung:** Dieser Subparameter gilt nur in einer Thread-Umgebung (CICS, Com-plete, Natural als Server).

**SFILE - Unterstützung von VSAM-Systemdateien**

`SFILE=value` wird benutzt, um VSAM-Systemdateien zu unterstützen.

Wert	Erklärung
ON	Unterstützung von VSAM-Systemdateien.
OFF	Keine Unterstützung von VSAM-Systemdateien. Dies ist der Standardwert.
CHECK	Es erfolgt eine Prüfung, ob die Natural-Systemdateien FNAT, FUSER und FDIC als Natural for VSAM Version 8.2 VSAM-Systemdateien mit der erforderlichen Schlüssellänge von 87 definiert sind.

**TAFE - Maximale Anzahl DDMs pro Natural-Session**

`TAFE=value` gibt die maximale Anzahl DDMs pro Natural-Session an.

Wert	Erklärung
0 oder 1 - 1000	Maximale Anzahl DDM-Felder.
10	Dies ist der Standardwert.



**Anmerkungen:**

1. Da es möglich ist, mehrere Deskriptoren in einer DDM zu definieren, hat der Subparameter TAFE Auswirkungen auf die Größe der FCT-, FWA-, OPV- und TAF-Puffer (siehe *Buffers for Memory Management*) in der *Natural for VSAM*-Dokumentation.
2. Wenn Sie den Wert von TAFE oder TAFN erhöhen, müssen Sie berücksichtigen, dass die zugordnete Speicherbereichsgröße nicht durch Addieren, sondern durch Multiplizieren dieser Werte berechnet wird.

**TAFN - Durchschnittliche Anzahl DDM-Felder**

TAFN=*value* gibt die durchschnittliche Anzahl DDM-Felder an, die in jedem Eintrag in der Tabelle von VSAM-Dateien enthalten sind, auf die zugegriffen wird.

Wert	Erklärung
0 oder 1 - 1000	Maximale Anzahl DDM-Felder: 0 - 1000.
50	Dies ist der Standardwert.



**Anmerkung:** Wenn Sie den Wert von TAFE oder TAFN erhöhen, müssen Sie berücksichtigen, dass die zugordnete Speicherbereichsgröße nicht durch Addieren, sondern durch Multiplizieren dieser Werte berechnet wird.

**TIMEOUT - Zeitbeschränkung in Sekunden für eine RLS-Anforderung**

TIMEOUT=*value* dient zur Unterstützung einer gemischten Umgebung mit RLS-/Nicht-RLS-Dateien unter z/OS CICS Version 5.3 oder höher in einer Natural for VSAM Session.

Wert	Erklärung
0 - 30	Zeitbeschränkung (in Sekunden).
0	Dies ist der Standardwert.



**Anmerkungen:**

1. Dieser Subparameter gilt nur für z/OS CICS Version 5.3 oder höher.
2. Natural und Natural for VSAM Version 6.2 sind plex-fähig; d.h., die Natural Session kann nacher einer Terminal-Ein-/Ausgabe durch den Workload Manager auf einem anderen z/OS in einem anderen CICS 5.3 fortgesetzt werden, vorausgesetzt die Ressourcen sind plex-fähig. Da dies bei Nicht-RLS-Dateien nicht der Fall ist, muss die Session im konversationellen Modus

laufen, sobald eine VSAM-Datei im Nicht-RLS-Modus geöffnet wird. Mit dem Subparameter `TIMEOUT` können Sie festlegen, dass Nicht-RLS-Dateien aus der Natural for VSAM FCT-Warteschlange gelöscht werden sollen. Wenn es für die betreffende Natural for VSAM Session keine weiteren Nicht-RLS-FCT-Einträge gibt, schaltet Natural for VSAM in den nicht-konversationellen Modus, was bedeutet, dass die z/OS Parallel Sysplex-Verarbeitung wieder möglich ist.

### TSAE - Maximale Anzahl verschachtelter READ- und FIND-Statements

`TSAE=value` dient dazu, die maximale Anzahl aller verschachtelten `READ`- und `FIND`-Statements anzugeben.

Wert	Erklärung
0 oder 1 - 100	Maximale Anzahl aller verschachtelten <code>READ</code> - und <code>FIND</code> -Statements.
10	Dies ist der Standardwert.

### TVS - Unterstützung von DFSMStvs

`TVS=value` dient dazu, die Unterstützung der DFSMS Transactional VSAM Services (DFSMStvs) zu ermöglichen.

Wert	Erklärung
ON	Unterstützung von DFSMStvs.
OFF	Keine Unterstützung von DFSMStvs. Dies ist der Standardwert.



#### Anmerkungen:

1. Dieser Subparameter gilt nur bei z/OS.
2. Wenn `TVS` auf `ON` gesetzt ist, werden die Subparameter `BTSUPP` und `ETSUPP` zwangsweise auf `ON` gesetzt. Der Subparameter `RLS` wird nur dann zwangsweise auf `ON` gesetzt, wenn `RLS` auf `OFF` gesetzt worden ist (`RLS=CHECK` wird nicht zwangsweise auf `ON` geändert).

### TVSD – Aktivierung der DFSMS Transactional VSAM Services

Die Aktivierung der DFSMS Transactional VSAM Services (DFSMStvs) erfolgt entweder durch Setzen des ACB-Parameters `RLSREAD` oder des JCL-Parameters `RLS`. Grundsätzlich öffnet Natural for VSAM alle VSAM-Dateien standardmäßig für die Ausgabe. Dieser Subparameter aktiviert die DFSMStvs durch Angabe des Lese-Integritätswertes des ACB-Parameters `RLSREAD`. Wenn Sie `RLSREAD` im Subparameter `TVSD` oder im Makro `NTVTVSD` angeben, um die DFSMStvs zu aktivieren, brauchen Sie die JCL nicht anzupassen.

Wenn Sie nur den VSAM-Subparameter `TVS=ON` setzen, ohne die entsprechende VSAM-Dateidefinition mit dem Subparameter `TVSD` oder dem Makro `NTVTVSD` anzugeben, müssen Sie die JCL wie unten beschrieben ändern. In diesem Fall müssen Sie den VSAM-Subparameter `RLS=CHECK` setzen.

Der Subparameter `TVSD` wird wie folgt angegeben:

```
TVSD=(dd-name,option)
```

Das Makro `NTVTVSD` wird wie folgt angegeben:

```
NTVTVSD dd-name,option
```

Dabei ist:

Wert	Erklärung
<i>dd-name</i>	DD/DLBL/FCT-Name der VSAM-Datei, auf die zugegriffen werden soll. Es gibt keinen Standardwert.
<i>option</i>	NRI - Keine Lese-Integrität („Dirty Read“). CR - Konsistentes Lesen. CRE - Explizites, konsistentes Lesen. Es gibt keinen Standardwert.

### UPDL - Größe der Aktualisierungstabelle

`UPDL=value` gibt die Größe der Tabelle an, die vom Natural/VSAM-Interface zum Speichern der aus Datensätzen zwecks anschließender Aktualisierung gelesenen Felder.

Wert	Erklärung
0 oder 1 - 500000	Größe der Tabelle (in Bytes).
8192	Bzw. 32768 wenn <code>SFILE=ON</code> gesetzt ist. Dies ist der Standardwert.



**Anmerkung:** Da die Datensätze von Natural zur Vermeidung von gegenseitigen Sperren nicht mit Hold gelesen werden, wird der Inhalt der `UPDL`-Tabelle benutzt, um zu prüfen, ob vor der Aktualisierungsaufforderung durch einen anderen Benutzer irgendwelche Änderungen erfolgt sind.

## Beispiel für VSAM-Parameter

---

```
VSAM=(RLS=ON,PATH=ON,KEYLGH=66,EXIT=(FILE1,EXIT1,400),EXIT=(FILE2,EXIT2))
```

## Beispiel für NTVSAM-Makro

---

```
NTVSAM RLS=ON,PATH=ON,KEYLGH=66  
NTVEXIT FILE1,EXIT1,400  
NTVEXIT FILE2,EXIT2
```



Dieser Natural-Profilparameter dient zum Einstellen der maximalen Größe des von Natural for VSAM benötigten Pufferbereichs.

<b>Mögliche Werte</b>	1 - 32767	Größe des Pufferbereichs in KB.
	0	Bei VSIZE=0 kann das Natural VSAM Interface nicht benutzt werden.
<b>Standard-Einstellung</b>	0	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	



#### Anmerkungen:

1. Dieser Natural-Profilparameter gilt nur, wenn das Natural VSAM Interface installiert ist.
2. Wenn Natural for VSAM installiert ist, werden die entsprechenden Pufferspeicher bei der Initialisierung der Natural-Session angefordert.
3. Wenn der angeforderte Speicherplatz nicht zur Verfügung steht, kann das Natural VSAM Interface nicht benutzt werden. Eine entsprechende Fehlermeldung informiert Sie bei der Initialisierung des Natural VSAM Interface darüber, welcher im Makro `NTVSAM` im VSAM-spezifischen Natural-Parametermodul angegebene Pufferspeicher nicht in den `VSIZE`-Bereich hinein passt. Sie können dann die Größe des `VSIZE`-Bereichs erhöhen.
4. Wenn Sie während einer Natural-Session keine VSAM-Unterstützung benötigen, wird empfohlen, Natural mit `VSIZE=0` aufzurufen, um den durch die Handhabung der ungenutzten Pufferspeicher verursachten zusätzlichen Aufwand zu vermeiden.



# 275

## WEBIO - Web I/O Interface Rendering

---

▪ WEBIO-Parameter-Syntax .....	846
▪ NTWEBIO-Makro-Syntax .....	846
▪ Schlüsselwort-Subparameter .....	847
▪ Beispiel für WEBIO-Parameter .....	848
▪ Beispiel für NTWEBIO-Macro .....	848

Dieser Natural-Profilparameter gibt Ihnen die Möglichkeit, die Umsetzung (das „Rendering“) bestimmter Merkmale der Anzeige des Natural Web I/O Interface in Bildschirmdarstellung durch den Webbrowser auf der Basis einer Stilvorlage („Style Sheet“) ein- und auszuschalten. Es entspricht dem Makro `NTWEBIO` im Natural-Parametermodul.

<b>Mögliche Werte</b>	Siehe <a href="#">WEBIO-Parameter-Syntax</a> .	
<b>Standard-Einstellung</b>	Siehe Standardwerte der <a href="#">Schlüsselwort-Subparameter</a> .	
<b>Dynamische Angabe</b>	ja	Der Parameter WEBIO kann nur dynamisch angegeben werden. Im Natural-Parametermodul wird das Makro NTWEBIO verwendet.
<b>Angabe innerhalb der Session</b>	nein	



**Anmerkungen:**

1. Standardmäßig ist die auf einer Stilvorlage basierende Wiedergabe der Nachrichtenzeile, der PF-Tasten und der Natural-Fensterobjekte ausgeschaltet.
2. Weitere Informationen siehe *Using Style Sheets* in der *Natural Web I/O Interface*-Dokumentation.

## WEBIO-Parameter-Syntax

Der Parameter WEBIO wird wie folgt angegeben:

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

Informationen zu Namen und Werten von Schlüsselwort-Subparametern siehe [Schlüsselwort-Subparameter](#).

## NTWEBIO-Makro-Syntax

Das Makro NTWEBIO wird wie folgt angegeben:

```
NTWEBIO ML=value, *
      KEYS=value, *
      WIN=value
```

Siehe [Schlüsselwort-Subparameter](#).

## Schlüsselwort-Subparameter

ML | KEYS | WIN

### ML - Nachrichtenzeile

ML=*value* dient zum Ein- bzw. Ausschalten der auf einer Stilvorlage basierenden Wiedergabe der Nachrichtenzeile.

Wert	Erklärung
ON	Die auf einer Stilvorlage basierende Umsetzung der Nachrichtenzeile ist eingeschaltet.
OFF	Die auf einer Stilvorlage basierende Umsetzung der Nachrichtenzeile ist ausgeschaltet.



**Anmerkung:** Siehe auch *Modifying the Message Line* in der *Natural Web I/O Interface*-Dokumentation.

### KEYS - PF-Tasten

KEYS=*value* dient zum Ein- bzw. Ausschalten der auf einer Stilvorlage basierenden Umsetzung der PF-Tasten.

Wert	Erklärung
ON	Die auf einer Stilvorlage basierende Umsetzung der PF-Tasten ist eingeschaltet.
OFF	Die auf einer Stilvorlage basierende Umsetzung der PF-Tasten ist ausgeschaltet.



**Anmerkung:** Siehe auch *Modifying the Style of the PF Key Buttons* in der *Natural Web I/O Interface*-Dokumentation.

### WIN - Fensterobjekte

WIN=*value* dient zum Ein- bzw. Ausschalten der auf einer Stilvorlage basierenden Umsetzung der Fensterobjekte.

Wert	Erklärung
ON	Die auf einer Stilvorlage basierende Umsetzung der Fensterobjekte ist eingeschaltet.
OFF	Die auf einer Stilvorlage basierende Umsetzung der Fensterobjekte ist ausgeschaltet.



**Anmerkung:** Siehe auch *Modifying the Natural Windows* in der *Natural Web I/O Interface*-Dokumentation.

## Beispiel für WEBIO-Parameter

---

```
WEBIO=(KEYS=ON,ML=ON)
```

## Beispiel für NTWEBIO-Macro

---

```
NTWEBIO KEYS=ON,ML=ON
```

Dieser Natural Profil- und Session-Parameter bestimmt, was geschehen soll, wenn ein angeforderter Datensatz von einem anderen Benutzer auf Hold gesetzt wurde.

<b>Mögliche Werte</b>	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 in den Hold-Status zu setzen, zur Ausgabe einer entsprechenden Fehlermeldung führt.	
	OFF	Eine Fehlermeldung wird ausgegeben, falls einer der gewünschten Datensätze nicht in den Hold-Status gesetzt werden kann.	
<b>Standard-Einstellung</b>	OFF		
<b>Dynamische Angabe</b>	ja		
<b>Angabe innerhalb der Session</b>	ja	Gültige Statements:	SET GLOBALS
		Gültiges Kommando:	GLOBALS
<b>Programmierschnittstelle (API)</b>	USR1005N	Siehe <i>SYSEXT Utility</i> in der Dokumentation <i>Debugger und Dienstprogramme</i> .	



#### Anmerkungen:

1. Dieser Natural Profil- und Session-Parameter gilt nur für Adabas-Datenbanken.
2. In einer Natural-Session kann der Profilparameter WH durch den Session-Parameter WH überschrieben werden.
3. 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 setzen. Weitere Informationen zur Hold-Verarbeitung siehe Adabas *Command Reference*-Dokumentation.
4. Unter Natural Security kann die Einstellung dieses Parameters durch die Session Parameters-Option des *Library Profile* überschrieben werden.

5. Wenn die Option `SKIP RECORDS IN HOLD` bei einem `READ-` oder `FIND-Statement` benutzt wird, dann wird der Datenbankzugriff immer so ausgeführt, als ob `WH=OFF` gesetzt ist. Falls ein Benutzer versucht, einen Datensatz zu lesen, der von einem anderen Benutzer in den Hold-Status versetzt worden ist, wird dieser Datensatz übersprungen und die Verarbeitung wird mit dem nächsten Datensatz in der Leseabfolge fortgesetzt. Es wird in diesem Fall keine Fehlermeldung ausgegeben.

# 277

## WORK - Arbeitsdatei-Zuweisungen

---

▪ WORK-Parameter-Syntax .....	852
▪ NETWORK-Makro-Syntax .....	853
▪ Schlüsselwort-Subparameter für alle Umgebungen .....	854
▪ Schlüsselwort-Subparameter für AM=STD in allen Umgebungen .....	859
▪ Schlüsselwort-Subparameter für AM=STD in z/OS-Umgebungen .....	861
▪ Schlüsselwort-Subparameter für AM=CICS .....	863
▪ Schlüsselwort-Subparameter für AM=COMP .....	864
▪ Schlüsselwort-Subparameter für AM=SMARTS .....	865

Dieser Natural-Profilparameter gestattet es Ihnen, die maximale Anzahl an Arbeitsdateien (Workfiles) festzulegen, die während der Session verwendet werden soll. Der Parameter WORK entspricht dem Makro NETWORK im Natural-Parametermodul.

<b>Mögliche Werte</b>	Siehe <a href="#">WORK-Parameter-Syntax</a> .	
<b>Standard-Einstellung</b>	Siehe Standardwerte der weiter unten beschriebenen Schlüsselwort-Subparameter. Je nach Zugriffsmethode und Umgebung kann es unterschiedliche Standard-Einstellungen geben.	
<b>Dynamische Angabe</b>	ja	Der Parameter WORK kann nur dynamisch angegeben werden. Im Natural-Parametermodul wird das Makro NETWORK verwendet.
<b>Angabe innerhalb der Session</b>	nein	



**Anmerkungen:**

1. Innerhalb einer Session können bis zu 32 logische Arbeitsdateien (durchnummeriert von 1 bis 32) verwendet werden.
2. Die für den Zugriff auf Arbeitsdateien in unterschiedlichen Umgebungen verwendeten Software-Komponenten werden als Zugriffsmethoden bezeichnet. Für die Dauer einer jeden Natural-Session kann jeder logischen Arbeitsdatei nur eine Zugriffsmethode zugewiesen werden. Die Zugriffsmethode wird mit dem Schlüsselwort-Subparameter AM festgelegt.
3. Bei z/OS unter TSO und im Batch-Modus brauchen Arbeitsdateien nicht in der JCL vordefiniert zu werden. Vorausgesetzt, sie werden mit dem Subparameter AM=STD definiert, können sie während der Session dynamisch durch ein Natural-Programm mittels des DEFINE WORK FILE-Statements oder der Anwendungsprogrammierschnittstelle (API) USR2021 (in der Library SYSEXT) zugewiesen werden.
4. Siehe auch *Print and Work File Handling with External Datasets in a Server Environment* in der *Operations-Dokumentation*).

## WORK-Parameter-Syntax

Mit dem Parameter WORK geben Sie zunächst eine oder mehrere logische Arbeitsdateinummern an und danach mehrere Schlüsselwort-Subparameter, mit denen Sie die Eigenschaften dieser Arbeitsdateien festlegen:

```
WORK=((work-file-numbers),keyword-subparameters,...)
```

Dabei ist:

Syntax-Element	Beschreibung
<i>work-file-numbers</i>	<p>Die Dateinummern müssen zuerst und in Klammern angegeben werden.</p> <ul style="list-style-type: none"> <li>■ Die Nummern können im Bereich von 1 bis 32 liegen.</li> <li>■ Die Reihenfolge der Nummern kann beliebig sein.</li> <li>■ Die Nummern müssen durch Kommas voneinander getrennt werden.</li> <li>■ Um einen Nummernbereich anzugeben, können Sie einen Bindestrich (-) verwenden.</li> </ul>
<i>keyword-subparameters</i>	<p>Die Schlüsselwort-Subparameter (die es für verschiedene Umgebungen gibt) werden weiter unten beschrieben.</p> <p>Falls für dieselbe Arbeitsdatei schon eine frühere (oder eine Standard-)Einstellung existiert, werden nur die Werte für den jeweils angegebenen Schlüsselwort-Subparameter überschrieben, alle übrigen Werte bleiben unverändert.</p>



**Anmerkung:** Der Profilparameter WORK kann mehrmals (abgetrennt durch ein Komma oder ein Leerzeichen) angegeben werden, um unterschiedliche Arbeitsdatei-Definitionen anzulegen.

### Beispiele:

```
WORK=((2,12,18),AM=STD,DEST='WORK**')
WORK=((1,3,6-11,15),AM=COMP,OPEN=INITOBJ,CLOSE=CMD)
```

## NETWORK-Makro-Syntax

Mit einem Makro NETWORK geben Sie zunächst eine oder mehrere logische Arbeitsdateinummern an und danach mehrere Schlüsselwort-Subparameter, mit denen Sie die Eigenschaften dieser Arbeitsdatei(en) festlegen:

```
NETWORK (work-file-numbers),keyword-subparameters,...
```

Dabei ist:

Syntax-Element	Beschreibung
<i>work-file-numbers</i>	<p>Die Dateinummern müssen zuerst und in Klammern angegeben werden.</p> <ul style="list-style-type: none"> <li>■ Die Nummern können im Bereich von 1 bis 32 liegen.</li> <li>■ Die Reihenfolge der Nummern kann beliebig sein.</li> <li>■ Die Nummern müssen durch Kommas voneinander getrennt werden.</li> <li>■ Um einen Nummernbereich anzugeben, können Sie einen Bindestrich (-) verwenden.</li> </ul>
<i>keyword-subparameters</i>	<p>Die Schlüsselwort-Subparameter (die es für verschiedene Umgebungen gibt) werden weiter unten beschrieben.</p> <p>Falls für dieselbe Arbeitsdatei schon eine frühere (oder eine Standard-)Einstellung existiert, werden nur die Werte für den jeweils angegebenen Schlüsselwort-Subparameter überschrieben, alle übrigen Werte bleiben unverändert.</p>



**Anmerkung:** Das Makro NETWORK kann mehrmals angegeben werden, um unterschiedliche Arbeitsdatei-Definitionen anzulegen.

**Beispiele:**

```
NETWORK ( 2,12,18 ),AM=STD,DEST='WORK**'
NETWORK ( 1,3,6-11,15 ),AM=COMP,OPEN=INITOBJ,CLOSE=CMD
```

## Schlüsselwort-Subparameter für alle Umgebungen

Die folgenden Schlüsselwort-Subparameter stehen für alle Umgebungen zur Verfügung:

[AM](#) | [DEST](#) | [OPEN](#) | [CLOSE](#) | [LRECL](#) | [TRUNC](#) | [PAD](#) | [PADCHRO](#) | [PADCHRI](#)

### AM - Typ der Zugriffsmethode

Mit *AM=value* geben Sie den Typ der Zugriffsmethode an, die verwendet werden soll.

Wert	Zugriffsmethode
STD	Sequenzielle Standard-Dateien (Batch, TSO).
COMP	Com-plete-Arbeitsdateien.
SMARTS	SMARTS-Arbeitsdateien. Arbeitsdateien in einem SMARTS Portable File System (PFS).
CICS	Transiente CICS-Daten oder temporärer Speicher.
PC	Entire Connection.
USER	Drittanbieter-Arbeitsdatei-Interface.

Wert	Zugriffsmethode
OFF	Nicht zugewiesen. Keine automatische Zuweisung, falls FAMSTD=OFF gesetzt ist.  <b>Anmerkung:</b> WORK=OFF ist gleichbedeutend mit WORK=(( 1 - 32 ) , AM=OFF). Es hat keine Auswirkung auf Einstellungen der anderen Schlüsselwort-Subparameter.
0	Nicht zugewiesen. Automatische Zuweisung, falls FAMSTD=OFF gesetzt ist.  Dies ist die Standard-Einstellung.



**Anmerkungen:**

1. Für eine Online-Session muss allen zu verwendenden Arbeitsdateien eine spezifische Zugriffsmethode zugewiesen werden.
2. Für eine Batch-Session werden alle Arbeitsdateien, denen keine spezifische Zugriffsmethode zugewiesen wurde, automatisch festgestellt. Ihnen wird dann die Standard-Zugriffsmethode (AM=STD) zugewiesen, vorausgesetzt sie wurden in der JCL vordefiniert. Siehe auch [FAMSTD - Überschreiben der Arbeitsdatei-Zugriffsmethodenzuweisungen](#).

**DEST - Name des externen Dataset**

DEST=*value* gibt den externen Dataset-Namen an.

Wert	Erklärung
1 - 8 Zeichen oder 1 - 7 Zeichen, je nach Zugriffsmethode und Umgebung.	Name des externen Dataset.  <b>Anmerkung:</b> Der Subparameter DEST entspricht dem <i>operand1</i> des Statements DEFINE WORK FILE. Der Subparameterwert kann durch eine DEFINE WORK FILE-Angabe überschrieben werden.

Die Bedeutung des Subparameters DEST ist abhängig von der Zugriffsmethode:

Zugriffsmethode	Bedeutung des Schlüsselwort-Subparameters DEST
AM=STD	DEST ist der logische Dataset-Name (DDNAME, LINK-Name, DTF-Name).  <b>Anmerkung:</b>  1. Wenn das Ziel für mehrere Dateien gelten soll, müssen zwei Sterne (**) für die Dateinummer angegeben werden. Diese werden für jede Arbeitsdatei durch die entsprechende logische Dateinummer ersetzt. Ein DEST-Wert, der zwei Sterne enthält, muss in Hochkommas gesetzt werden, wenn er als dynamischer Parameter verwendet werden soll.  2. Die Standard-Einstellung ist DEST='CMWKF**' für z/OS-Umgebungen.

AM=CICS	<p>Für Arbeitsdateien unter CICS gibt es keine Standard-Einstellung. Hier ist der Subparameter DEST zwingend erforderlich, d.h., dass CICS-Arbeitsdateien ohne gültige DEST-Angabe ignoriert werden.</p> <p>Das Natural CICS Interface unterstützt außerdem eine Variable (siehe TERMVAR=&amp;TID-Parameter im Generierungsmakro NTCICSP, als Teil des DEST-Werts, der bei der Angabe durch die tatsächliche CICS-Terminalkennung ersetzt wird; siehe auch <i>Natural Print and Work Files under CICS</i> in der <i>TP Monitor Interfaces</i>-Dokumentation).</p>
AM=COMP	<p>DEST definiert den Namen der Com-plete SD-Datei. Die Länge ist auf maximal 8 Zeichen begrenzt.</p> <p><b>Anmerkung:</b></p> <ol style="list-style-type: none"> <li>1. Wenn die Datei mit TYPE=TID definiert wird, dann wird an den DEST-Wert der Com-plete Stack Level angehängt. Die Länge ist entsprechend eingeschränkt auf 7 Zeichen.</li> <li>2. SD-Dateinamen, die mit &amp;&amp; beginnen, werden als temporäre Dateien behandelt, die beim Beenden von Natural automatisch gelöscht werden.</li> </ol>

### OPEN - Zeitpunkt der Datei-Öffnung

Mit OPEN=*value* bestimmen Sie den Zeitpunkt der Datei-Öffnung.

Wert	Die Datei wird geöffnet ...
INIT	für die Ausgabe bei der Initialisierung der Session.
OBF	gemäß der OPEN-Standard-Einstellung für unterschiedliche Umgebungen (Batch, CICS, Com-plete, TSO).
OBJ	<p>wenn die Ausführung des ersten Objekts, das auf die Arbeitsdatei zugreift, beginnt.</p> <p>Dies ist der Standardwert.</p>
INITOBF	für die Ausgabe bei der Initialisierung der Session. Bei jeder anschließenden erneuten Öffnung der Datei wird die OPEN-Standard-Einstellung für unterschiedliche Umgebungen (Batch, CICS, Com-plete, TSO) gesetzt.
OBJ1	wenn die Ausführung des ersten Objekts auf Ebene 1, das auf die Arbeitsdatei zugreift, beginnt. Andernfalls wird sie geöffnet, wenn zum ersten Mal auf sie zugegriffen wird.
ACC	wenn zum ersten Mal durch ein Statement auf sie zugegriffen wird.
INITOBJ	für die Ausgabe bei der Initialisierung der Session. Jede anschließende erneute Öffnung der Datei wird durchgeführt, wenn die Ausführung des ersten Objekts, das auf die Arbeitsdatei zugreift, beginnt.
INITOBJ1	wenn die Ausführung des ersten Objekts auf Ebene 1, das auf die Arbeitsdatei zugreift, beginnt. Andernfalls wird sie geöffnet, wenn zum ersten Mal auf sie zugegriffen wird.
INITACC	für die Ausgabe bei der Initialisierung der Session. Jede anschließende erneute Öffnung der Datei wird durchgeführt, wenn zum ersten Mal durch ein Statement auf sie zugegriffen wird.

## CLOSE - Zeitpunkt der Datei-Schließung

Mit `CLOSE=value` bestimmen Sie den Zeitpunkt der Datei-Schließung.

Wert	Die Datei wird geschlossen ...
OBJ	entweder, wenn die Verarbeitung des Objekts, in dem zuerst auf die Datei zugegriffen wurde, beendet ist, oder, wenn der Kommandomodus, NEXT- oder MAINMENU-Modus erreicht ist.
CMD	wenn der Kommandomodus, NEXT- oder MAINMENU-Modus erreicht ist.  Dies ist die Standard-Einstellung.
FIN	beim Session-Ende.  <b>Anmerkung:</b> Bei <code>CLOSE=FIN</code> bewirkt ein <code>DEFINE WORK FILE</code> -Statement einen Fehler, wenn die Arbeitsdatei bereits geöffnet war. Ein <code>CLOSE WORK FILE</code> -Statement für die Arbeitsdatei wird ignoriert. Wenn die Datei-Ende-Bedingung während der Ausführung des <code>READ WORK FILE</code> -Statements auftritt, schließt Natural die Arbeitsdatei sofort.
USER	Dieser Wert gibt an, dass die Arbeitsdatei nur dann geschlossen wird, wenn die Datei offen ist und eine der folgenden Bedingungen zutrifft: <ul style="list-style-type: none"> <li>■ ein <code>CLOSE WORK FILE</code>-Statement wird abgesetzt,</li> <li>■ ein <code>DEFINE WORK FILE</code>-Statement wird abgesetzt,</li> <li>■ Session-Ende.</li> </ul>

## LRECL - Standard- und Maximal-Datensatzlänge des Dataset

Mit `LRECL=value` bestimmen Sie die Datensatzlänge des Dataset.

Wert	Erklärung
0 oder 5 - 32767	Datensatzlänge in Bytes.
0	Dies ist der Standardwert.



**Anmerkung:** Dieser Subparameter dient insbesondere zur Überprüfung auf Trunkierung und Auffüllung. Weitere Informationen zu `AM=STD` siehe Subparameter `LRECL` im Abschnitt [Schlüsselwort-Subparameter für AM=STD in allen Umgebungen](#).

### TRUNC - Trunkierung von Ausgabe-Datensätzen

Mit `TRUNC=value` bestimmen Sie, ob die Ausgabe-Datensätze trunziert werden oder nicht.

Wert	Erklärung
ON	Ausgabe-Datensätze, die länger sind als die Datensatzlänge (LRECL) des Dataset, werden trunziert.
OFF	Der Fehler NAT1512 wird ausgegeben, wenn ein Ausgabe-Datensatz länger ist als die Datensatzlänge des Dataset.  Dies ist die Standard-Einstellung.

### PAD - Auffüllen von Ausgabe-Datensätzen

Mit `PAD=value` bestimmen Sie, ob die Ausgabe-Datensätze aufgefüllt werden oder nicht (gilt nur bei Datasets mit fester Datensatzlänge).

Wert	Erklärung
ON	Ausgabe-Datensätze, die kürzer sind als die Datensatzlänge (LRECL) des Dataset, werden aufgefüllt. Die mit Auffüllzeichen werden mit dem Schlüsselwort-Subparameter <code>PADCHRO</code> definiert.  Dies ist der Standardwert.
OFF	Der Fehler NAT1510 wird ausgegeben, wenn ein Ausgabe-Datensatz kürzer ist als die Datensatzlänge des Dataset.

### PADCHRO - Zeichen zum Auffüllen von Ausgabe-Datensätzen

Mit `PADCHRO=value` definieren Sie das Zeichen, das zum Auffüllen von Ausgabe-Datensätzen verwendet wird, wenn für die Arbeitsdatei `PAD=ON` angegeben wurde.

Wert	Erklärung
'x' oder x'xx'	Ein Zeichen <i>x</i> in einzelnen Hochkommas oder ein Hex-Zeichen <i>xx</i> .
x'00'	Dies ist der Standardwert.

## PADCHRI - Zeichen zum Auffüllen von Eingabe-Datensätzen

Mit `PADCHRO=value` definieren Sie das Zeichen, das zum Auffüllen von Eingabe-Datensätzen verwendet wird.

Wert	Erklärung
'x' oder x'xx'	Ein Zeichen <i>x</i> in einzelnen Hochkommas oder ein Hex-Zeichen <i>xx</i> .
x'40'	Leerzeichen.  Dies ist der Standardwert.

## Schlüsselwort-Subparameter für AM=STD in allen Umgebungen

Die folgenden Schlüsselwort-Subparameter stehen für `AM=STD` in allen Umgebungen zur Verfügung:

[RECFM](#) | [BLKSIZE](#) | [LRECL](#)

### RECFM - Standard-Datensatzformat des Dataset

Mit `RECFM=value` bestimmen Sie das Standard-Datensatzformat des Dataset.

Unterstützte Formate:

Format	Erklärung
F	Fest
V	Variable
U	Undefiniert
B	In Blöcke unterteilt
S	Segmentiert
A	ASA
M	Machinensteuerzeichen

Folgende Werte und Wertekombinationen sind möglich:

Wert	Erklärung
F, FA, FM, FB, FBA, FBM, V, VA, VM, VB, VBA, VBM, VBS, VBSA, VBSM, U, UA, UM	Diese Werte bzw. Wertekombinationen können angegeben werden.
VB	In variable Blöcke unterteilt. Dies ist der Standardwert.

 **Anmerkung:** Die RECFM-Angabe gilt nur, wenn in der JCL oder (nur bei z/OS) im Dataset-DCB kein Datensatzformat vordefiniert ist.

### BLKSIZE - Standard-Block-Größe des Dataset

Mit `BLKSIZE=value` bestimmen Sie die Standard-Blockgröße des Dataset.

Mögliche Werte:

Wert	Erklärung
0 or 8 - 32767	Standard-Blockgröße in Bytes.
4628	Dies ist der Standardwert.

 **Anmerkung:** Die BLKSIZE-Angabe gilt nur, wenn in der JCL oder (nur bei z/OS) im Dataset-DCB keine Blockgröße vordefiniert ist.

### LRECL - Standard- und Maximum-Datensatzlänge des Dataset

Mit `LRECL=value` bestimmen Sie die Standard-Datensatzlänge des Dataset.

Mögliche Werte:

Wert	Erklärung
0 or 5 - 32767	Datensatzlänge des Dataset in Bytes.
0	Dies ist der Standardwert.

 **Anmerkungen:**

1. Dieser Subparameter wird insbesondere zur Prüfung bei der Trunkierung und Auffüllung genutzt.
2. Für `RECFM=V(B)` enthält der LRECL-Wert ein 4-Byte-Datensatz-Deskriptorwort.

3. Bei Angabe von `LRECL=0` gilt folgendes: Bei `RECFM=V (B)` nimmt `LRECL` den Standardwert `BLKSIZE-4` an. Bei `RECFM=U` nimmt `LRECL` den Standardwert `BLKSIZE` an. Bei `RECFM=F (B)` wird die maximale Datensatzlänge in dem momentan ausgeführten Natural-Programm genommen, wenn die Datei geöffnet wird, z.B. führt dies bei `OPEN=INIT` zu einem Fehler.
4. Die `LRECL`-Angabe gilt nur, wenn in der JCL oder (nur bei z/OS) in der Dataset-DCB keine Datensatzlänge vordefiniert ist.

## Schlüsselwort-Subparameter für `AM=STD` in z/OS-Umgebungen

Die folgenden Schlüsselwort-Subparameter stehen nur für `AM=STD` in z/OS-Umgebungen zur Verfügung:

`REREAD` | `FREE` | `BUFNO` | `DISP` | `VMAX`

### **REREAD - Schließung von Banddatei-Datasets**

Mit `REREAD=value` setzen Sie die `REREAD`-Option für das Schließen der Banddatei.

Wert	Erklärung
ON	Die <code>REREAD</code> -Option wird für <code>CLOSE SVC</code> gesetzt. Damit wird das Volume repositioniert, um den Dataset erneut zu verarbeiten.  Dies ist der Standardwert.
OFF	Die <code>REREAD</code> -Option wird für <code>CLOSE SVC</code> nicht gesetzt.

### **FREE - Aufhebung der Dataset-Zuordnung beim Schließen der Datei**

Mit `FREE=value` bestimmen Sie, ob die Zuordnung des Dataset aufgehoben wird, wenn die Datei geschlossen wird.

Wert	Erklärung
ON	Die <code>FREE</code> -Option wird für <code>CLOSE SVC</code> gesetzt. Dadurch wird die Zuordnung des Dataset aufgehoben, wenn er geschlossen wird (und nicht bei der Beendigung des Schritts).
OFF	Die <code>FREE</code> -Option wird für <code>CLOSE SVC</code> nicht gesetzt.  Dies ist der Standardwert.

## BUFNO - Standard-Anzahl der z/OS-E/A-Pufferspeicher des Dataset

Mit `BUFNO=value` legen Sie die Standard-Anzahl der z/OS-E/A-Pufferspeicher des Dataset fest.

Wert	Erklärung
0 oder 1 - 255	Standard-Anzahl der z/OS-E/A-Pufferspeicher des Dataset.
0	Bei <code>BUFNO=0</code> weist z/OS standardmäßig fünf E/A-Pufferspeicher zu. Dies ist der Standardwert.



### Anmerkungen:

1. Die Anzahl der E/A-Pufferspeicher kann die Performance beim Zugriff auf Arbeitsdatei drastisch verbessern. Beachten Sie, dass die Speicherung für E/A-Pufferspeicher unterhalb der 16 MB-Grenze zugewiesen wird.
2. Die `BUFNO`-Angabe gilt nur, wenn der `BUFNO`-Parameter nicht in der JCL für den Dataset angegeben ist.

## DISP - Arbeitsdatei für Änderung öffnen

Mit `DISP=value` geben Sie an, dass die Arbeitsdatei zur Änderung geöffnet wird.

Wert	Erklärung
MOD	Neue Datensätze werden am Ende der Arbeitsdatei hinzugefügt. <b>Anmerkung:</b> Das entspricht dem Parameter <code>DISP=MOD</code> im JCL DD Statement.
NOMOD	Die Arbeitsdatei wird von Beginn an neu geschrieben. Dies ist der Standardwert.

## VMAX - LRECL-Einstellung für variables Datensatzformat steuern

`VMAX=value` steuert die `LRECL`-Einstellung für eine Ausgabedatei mit variablem Datensatzformat (`RECFM=V`).

Wert	Erklärung
ON	Vorausgesetzt, dass für die Datei ein <code>BLKSIZE</code> -Wert existiert, der nicht Null ist, wird mit <code>VMAX=ON</code> für variables Datensatzformat <code>LRECL=BLKSIZE-4</code> gesetzt, unabhängig von der Einstellung in der DCB oder im <code>LRECL</code> -Subparameter.
NAT	<code>LRECL</code> wird auf die Länge +4 des größten Datensatzes im Anwendungsprogramm gesetzt, wenn dieser Wert kleiner ist als im <code>LRECL</code> in der DCB für den Dataset.
OFF	Es wird die <code>LRECL</code> -Einstellung aus der DCB für den Dataset verwendet. Dies ist der Standardwert.

## Schlüsselwort-Subparameter für AM=CICS

Die folgenden Schlüsselwort-Subparameter stehen nur für **AM=CICS** zur Verfügung:

**TYPE** | **DISP**

### TYPE - Typ des CICS-Speichermediums

TYPE=*value* gibt den zu verwendenden Typ des CICS-Speichermediums an.

Wert	Erklärung
MAIN	Temporärer Hauptspeicher.
AUX	Temporärer Hilfsspeicher.
TD	Transiente Daten.

### Standardwert

Der verwendete Standardwert hängt von der Einstellung des Subparameters **DEST** ab. Wenn der Wert des Subparameters **DEST** mit einer gültigen transienten CICS-Datenwarteschlange übereinstimmt, nimmt der Subparameter **TYPE** die Standard-Einstellung **TD** an, andernfalls wird **MAIN** als Standard-Einstellung angenommen.

### DISP - Temporäre CICS-Speicherwarteschlangen-Disposition

DISP=(*value1*, *value2*) gibt die temporäre CICS-Speicherwarteschlangen-Disposition an.

Wertepaar	Erklärung
(NEW, KEEP)	Die Speicherwarteschlange wird gelöscht, wenn die Datei geöffnet wird. Dies ist der Standardwert.
(NEW, DELETE)	Die Speicherwarteschlange wird gelöscht, wenn die Datei geöffnet und wenn sie geschlossen wird.
(OLD, DELETE)	Die Speicherwarteschlange wird gelöscht, wenn die Datei geschlossen wird.
(OLD, KEEP)	Die Speicherwarteschlange wird nicht gelöscht.



**Anmerkung:** Die **DISP**-Angabe gilt nicht für CICS-Temporärdaten-Warteschlangen außerhalb der Partition.

## Schlüsselwort-Subparameter für AM=COMP

Die folgenden Schlüsselwort-Subparameter stehen nur für **AM=COMP** zur Verfügung:

**TYPE** | **BLOCKS** | **BLKSIZE**

### TYPE - Art des Speicherzugriffs

TYPE=*value* gibt die zu verwendende Art des Speicherzugriffs an.

Wert	Erklärung
SHR	Gemeinsam genutzter Zugriff, d.h. auf die Arbeitsdatei können alle Benutzer zugreifen.
TID	Die Arbeitsdatei steht nur für die aktuelle Com-plete-Terminalkennung zur Verfügung.
DYN	Die Arbeitsdatei steht nur für die aktuelle Terminal-Stack-Ebene zur Verfügung.

### BLOCKS - Anzahl der Speicherblöcke

BLOCKS=*value* gibt die Anzahl der zuzuweisenden Speicherblöcke an.

Mögliche Werte:

Wert	Erklärung
1 - 9999	Anzahl der zuzuweisenden Speicherblöcke.
20	Dies ist der Standardwert.

### BLKSIZE - Größe der Speicherblöcke

BLKSIZE=*value* gibt die Größe der Speicherblöcke des Dataset an.

Wert	Erklärung
0 oder 8 - 32767	Größe der Speicherblöcke in Bytes.
4628	Dies ist der Standardwert.

## Schlüsselwort-Subparameter für AM=SMARTS

Die folgenden Schlüsselwort-Subparameter stehen nur für **AM=SMARTS** zur Verfügung:

**DEST** | **TYPE** | **DISP**

### DEST - Name der Arbeitsdatei

`DEST=value` gibt den Namen der Arbeitsdatei an.

Wert	Erklärung
1 - 8 Zeichen.	Name der Arbeitsdatei.



#### Anmerkungen:

1. Da die `DEST`-Klausel auf maximal 8 Zeichen eingeschränkt ist, ist es nutzlos, eine Datei mit absoluter PFS-Pfadangabe zu definieren.
2. Der in der `DEST`-Klausel angegebene Name bezieht sich auf das Wurzelverzeichnis der Arbeitsdatei. Dieses Verzeichnis wird mit der Umgebungsvariablen `NAT_WORK_ROOT` angegeben.
3. Zur Angabe einer Datei mit absoluter Pfaddefinition muss das Statement `DEFINE WORK FILE` verwendet werden.

### TYPE - Art des Speicherzugriffs

`TYPE=value` gibt die zu verwendende Art des Speicherzugriffs an.

Wert	Erklärung
BIN	Jede Zeile wird ohne abschließendes Zeilenendezeichen in die Arbeitsdatei geschrieben. Dies ist der Standardwert.
TXT	Jede Zeile wird mit einem abschließenden Zeilenendezeichen (x'15') in die Arbeitsdatei geschrieben.

## DISP - Öffnungsmodus der Datei

DISP=(value1,value2,value3) gibt den Öffnungsmodus der Arbeitsdatei an.

Wert	Erklärung	
value1	value1 gibt an, ob eine vorhandene Datei gelöscht werden soll oder ob neue Daten an die Datei angehängt werden sollen.	
	NEW	Eine vorhandene Datei wird gelöscht, wenn die Datei zum Schreiben geöffnet wird.  Dies ist der Standardwert.
	OLD oder MOD	Neue Daten werden am Ende der Datei angehängt.
value2	value2 gibt an, ob eine Datei nach dem Zugriff beibehalten oder entfernt werden soll.	
	KEEP	Permanente Datei, die nach dem Schließen beibehalten wird.  Dies ist der Standardwert.
	DELETE	Temporäre Datei, die nach dem Schließen entfernt wird.
value3	value3 gibt an, ob ein Benutzer exklusiven Zugang zur Datei hat oder nicht.	
	SHR	Gemeinsam genutzter Zugang, d.h. die Arbeitsdatei ist für alle Benutzer zugänglich.  Dies ist der Standardwert.
	OWN	Exklusiver Zugang, die Arbeitsdatei ist für die aktuelle Complete-Benutzerkennung zugänglich. Dateien mit exklusivem Zugang befinden sich in einem zusätzlichen Verzeichnis, das den Namen der aktuellen Benutzerkennung.

### Beispiel 1:

DISP=(NEW,KEEP,SHR)

### Beispiel 2:

Wenn Sie nur den ersten Wert (mit oder ohne Klammern) angeben, wird den übrigen Werten jeweils die Standardeinstellung zugewiesen:

DISP=(MOD)

oder

DISP=MOD

Beide Angaben entsprechen der folgenden Angabe:

DISP=(MOD,KEEP,SHR)



Dieser Natural-Profilparameter dient zur Angabe der Größe der Natural-Arbeitsspeicher unterhalb und oberhalb der 16 MB-Grenze für die Dauer einer Natural-Session.

<b>Mögliche Werte</b>	Siehe <i>Syntax-Beschreibung</i> .	
<b>Standard-Einstellung</b>	(32,128,2097151,2097151)	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	



#### Anmerkungen:

1. Natural verwendet Arbeitsspeicher unterhalb und oberhalb der 16 MB-Grenze. Innerhalb dieser Arbeitsspeicher werden alle temporären physischen Speicheranforderungen bedient.
2. Physischen Speicher verwendet Natural nur in besonderen Situationen zur Übergabe von Parameterbereichen außerhalb des Thread (während der Thread freigegeben wird) während der Ausführung des CALL-Statements mit der Call-by-Value-Option, die unter CICS mit einem Statement SET CONTROL 'P=V' angegeben wird.
3. Der Vorteil der Arbeitsspeicher besteht darin, dass Natural im Falle von vielen Anforderungen von physischem Speicher diese Anforderungen selbst erfüllen kann, anstatt sie an das Betriebssystem zu übergeben.

#### Syntax-Beschreibung

Der Parameter WPSIZE wird wie folgt angegeben:

`WPSIZE=(size-below,size-above,maximum-below,maximum-above)`

Dabei ist:

Syntax-Element	Erklärung
<i>size-below</i>	<i>size-below</i> (0-1024) ist die Größe eines Natural-Arbeitsspeichers in KB unterhalb der 16 MB-Grenze. Der Wert 0 bedeutet, dass kein Arbeitsspeicher zugewiesen wird, d.h., alle Anforderungen nach physischem Speicherplatz unterhalb der 16 MB werden direkt an das Betriebssystem übergeben.
<i>size-above</i>	<i>size-above</i> (0-16384) ist die Größe eines Natural-Arbeitsspeichers in KB oberhalb der 16 MB-Grenze. Der Wert 0 bedeutet, dass kein Arbeitsspeicher zugewiesen wird, d.h., alle Anforderungen nach physischem Speicherplatz oberhalb der 16 MB werden direkt an das Betriebssystem übergeben.
<i>maximum-below</i>	<i>maximum-below</i> (0-2097151) begrenzt den gesamten physischen Speicher in KB, der unterhalb der 16 MB-Grenze zugewiesen werden kann. Der Wert 0 bedeutet, dass kein physischer Speicher unterhalb der 16 MB-Grenze zugewiesen werden kann.
<i>maximum-above</i>	<i>maximum-above</i> (0-2097151) begrenzt den gesamten physischen Speicher in KB, der oberhalb der 16 MB-Grenze zugewiesen werden kann. Der Wert 0 bedeutet, dass kein physischer Speicher oberhalb der 16 MB-Grenze zugewiesen werden kann.



**Anmerkungen:**

1. Wenn ein Arbeitsspeicher aufgebraucht ist, wird ein weiterer Arbeitsspeicher mit derselben Größe zugewiesen.
2. Wenn die Größe des angeforderten physischen Speichers die Größe des angegebenen Arbeitsspeichers übersteigt, erfolgt eine `GETMAIN`-Anforderung nach mehr Speicherplatz.
3. Subparameter, die nicht geändert werden sollen, können weggelassen werden, z.B. können Sie `WPSIZE=(,1000)` angeben, wenn Sie die Größe des Natural-Arbeitsspeicher nur oberhalb der 16 MB-Grenze auf 1000 KB setzen wollen.
4. Natural weist die Arbeitsspeicher außerhalb des Natural-Speicher-Thread gemäß den angegebenen Einstellungen zu. Ein Arbeitsspeicher wird während der ersten Anforderung von physischem Speicher zugewiesen und wird während der nächsten Terminal-Ein-/Ausgabe freigegeben.
5. Für Umgebungen ohne Thread (z.B. Batch, TSO) lautet die empfohlene Einstellung `WPSIZE=(0,0)`. Damit kann virtueller Speicherplatz eingespart werden. Ausnahme: Dies gilt nicht, wenn der Natural zIIP Enabler installiert und aktiv ist.
6. Wenn der Natural zIIP Enabler installiert und aktiv ist (nur bei z/OS Batch und TSO), kann bei Wahl einer geeigneten Einstellung des Profilparameters `WPSIZE` durch Verringerung der Anzahl an physischen `GETMAIN`-Anweisungen die Anzahl der Umschaltungen in den TCB-Modus verringert werden. Das gilt ebenso, wenn der Profilparameter `THSIZE` benutzt wird.
7. Hinsichtlich der Bedeutung des Profilparameters `THSIZE` im Zusammenhang mit Natural for zIIP siehe Abschnitt *Thread Size Specification - THSIZE Parameter* im Abschnitt *Tuning zIIP Usage* in der *Natural for zIIP* -Dokumentation.

# 279

## WSISIZE - Puffer für das Natural Workstation Interface

---

Dieser Natural-Profilparameter ist nur gültig, wenn das Natural Workstation Interface installiert ist

<b>Mögliche Werte</b>	10 - 256	Größe des Pufferbereichs in KB. <b>Anmerkung:</b> Falls der angeforderte Speicherplatz nicht vorhanden ist, kann das Natural Workstation Interface nicht benutzt werden.
	0	Das Natural Workstation Interface kann nicht benutzt werden.
<b>Standard-Einstellung</b>	0	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	



**Anmerkung:** Alternativ können Sie den äquivalenten Natural-Profilparameter [DS](#) oder das Parametermakro [NTDS](#) benutzen.



# 280

## XML - Statements PARSE XML und REQUEST

### DOCUMENT aktivieren/deaktivieren

---

▪ XML-Parameter-Syntax .....	874
▪ NXML-Makro-Syntax .....	875
▪ Schlüsselwort-Subparameter .....	875
▪ Beispiel für XML Parameter .....	881
▪ Beispiel für NXML Makro .....	881

Dieser Natural-Profilparameter dient zum Aktivieren/Deaktivieren der Statements `REQUEST DOCUMENT` und `PARSE XML`. Dieser Profilparameter entspricht dem Makro `NTXML` im Natural-Parametermodul.

<b>Mögliche Werte</b>	Siehe <a href="#">XML-Parameter-Syntax</a> .	
<b>Standard-Einstellung</b>	XML=(OFF)	
<b>Dynamische Angabe</b>	ja	Der Profilparameter XML kann nur dynamisch angegeben werden. Im Natural-Parametermodul wird das Makro <code>NTXML</code> verwendet.
<b>Angabe innerhalb der Session</b>	nein	



#### Anmerkungen:

1. Als Voraussetzung für die Benutzung des Profilparameters XML muss der Profilparameter `CFICU` auf `CFICU=ON` gesetzt werden.
2. Siehe auch *Statements für den Internet- und XML-Zugriff* im *Leitfaden zur Programmierung*.

## XML-Parameter-Syntax

Der Profilparameter XML wird wie folgt angegeben:

```
XML=(ON,keyword-subparameter=value,keyword-subparameter=value,...)
```

Oder:

```
XML=(OFF,keyword-subparameter=value,keyword-subparameter=value,...)
```

Dabei ist:

Syntax-Element / Wert	Erklärung
ON	Aktivierung des XML-Supports gemäß der <a href="#">Subparameter</a> -Einstellungen weiter unten.  Für die Verwendung der Statements <code>PARSE XML</code> und <code>REQUEST DOCUMENT</code> müssen auch die Subparameter <code>RDOC</code> und <code>PARSE</code> auf <code>ON</code> gesetzt werden.
OFF	Deaktivierung des XML-Supports. Alle XML-Subparameter-Einstellungen werden ignoriert.  Dies ist der Standardwert.
<i>keyword-subparameter</i>	Namen und Werte der Schlüsselwort-Subparameter siehe <a href="#">Schlüsselwort-Subparameter</a> .

## NTXML-Makro-Syntax

Das Makro NTXML wird wie folgt angegeben:

```

NTXML ON *
    PARSE=value, *
    RDCP=value, *
    RDIPV6=value, *
    RDNOP=value, *
    RDP=value, *
    RDOC=value, *
    RDPPORT=value, *
    RDPPOV6=value, *
    RDPS=value, *
    RDPSV6=value, *
    RDPV6=value, *
    RDSPORT=value, *
    RDSPOV6=value, *
    RDTOUT=value, *
    RDV4MAP=value

```

Oder:

```

NTXML OFF

```

Beschreibung der Syntax-Elemente siehe [XML-Parameter-Syntax](#).

## Schlüsselwort-Subparameter

PARSE | RDCP | RDIPV6 | RDNOP | RDOC | RDP | RDPPORT | RDPPOV6 | RDPS | RDPSV6 | RDPV6 | RDSPORT  
| RDSPOV6 | RDTOUT | RDV4MAP |



**Anmerkung:** Die Schlüsselwort-Subparameter RDPS, RDPSV6, RDSPORT und RDSPOV6, sind zurzeit nur für z/OS vorgesehen.

**PARSE - Unterstützung des PARSE XML-Statements**

`PARSE=value` aktiviert bzw. deaktiviert die Unterstützung des Statements `PARSE XML statement`.

Wert	Erklärung
ON	Die Verwendung des Statements <code>PARSE XML</code> wird unterstützt.
OFF	Die Verwendung des Statements <code>PARSE XML</code> wird nicht unterstützt. Dies ist der Standardwert.

**RDCP - Name der Standard-HTML-/XML-Codepage**

`RDCP=value` dient zur Angabe der Standard-Codepage, die verwendet wird, wenn `code-page-in` im `REQUEST DOCUMENT`-Statement nur Leerzeichen enthält.

Wert	Erklärung
<code>code-page-name</code>	Name der Standard-Codepage.
ISO 8859-1:1987	Dies ist der Standardwert.

**RDIPV6 - Unterstützung von IPv6 für das REQUEST DOCUMENT-Statement**

`RDIPV6=value` aktiviert bzw. deaktiviert die Unterstützung des Protokolls IPv6 beim `REQUEST DOCUMENT`-Statement.

Wert	Erklärung
ON	Die Verwendung des Protokolls IPv6 wird unterstützt, falls es auf dem lokalen Host zur Verfügung steht.
OFF	Die Verwendung des Protokolls IPv6 wird nicht unterstützt. Dies ist der Standardwert.

**RDNOP - Name der lokalen Domäne**

`RDNOP=value` dient zur Angabe der lokalen Domäne(n), die direkt, d.h. nicht über den Proxy-Server, angesprochen werden soll(en).

Wert	Erklärung
<code>domain-name(s)</code>	Name(n) der lokalen Domäne(n).  <b>Anmerkung:</b> 1. Leerzeichen sind nicht zulässig.

Wert	Erklärung
	<p>2. Stern-Notation für Präfixe kann nur in der Form <code>*.xxx</code> und nicht in der Form <code>.xxx</code> verwendet werden.</p> <p>3. Mehrere Namenseinträge müssen durch ein Semikolon (;) voneinander abgetrennt werden.</p>
OFF	<p>RDNOP=OFF bedeutet, dass keine URL definiert ist.</p> <p>Dies ist der Standardwert.</p>



**Anmerkung:** Die Angabe einer IPv6-Adresse oder von IPv6-Adresspräfixen ist möglich.

### RDOC - Unterstützung des REQUEST DOCUMENT-Statements

RDOC=*value* aktiviert bzw. deaktiviert die Unterstützung des Statements REQUEST DOCUMENT.

Wert	Erklärung
ON	Die Verwendung des Statements REQUEST DOCUMENT wird unterstützt.
OFF	Die Verwendung des Statements REQUEST DOCUMENT wird nicht unterstützt.
	Dies ist der Standardwert.

### RDP - URL des Proxy-Servers

RDP=*value* dient zur Angabe der URL des Proxy-Servers, über den alle Internet-(nicht Intranet-)HTTP-Anfragen geleitet werden müssen.

Wert	Erklärung
<i>url</i>	URL des Proxy-Servers. Leerzeichen sind nicht zulässig.
OFF	RDP=OFF bedeutet, dass keine URL definiert ist.
	Dies ist der Standardwert.

### RDPPORT - Proxy-Port-Nummer

RDPPORT=*value* dient zur Angabe der Proxy-Port-Nummer, falls gesetzt.

Value	Explanation
0 oder 1 - 65535	Port-Nummer.
80	Dies ist der Standardwert.

### RDPPOV6 - IPv6-Proxy-Port-Nummer

RDPPOV6=*value* dient zur Angabe der Port-Nummer des IPv6-Proxy-Servers, falls gesetzt.

Wert	Erklärung
0 oder 1 - 65535	Port-Nummer.
80	Dies ist der Standardwert.

### RDPS - URL des SSL-Proxy-Server

RDPS=*value* dient zur Angabe der URL des SSL-Proxy Servers, über den alle Internet-(nicht Intranet-)HTTP-Anfragen geleitet werden müssen.

Wert	Erklärung
<i>url</i>	URL des SSL-Proxy-Server. Leerzeichen sind nicht zulässig.
OFF	RDPS=OFF bedeutet, dass keine URL definiert ist. Dies ist der Standardwert.



**Anmerkung:** Dieser Schlüsselwort-Subparameter ist zurzeit nur für z/OS verfügbar.

### RDPSV6 - URL des IPv6-SSL-Proxy-Servers

RDPSV6=*value* dient zur Angabe der URL des IPv6-SSL-Proxy Servers, über den alle Internet-(nicht Intranet-)HTTP-Anfragen geleitet werden müssen.

Wert	Erklärung
<i>url</i>	URL des IPv6-SSL-Proxy-Servers. Leerzeichen sind nicht zulässig.
OFF	RDPSV6=OFF bedeutet, dass keine URL definiert ist. Dies ist der Standardwert.



**Anmerkungen:**

1. Dieser Schlüsselwort-Subparameter ist zurzeit nur für z/OS verfügbar.
2. Die Angabe einer nativen IPv6-Adresse in der URL-Notation ist möglich.

**RDPV6 - URL des IPv6-Proxy-Servers**

RDPV6=*value* dient zur Angabe der URL des IPv6-Proxy-Servers, über den alle Internet-(nicht Intranet-)HTTP-Anfragen geleitet werden müssen.

Wert	Erklärung
<i>url</i>	URL des IPv6-Proxy-Servers. Leerzeichen sind nicht zulässig.
OFF	RDPV6=OFF bedeutet, dass keine URL definiert ist. Dies ist der Standardwert.



**Anmerkung:** Die Angabe einer IPv6-Adresse ist möglich.

**RDSPORT – SSL Proxy Port-Nummer**

RDSPORT=*value* dient zur Angabe der Port-Nummer des SSL-Proxy-Servers, falls gesetzt.

Wert	Erklärung
0 - 65535	Port-Nummer des SSL Proxy Server.
443	Dies ist der Standardwert.



**Anmerkung:** Dieser Schlüsselwort-Subparameter ist zurzeit nur für z/OS verfügbar.

**RDSPOV6 – IPv6-SSL-Proxy-Port-Nummer**

RDSPOV6=*value* dient zur Angabe der Port-Nummer des IPv6-SSL-Proxy-Port-Servers, falls eine eingestellt ist.

Wert	Erklärung
0 oder 1 - 65535	Port-Nummer des IPv6-SSL-Proxy-Port-Servers.
443	Dies ist der Standardwert.



**Anmerkung:** Dieser Schlüsselwort-Subparameter ist zurzeit nur für z/OS verfügbar.

**RDTOUT - Zeitüberschreitungswert für laufende HTTP-Anfragen**

RDTOUT=*value* dient zur Angabe des Zeitüberschreitungswerts (in Sekunden) für in Ausführung befindliche HTTP-Anfragen.

Dieser Schlüsselwort-Subparameter wird nicht durch die nur für IPv4 vorgesehenen Lademodule NAT2TCP4 (z/OS) unterstützt.

Wert	Erklärung
0	Natural leistet keine Zeitüberschreitungskontrolle. Es gelten gelten die Standardeinstellungen. Dies ist der Standardwert.
1 - 32767	Zeit in Sekunden, nach der ein Zeitüberschreitungsfehler ausgegeben wird, wenn eine der folgenden Socket-Funktionen nicht innerhalb des angegebenen Zeitintervalls ausgeführt werden kann: Verbinden, Senden oder Empfangen.

**RDV4MAP - Unterstützung von in IPv4 abgebildeten Adressen unter IPv6**

RDV4MAP=*value* aktiviert bzw. deaktiviert die Verwendung von von in IPv4 abgebildeten IPv6-Adressen für URLs in symbolischer Notation. Numerische, in IPv4 abgebildete IPv6-Adressen werden im IPv6-Modus immer zugelassen.

Dieser Schlüsselwort-Subparameter wird nicht durch die nur für IPv4 vorgesehenen Lademodule NAT2TCP4 (z/OS) unterstützt.

Wert	Erklärung
ON	Aktiviert: In IPv4 abgebildete Adressen sind im IPv6-Protokoll zugelassen.
OFF	Deaktiviert: In IPv4 abgebildete Adressen sind im IPv6-Protokoll nicht zugelassen. Dies ist der Standardwert.

**Aus Sicherheitsgründen:**

Verwenden Sie in IPv4 abgebildete nur in IPv6-Umgebungen, in denen kein IPv4 TCP/IP Stack (oder Dual Stacking Mode) zur Verfügung steht, um auf IPv4-basierte HTTP-Server zuzugreifen. Weitere Informationen zu in IPv4 abgebildeten IPv6-Adressen siehe entsprechende IPv6-Dokumentation.

## Beispiel für XML Parameter

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

```
XML=(ON,RDP='HTTPPROXY.MYCOMPANY.COM',RDPPORT=8080,RDPS='SSLPROXY.MYCOMPANY.COM',RDSPOINT=443,RDNOP='*.MYCOMPANY.COM;2AEO:4899:200:1E00:',RDOC=ON,PARSE=ON,RDIPV6=ON,RDPV6='V6HTTPPROXY.MYCOMPANY.COM',RDPPOV6=888)
```



**Anmerkung:** Die Schlüsselwort-Subparameter RDSPOINT, RDSPOV6RDPSV6 und RDPS sind zurzeit nur für z/OS verfügbar.

## Beispiel für NTXML Makro

```
NTXML ON,RDP=HTTPPROXY.MYCOMPANY.COM,*
      RDPPORT=8080,*
      RDPS=SSLPROXY.MYCOMPANY.COM,*
      RDSPOINT=443,*
      RDNOP=*.MYCOMPANY.COM,*
      RDOC=ON,*
      PARSE=ON
```

```
NTXML ON,RDP=HTTPPROXY.MYCOMPANY.COM,*
      RDPPORT=8080,*
      RDPS=SSLPROXY.MYCOMPANY.COM,*
      RDSPOINT=443,*
      RDNOP=*.MYCOMPANY.COM,*
      RDOC=ON,*
      RDIPV6=ON,*
      RDPV6=V6HTTPPROXY.MYCOMPANY.COM,*
      RDPSV6=V6SSLPROXY.MYCOMPANY.COM,*
      RDSPOV6=8443
```



**Anmerkung:** Die Schlüsselwort-Subparameter RDSPOINT, RDSPOV6RDPSV6 und RDPS sind zurzeit nur für z/OS verfügbar.



# 281

## XREF - Erzeugung von XRef-Daten für Natural

---

- Möglichkeiten, den XREF-Parameter zu setzen ..... 884
- XRef-Datenerzeugung ..... 885
- Erweiterte XRef-Datenerzeugung (nur für interne Verwendung) ..... 885

Dieser Natural-Profilparameter dient zum Ein- und Ausschalten der Erzeugung von Verweistabellendaten (XRef Data) für Natural. Dieser Parameter bestimmt außerdem, wie die XRef-Daten behandelt werden, wenn Natural Member mit den Natural Utilities `SYSMAIN` oder `INPL` oder mit dem Object Handler verarbeitet werden.

<b>Mögliche Werte</b>	ON	XRef-Daten werden in den oben aufgeführten Fällen erzeugt. Die Dokumentationsvorbereitung wird nicht abgeprüft.	
	OFF	XRef-Daten werden nicht erzeugt. Die Dokumentationsvorbereitung wird nicht abgeprüft.	
	FORCE	Ein Natural-Objekt kann nur dann katalogisiert werden, wenn für dieses Implementierungsobjekt schon ein Dokumentationsobjekt existiert. XRef-Daten werden in den oben aufgeführten Fällen erzeugt.	
	DOC	Ein Natural-Objekt kann nur dann katalogisiert werden, wenn für dieses Implementierungsobjekt schon ein Dokumentationsobjekt existiert. XRef-Daten werden nicht erzeugt.	
<b>Standard-Einstellung</b>	OFF		
<b>Dynamische Angabe</b>	ja		
<b>Angabe innerhalb der Session</b>	ja	Gültige Statements:	keine
		Gültige Kommandos:	XREF

Folgende Themen werden nachfolgend behandelt:

## Möglichkeiten, den XREF-Parameter zu setzen

Es gibt mehrere Möglichkeiten, den Natural-Profilparameter `XREF` zu setzen:

- Im Natural-Parametermodul.
- Als dynamischen Parameter beim Starten einer Natural-Session.
- In Natural Security. Falls Natural Security verwendet wurde, um den `XREF`-Parameter zu setzen, kann das Kommando `XREF` nur verwendet werden, um diese Einstellung zu erzwingen (indem von `ON` auf `FORCE`, von `OFF` auf `ON` oder `FORCE` umgeschaltet wird).
- Mit dem Natural-Systemkommando `XREF`. Falls Natural Security nicht installiert ist, dann wird der `XREF`-Parameter üblicherweise mit dem Natural-Kommando `XREF` gesetzt. Das Kommando `XREF ?` zeigt die aktuelle Einstellung des `XREF`-Parameters an

## XRef-Datenerzeugung

---

XRef-Daten werden zwei Fällen erzeugt:

- Der Natural Compiler schreibt XRef-Daten für Natural-Programme und Natural Data Areas, wenn diese katalogisiert werden (vorausgesetzt, der Parameter XREF wurde entweder auf ON oder FORCE gesetzt, siehe unten).
- Natural Security schreibt XRef-Daten für Programme, die als Startup-, Restart- oder Error-Transaction-Programme in einer Anwendung verwendet werden oder als spezieller Link, wenn der Parameter XREF in der Natural Security-Definition der Anwendung auf ON oder FORCE gesetzt und eine Benutzersystemdatei für die Anwendung definiert ist.

Der XREF-Parameter steuert die Kompilierung in zweierlei Hinsicht:

- Erzeugung von XRef-Daten in den oben beschriebenen Fällen und
- Erfüllen der Vorbedingung, dass Implementierungsobjekte dokumentiert werden. Die Einhaltung dieser Vorbedingung kann dadurch sichergestellt werden, dass die Fertigstellung der Katalogisierungsoperation nur für Objekte erlaubt wird, die in der Predict-Systemdatei FDIC oder in der Development Server-Systemdatei in einer Natural Single Point of Development (SPoD)-Umgebung dokumentiert sind.

## Erweiterte XRef-Datenerzeugung (nur für interne Verwendung)

---



**Wichtig:** Der erweiterte XREF-Parameter ist für die interne Verwendung durch Natural vorgesehen.



# 282

## YD - Jahresdifferenz einstellen

Dieser Natural-Profilparameter kann verwendet werden, um das aktuelle Maschinendatum (gemäß Auslesung der internen Maschinenzeit) anzupassen, indem eine Anzahl von Jahren zu diesem Datum addiert oder davon subtrahiert wird. Das kann in Ländern von Nutzen sein, in denen verschiedene Kalender zum Einsatz kommen.

<b>Mögliche Werte</b>	-499 bis 499	Der Parameter wird angegeben als $YD=+nnn$ oder $YD=- nnn$  Dabei ist <i>nnn</i> die Anzahl der Jahre. Wenn der Profilparameter <b>MAXYEAR</b> auf 9999 gesetzt wird, ist die Obergrenze 7999.
<b>Standard-Einstellung</b>	0	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	



### Anmerkungen:

1. Falls das aktuelle Jahr ein Schaltjahr ist, das aus der YD-Einstellung resultierende aber nicht, dann wird anstelle des 29. Februars der 1. März verwendet.
2. Das Jahr, das sich aus der Summe der Einstellungen der Profilparameter TD, DD und YD ergibt, muss im Bereich von 1583 bis 2699 liegen. Wenn der Profilparameter MAXYEAR auf 9999 gesetzt ist, dann ist die obere Jahresgrenze 7999.



# 283

## YSLW - Gleitendes oder festes Jahr-Fenster

---

- Beispiele für den YSLW-Parameter ..... 891

Dieser Natural-Profilparameter dient zur Angabe eines Bereichs von Jahren, der durch das „gleitende Jahr-Fenster“ oder das „feste Jahr-Fenster“ abgedeckt wird.



**Anmerkung:** Bei diesen Mechanismen wird davon ausgegangen, dass ein Datum mit einem zweistelligen Jahr innerhalb eines „Fensters“ von 100 Jahren liegt. Innerhalb dieser 100 Jahre wird jede zweistellige Jahreseinstellung eindeutig zu einem spezifischen Jahrhundert zugeordnet, so dass es keine Verwirrung gibt, welches Jahrhundert gemeint ist.

<b>Mögliche Werte</b>	Normale Einstellung	0	Wenn Sie den Parameter auf 0 setzen, wird das aktuelle Jahrhundert angenommen. Es wird kein gleitendes oder festes Jahr-Fenster verwendet.
	Gleitendes Jahr-Fenster	1 - 99	Wenn Sie den Parameter auf einen Wert im Bereich zwischen 1 - 99 setzen, legen Sie fest, wann der 100-Jahr-Bereich in der Vergangenheit beginnt. Die YSLW-Einstellung wird vom aktuellen Jahr subtrahiert, um das erste Jahr des Fensterbereichs zu bestimmen.  Siehe <a href="#">Beispiel für ein gleitendes Jahr-Fenster</a> weiter unten.
	Festes Jahr-Fenster	1582 - 2600	Wenn Sie den Parameter auf einen Wert im Bereich zwischen 1582 - 2600 setzen, legen Sie das erste Jahr eines 100-Jahre-Bereichs fest. Die Obergrenze des 100-Jahre-Bereichs wird berechnet, indem 99 zum angegebenen Wert addiert wird.  Siehe <a href="#">Beispiel für ein festes Jahr-Fenster</a> weiter unten.
<b>Standard-Einstellung</b>	0		Es wird kein gleitendes oder festes Jahr-Fenster verwendet.
<b>Dynamische Angabe</b>	ja		
<b>Angabe innerhalb der Session</b>	nein		

Ausgewertet wird der Parameter YSLW zur Laufzeit, wenn eine alphanumerische Datumseinstellung mit einer zweistelligen Jahreskomponente in eine Datumsvariable verschoben wird. Dies gilt für Datumseinstellungen, die

- in der mathematischen Funktion VAL verwendet werden,
- zusammen mit der Option IS(D) in einer logischen Bedingung verwendet werden,
- vom Natural-Stack als Eingabedaten gelesen werden,
- oder als Eingabedaten in eine Maske (Eingabemaske (Input Map)) eingegeben werden.

Siehe auch *Verarbeitung von Datumsinformationen im Leitfaden zur Programmierung*.

## Beispiele für den YSLW-Parameter

---

### Beispiel für ein gleitendes Jahr-Fenster

Wenn das aktuelle Jahr 2014 ist und wenn Sie  $YSLW=40$  angeben, dann deckt das gleitende Fenster die Jahre 1974 bis 2073 ab. Eine zweistellige Jahreseinstellung  $nn$  von 74 bis 99 wird dann entsprechend als  $19nn$  interpretiert, während eine zweistellige Jahreseinstellung  $nn$  von 00 bis 73 als  $20nn$  interpretiert wird.

Siehe auch Beispiele unter *Gleitendes Jahr-Fenster — der YSLW-Parameter* und *Kombinationen von DFSTACK und YSLW* im Leitfaden zur Programmierung.

### Beispiel für ein festes Jahr-Fenster

Wenn Sie  $YSLW=1985$  angeben, dann deckt das feste Fenster die Jahre 1985 bis 2084 ab. Eine zweistellige Jahreseinstellung  $nn$  von 85 bis 99 wird dann entsprechend als  $19nn$  interpretiert, während eine zweistellige Jahreseinstellung  $nn$  von 00 bis 84 als  $20nn$  interpretiert wird.



# 284 ZD - Division durch Null

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

<b>Mögliche Werte</b>	ON	Natural gibt eine Fehlermeldung aus, falls versucht wird, eine Zahl durch Null (0) zu dividieren.
	OFF	Natural gibt bei einer Division durch Null als Ergebnis eine Null aus.
<b>Standard-Einstellung</b>	ON	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	ja	
<b>Gültige Statements</b>	SET GLOBALS	
<b>Gültiges Kommando</b>	GLOBALS	
<b>Programmierschnittstelle (API)</b>	USR1005N	Siehe <i>SYSEXT Utility</i> in der Dokumentation <i>Debugger und Dienstprogramme</i> .



## Anmerkungen:

1. In einer Natural-Session kann der Profilparameter ZD durch den Session-Parameter ZD überschrieben werden.
2. Unter Natural Security kann die Einstellung dieses Parameters durch die *Session Parameters*-Option des Library-Profiles überschrieben werden.



# 285

## ZIIP - zIIP-Verarbeitung (nur z/OS)

---

▪ ZIIP-Parameter-Syntax .....	896
▪ NTZIIP-Makro-Syntax .....	897
▪ Schlüsselwort-Subparameter .....	897
▪ Beispiel für ZIIP-Parameter .....	903
▪ Beispiel für NTZIIP-Macro .....	903

Dieser Natural-Profilparameter dient zum Konfigurieren der zIIP-Verarbeitung unter Natural. Der Profilparameter ZIIP entspricht dem Makro [NTZIIP](#) im Natural-Parametermodul.

 **Anmerkung:** Die Einstellung des Profilparameters ZIIP wird nur dann wirksam, wenn Natural Batch for zIIP, Natural for CICS for zIIP, Natural for Com-plete for zIIP oder Natural for IMS for zIIP in Ihrer z/OS-Umgebung installiert worden ist. Weitere Informationen siehe *Installing Natural for zIIP* in der *Installation for z/OS*-Dokumentation.

<b>Mögliche Werte</b>	Siehe <a href="#">ZIIP-Parameter-Syntax</a> .	
<b>Standard-Einstellung</b>	AUTO	Siehe <a href="#">ZIIP-Parameter-Syntax</a> und <a href="#">Schlüsselwort-Subparameter</a> .
<b>Dynamische Angabe</b>	ja	Der Profilparameter ZIIP kann nur dynamisch angegeben werden. Im Natural-Parametermodul wird das Makro <a href="#">NTZIIP</a> verwendet.
<b>Angabe innerhalb der Session</b>	ja	Mit dem Systemkommando ZIIP kann die Erfassung von Statistikinformationen über das Umschalten von Komponenten gesteuert und die Anzeige von Informationen über die Natural-zIIP-Verarbeitung angezeigt werden.

Dieses Dokument behandelt folgende Themen:

## ZIIP-Parameter-Syntax

Der Parameter ZIIP wird wie folgt angegeben:

```
ZIIP=(state,keyword-subparameter=value,keyword-subparameter=value,...)
```

Oder:

```
ZIIP=state
```

Dabei kann *state* den Wert ON, OFF oder AUTO haben.

Wert	Erklärung
ON	zIIP-Unterstützung aktiviert.
OFF	zIIP-Unterstützung nicht aktiviert.
AUTO	Die zIIP-Unterstützung wird nur dann aktiviert, wenn mindestens ein zIIP online ist oder wenn der z/OS-Parameter PROJECTCPU=YES im SYS1.PARMLIB-Member IEAOPT gesetzt ist.  Dies ist der Standardwert.
<i>keyword-subparameter</i>	Siehe <a href="#">Schlüsselwort-Subparameter</a> .



**Anmerkung:** Wenn einer der Werte ON, OFF oder AUTO in Klammern innerhalb einer Liste angegeben wird, muss er an erster Stelle stehen.

## NTZIIP-Makro-Syntax

Das Makro NTZIIP wird wie folgt angegeben:

```

NTZIIP state, *
      IMSG=value, *
      IOCSIZE=value, *
      PNR=value, *
      PRINT=value, *
      PWCSIZE=value, *
      PWCXAM=value, *
      STAT=value, *
      TMSG=value
    
```

Siehe [Schlüsselwort-Subparameter](#).

Beschreibung von *state* siehe [ZIIP-Parameter-Syntax](#).

## Schlüsselwort-Subparameter

[IMSG](#) | [IOCSIZE](#) | [PNR](#) | [PRINT](#) | [PWCSIZE](#) | [PWCXAM](#) | [STAT](#) | [TMSG](#)

### IMSG – Natural zIIP Support-Meldung

Dieser Schlüsselwort-Subparameter bestimmt, ob Natural eine Systemmeldung NAT7070 ausgibt, nachdem bei der Initialisierung der Session der Natural zIIP Support erfolgreich ermöglicht worden ist.

Wert	Erklärung
ON	Die zIIP Support-Meldung wird ausgegeben.  Dies ist der Standardwert.
OFF	Die zIIP Support-Meldung wird nicht ausgegeben.

## IOCSIZE - Cache-SpeichergroÙe für primäre Batch-Ein-/Ausgabedateien

Dieser Schlüsselwort-Subparameter kann benutzt werden, um die Cache-SpeichergroÙen für die Verarbeitung der primären Batch-Ein-/Ausgabedateien CMSYNIN, CMOBJIN und CMPRINT festzulegen. Dies kann von Nutzen sein, wenn Natural im zIIP-Modus läuft, weil so Mehraufwand verringert wird, der durch das Umschalten vom zIIP zum GCP (General Central Processor) und umgekehrt verursacht wird. Es gibt je einen Cache-Speicher für die CMSYNIN-Eingabedatei, die CMOBJIN-Eingabedatei und die CMPRINT-Ausgabedatei.

Der Vorteil eines Cache-Speichers liegt darin, dass nicht unmittelbar auf die Datei zugegriffen wird, sondern alle Ein- und Ausgabedaten zunächst in dem Cache-Speicher gesammelt werden. Wenn der Cache-Speicher für die CMPRINT-Datei voll ist, wird er geleert und alle Datensätze werden auf einmal geschrieben.

Damit die Daten nicht zu lange innerhalb des CMPRINT-Cache-Puffers verborgen bleiben, löst ein Zeitscheiben-Mechanismus eine automatische Bereinigung aus, wenn die folgenden Bedingungen beide erfüllt sind:

- wenn ein neuer Datensatz in den (nicht leeren) CMPRINT-Cache-Puffer eingegeben wird und
- wenn der erste Datensatz im CMPRINT-Cache-Puffer vor mehr als 60 Sekunden geschrieben wurde.

Bei einer CMSYNIN- bzw. CMOBJIN-Eingabedatei werden alle Datensätze für die gegebene Datei gelesen und in dem Eingabe-Cache-Speicher für diese Datei gespeichert. Anschließend liest Natural die Eingabedatensätze direkt aus dem Eingabe-Cache-Speicher.

 **Vorsicht:** Die Verwendung des Schlüsselwort-Subparameters IOCSIZE kann das Erscheinen der an CMPRINT gesendeten Ausgabe verzögern.

### IOCSIZE-Parameter-Syntax

Der Subparameter IOCSIZE wird wie folgt angegeben:

IOCSIZE=(CMSYNIN-cache-size, CMOBJIN-cache-size, CMPRINT-cache-size)

Dabei ist:

Syntax-Element	Wert	Erklärung
CMSYNIN-cache-size	1 - 2097151	CMSYNIN-cache-size in KB dient zur Festlegung der Größe des Cache-Speichers für die Eingabedatei CMSYNIN.
	0	Für die Eingabedatei CMSYNIN wird kein Zwischenspeicher zugeteilt. Dies ist der Standardwert.
CMOBJIN-cache-size	1 - 2097151	CMOBJIN-cache-size in KB dient zur Festlegung der Größe des Cache-Speichers für die Eingabedatei CMOBJIN.

Syntax-Element	Wert	Erklärung
	0	Für die Eingabedatei CMOBJIN wird kein Zwischenspeicher zugeteilt. Dies ist der Standardwert.
<i>CMPRINT-cache-size</i>	64 - 2097151	<i>CMPRINT-cache-size</i> in KB dient zur Festlegung der Größe des Cache-Speichers für die Ausgabedatei CMPRINT.
	0	Für die Ausgabedatei CMPRINT wird kein Zwischenspeicher zugeteilt. Dies ist der Standardwert.



**Anmerkungen:**

1. Die Standardeinstellung ist  $IOCSIZE=(0,0,0)$ . Wenn Sie einen Standardwert ändern möchten, brauchen Sie nur die zu ändernde Größe anzugeben, zum Beispiel  $IOCSIZE=(,200)$ , um 200 KB nur für den CMOBJIN-Cache-Speicher zuzuteilen.
2. Während der Session können Sie den Subparameter *IOCSIZE* nicht ändern.

**PNR – Druckdatei für zIIP-Verarbeitungsinformationen (nur in Batch Sessions)**

Mit diesem Schlüsselwort-Subparameter können Sie die Datei festlegen, in die die Informationen über die zIIP-Verarbeitung (siehe Schlüsselwort-Subparameter [PRINT](#)) am Ende einer Batch Session gedruckt wird. Bei Server- und in TSO-Sessions wird der Subparameter *PNR* ignoriert.

Wert	Erklärung
0	Die Informationen über die zIIP-Verarbeitung werden an den Standard-Drucker geleitet (CMPRINT).  Dies ist der Standardwert.
1 - 31	Die Informationen über die zIIP-Verarbeitung werden in die angegebene Druckdatei geleitet. Wenn beim Beenden der Session die angegebene Druckdatei nicht zur Verfügung steht, gibt Natural eine Fehlermeldung zurück.  Weitere Informationen zur Definition von Natural-Druckdateien siehe Profilparameter <i>PRINT</i>

**PRINT – zIIP-Verarbeitungsinformationen drucken (nur in Batch Sessions)**

Dieser Schlüsselwort-Subparameter kann benutzt werden, um am Ende der Batch Session automatisch einen Ausdruck mit Informationen über die zIIP-Verarbeitung zu erhalten. Bei Server- und in TSO-Sessions wird dieser Subparameter ignoriert.

Wert	Erklärung
INFO	Nur allgemeine zIIP-Verarbeitungsinformationen drucken.
STAT	Nur zIIP-Komponenten-Umschaltstatistikinformationen drucken.
ALL	Sowohl allgemeine zIIP-Verarbeitungsinformationen als auch zIIP-Komponenten-Umschaltstatistikinformationen drucken.
OFF	Am Ende der Session wird kein Ausdruck der zIIP-Verarbeitungsinformationen erzeugt. Dies ist der Standardwert.

### PWCSIZE - Cache-Speichergroße für Druck- und Arbeitsdateien

Dieser Schlüsselwort-Subparameter kann benutzt werden, um die Größe von Cache-Speichern für die Ein-/Ausgabeverarbeitung bei Druck- und Arbeitsdateien festzulegen. Dies ist von Nutzen, wenn Natural im zIIP-Modus läuft, weil so Mehraufwand verringert wird, der durch das Hin- und Herschalten zwischen zIIP- und GCP-Prozessor (General Central Processor) entsteht. Es gibt einen Cache-Speicher für Druckdateien, einen für Ausgabe-Arbeitsdateien und einen für jede Eingabe-Arbeitsdatei.

Der Vorteil eines Cache-Speichers liegt darin, dass nicht unmittelbar auf die Datei zugegriffen wird, sondern alle Ein-/Ausgabedaten zunächst im Cache-Speicher gesammelt werden. Wenn der Cache-Speicher für Druckdateien oder Ausgabe-Arbeitsdateien voll ist, wird der Cache-Speicher geleert, d.h., alle Datensätze werden auf einmal in die betreffenden Dateien geschrieben. Bei der Arbeitsdatei-Eingabe werden alle Datensätze für eine gegebene Datei gelesen und im Eingabe-Cache-Speicher für diese Datei gespeichert. Anschließend liest Natural die Eingabedatensätze direkt aus dem Eingabe-Cache-Speicher.

 **Vorsicht:** Die Verwendung des Schlüsselwort-Subparameter `PWCSIZE` kann sich negativ auf den Ablauf Ihrer Anwendung auswirken. Beispielsweise kann die Anwendung gesendete Datensätze erst zu einem späteren Zeitpunkt erhalten, da ja die Datensätze nicht sofort auf den gewünschten Datenträger geschrieben werden. Darüber hinaus kann das Verhalten im Fehlerfall unterschiedlich sein. Beispiel: Falls der Fehler NAT1507 (Arbeits-/Druckdatei voll) auftritt, wird dieser nicht sofort angezeigt. Stattdessen wird später beim Entleeren des Cache-Speichers die Meldung NAT1532 (Fehler beim Leeren des Arbeits-/Druckdatei-Cache-Speichers) angezeigt. Den tatsächlich aufgetretenen Fehler können Sie sich dann nur mit Hilfe des Systemkommandos `LASTMSG` anzeigen lassen. Das bedeutet, dass eine `ON ERROR`-Klausel beim Fehler NAT1507 nicht mehr funktioniert.

### PWCSIZE-Parameter-Syntax

Der Subparameter `PWCSIZE` wird wie folgt angegeben:

PWCSIZE=(*print-size,work-input-size,work-output-size*)

Dabei ist:

Syntax-Element	Wert	Erklärung
<i>print-cache-size</i>	1 - 2097151	<i>print-cache-size</i> in KB dient zur Festlegung der Größe des Cache-Speichers für alle Druckdateien.
	0	Es wird kein Cache-Speicher für Druckdateien zugeteilt. Dies ist der Standardwert.
<i>work-input-cache-size</i>	1 - 2097151	<i>work-input-cache-size</i> in KB dient zur Festlegung der Größe eines Cache-Speichers für jede Eingabe-Arbeitsdatei.
	0	Es wird kein Cache-Speicher für jede Eingabe-Arbeitsdateien zugeteilt. Dies ist der Standardwert.
<i>work-output-cache-size</i>	1 - 2097151	<i>work-output-cache-size</i> in KB dient zur Festlegung der Größe des Cache-Speichers für alle Ausgabe-Arbeitsdateien.
	0	Es wird kein Cache-Speicher für alle Ausgabe-Arbeitsdateien zugeteilt. Dies ist der Standardwert.



#### Anmerkungen:

1. Die Standardeinstellung ist PWCSIZE=(0,0,0). Wenn Sie einen Standardwert ändern möchten, brauchen Sie nur die zu ändernde Größe anzugeben, zum Beispiel PWCSIZE=(,200), um nur dem Eingabe-Pufferspeicher einen Wert von 200 KB zuzuweisen.
2. Während der Session können Sie den Subparameter PWCSIZE nicht ändern.

### PWCXAM - Zwischenspeicherung von Druck- und Arbeitsdateien deaktivieren

Dieser Schlüsselwort-Subparameter kann benutzt werden, um die Zwischenspeicherung („Caching“) von Druck- und Arbeitsdateien (definiert mit dem Schlüsselwort-Subparameter [PWCSIZE](#)) bei einzelnen oder mehreren, für die Dateien definierten Zugriffsmethoden zu deaktivieren.

Die Deaktivierung bei Zugriffsmethoden kann nützlich sein, wenn Sie eine negative Auswirkung auf den logischen Ablauf Ihrer Anwendung vermeiden möchten (siehe auch die [Warnung](#) beim Subparameter PWCSIZE).

#### PWCXAM-Parameter-Syntax

Der Subparameter PWCXAM wird wie folgt angegeben:

Bei einzelner Zugriffsmethode:

`PWCXAM=access-method`

Bei mehreren Zugriffsmethoden:

`PWCXAM=(access-method1,access-method2,...)`

Die obige Syntax gilt für das NTZIIP-Macro im Natural-Parametermodul.

Wenn Sie mehr als einen Wert dynamisch angeben wollen, müssen Sie PWCXAM mehrmals angeben.

*access-method* definiert den Typ der zu benutzenden Zugriffsmethode. Die Werte, die Sie angeben können, entsprechen den Werten des Subparameters **AM** des Profilparameters **PRINT** bzw. des Subparameters **AM** des Profilparameters **WORK**.

### STAT – zIIP-Komponenten-Umschaltstatistikinformationen

Dieser Schlüsselwort-Subparameter steuert die Erfassung von Statistikinformationen über das Umschalten von Komponenten in den TCB-Modus. Diese Statistikinformationen können während der Session durch Absetzen des Systemkommandos **ZIIP** angezeigt oder am Ende der Session (nur im Batch-Betrieb) durch den Subparameter **PRINT** automatisch ausgelöst werden. Darüber hinaus können die Statistikinformationen mit dem Systemkommando **ZIIP** gesteuert werden.

Wert	Erklärung
ON	Erfassung von Statistikinformationen über das Umschalten von Komponenten aktivieren.
OFF	Erfassung von Statistikinformationen über das Umschalten von Komponenten deaktivieren. Dies ist der Standardwert.



**Anmerkung:** Die Einstellung des Schlüsselwort-Subparameters **STAT** kann durch die Einstellung des Option **STAT** im Systemkommando **ZIIP** und über die Anwendungsprogrammierschnittstelle **USR8204N**, die **ZIIP**-Kommandofunktionen ausführt, außer Kraft gesetzt werden. Siehe entsprechende Abschnitte in der *Systemkommandos*-Dokumentation.

### TMSG – Bestimmungsort der Natural zIIP Support-Initialisierungsmeldung

Dieser Schlüsselwort-Subparameter bestimmt, ob Natural die zIIP Support-Initialisierungsmeldungen an das Job-Protokoll oder an das Natural-Standard-Ausgabegerät, d.h. Benutzer-Terminal bei Online-Umgebungen oder **CMPRINT** bei Batch-Betrieb, sendet. Dies kann für Testzwecke hilfreich sein.

Wert	Erklärung
ON	Alle zIIP Support-Initialisierungsmeldungen werden an das Natural-Standard-Ausgabegerät gesendet.
OFF	Alle zIIP Support-Initialisierungsmeldungen werden an das Job-Protokoll gesendet. Dies ist der Standardwert.

## Beispiel für ZIIP-Parameter

```
ZIIP=(ON,STAT=ON,PWCSIZE=(200,,300),IOCSIZE=(100,100,500))
```

## Beispiel für NTZIIP-Macro

```
NTZIIP AUTO, *
      STAT=ON, *
      PWCSIZE=(500,)*
      IOCSIZE=(,2000), *
      PRINT=ALL
```



# 286

## ZP - Anzeige von Nullwerten

Dieser Natural Profil- und Session-Parameter gibt an, wie ein Feld ausgegeben werden soll, das nur Nullen enthält.

<b>Mögliche Werte</b>	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.
<b>Standard-Einstellung</b>	ON	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	ja	
<b>Gültige Statements</b>	DISPLAY FORMAT INPUT PRINT REINPUT SET GLOBALS WRITE	
<b>Gültiges Kommando</b>	GLOBALS	
<b>Programmierschnittstelle (API)</b>	USR1005N	Siehe <i>SYSEXT Utility</i> in der Dokumentation <i>Debugger und Dienstprogramme</i> .



### Anmerkungen:

1. Dieser Natural Profil- und Session-Parameter 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.
2. In einer Natural-Session kann der Profilparameter ZP durch den Session-Parameter ZP überschrieben werden.
3. Siehe auch *Parameter zur Beeinflussung der Ausgabe von Feldern im Leitfaden zur Programmierung*.



# 287

## ZSIZE - Größe des Entire DB-Pufferbereichs

---

Dieser Natural-Profilparameter gilt nur bei Entire DB. Er gibt die Größe des für Entire DB benötigten Pufferbereichs.

<b>Mögliche Werte</b>	1 - 64	Größe des Pufferbereichs in KB.
	0	Wenn ZSIZE=0 gesetzt ist oder wenn der erforderliche Speicherplatz nicht zur Verfügung steht, kann das Entire DB Interface nicht benutzt werden.
<b>Standard-Einstellung</b>	0	
<b>Dynamische Angabe</b>	ja	
<b>Angabe innerhalb der Session</b>	nein	



### Anmerkungen:

1. Dieser Natural-Profilparameter gilt nur bei Entire DB.
2. Alternativ können Sie den Natural-Profilparameter **DS** oder das Makro **NTDS** benutzen,.



## Stichwortverzeichnis

---

### A

ABLOG  
parameter, 27  
AD  
parameter, 29  
ADAACBX  
parameter, 37  
ADANAME  
parameter, 43  
ADAPRM  
parameter, 45  
ADASBV  
parameter, 47  
AL  
parameter, 49  
ASIZE  
parameter, 51  
ASPSIZE  
parameter, 53  
ATTN  
parameter, 55  
AUTO  
parameter, 57

### B

BPC64  
parameter, 61  
BPCSIZE  
parameter, 63  
BPI  
Parameter, 65  
BPLIST  
parameter, 75  
BPMETH  
parameter, 77  
BPNAME  
parameter, 79  
BPPROP  
parameter, 81  
BPSFI  
parameter, 83  
BPSIZE  
parameter, 85  
BPTEXT  
parameter, 87  
BSIZE  
parameter, 89

BX  
parameter, 91

### C

CANCEL  
parameter, 93  
CC  
parameter, 95  
CCTAB  
parameter, 99  
CD  
parameter, 105  
CDYNAM  
parameter, 107  
CF  
parameter, 109  
CFICU  
Parameter, 111  
CFWSIZE  
parameter, 119  
CICSP  
parameter, 121  
CLEAR  
parameter, 143  
CM  
parameter, 145  
CMPO  
parameter, 147  
CP  
Parameter, 161  
CPCVERR  
parameter, 165  
CPOBJIN  
Parameter, 167  
CPPRINT  
Parameter, 169  
CPSYNIN  
Parameter, 171  
CSIZE  
parameter, 173  
CSTATIC  
parameter, 175  
CV  
parameter, 179  
CVMIN  
parameter, 181

## D

DATSIZE  
 Parameter, 183  
 DB  
 Parameter, 187  
 DB2  
 Parameter, 193  
 DB2SIZE  
 Parameter, 209  
 DBCLOSE  
 Parameter, 211  
 DBGAT  
 Parameter, 213  
 DBGERR  
 Parameter, 217  
 DBID  
 Parameter, 219  
 DBOPEN  
 Parameter, 221  
 DBROLL  
 Parameter, 223  
 DBUPD  
 Parameter, 225  
 DC  
 Parameter, 227  
 DD  
 Parameter, 229  
 DELETE  
 Parameter, 231  
 DF  
 Parameter, 233  
 DFOUT  
 Parameter, 235  
 DFSTACK  
 Parameter, 237  
 DFTITLE  
 Parameter, 239  
 DL  
 Parameter, 241  
 DLISIZE  
 Parameter, 243  
 DO  
 Parameter, 245  
 DS  
 Parameter, 247  
 DSC  
 Parameter, 253  
 DSIZE  
 Parameter, 255  
 DTFORM  
 Parameter, 257  
 DU  
 Parameter, 259  
 DUE  
 Parameter, 261  
 DY  
 Parameter, 263  
 DYNPARM  
 Parameter, 269

## E

ECHO

Parameter, 273  
 EDBP  
 Parameter, 275  
 EDPSIZE  
 Parameter, 285  
 EJ  
 Parameter, 287  
 EM  
 Parameter, 289  
 EMFM  
 Parameter, 305  
 EMU  
 Parameter, 307  
 ENDBT  
 Parameter, 309  
 ENDMSG  
 Parameter, 311  
 ES  
 Parameter, 313  
 ESCAPE  
 Parameter, 315  
 ESIZE  
 Parameter, 317  
 ET  
 Parameter, 319  
 ETA  
 Parameter, 321  
 ETDB  
 Parameter, 323  
 ETEOP  
 Parameter, 325  
 ETID  
 Parameter, 327  
 ETIO  
 Parameter, 329  
 ETRACE  
 Parameter, 331  
 ETSYNC  
 Parameter, 333  
 EXCSIZE  
 Parameter, 335  
 EXRSIZE  
 Parameter, 337

## F

FAMSTD  
 Parameter, 339  
 FC  
 Parameter, 341, 343  
 FCDP  
 Parameter, 345  
 FDIC  
 Parameter, 347  
 FL  
 Parameter, 349  
 FNAT  
 Parameter, 351  
 FNR  
 Parameter, 355  
 FREGDA  
 Parameter, 361  
 FREG  
 Parameter, 363

FS  
 Parameter, 367  
 FSEC  
 Parameter, 369  
 FSPOOL  
 Parameter, 375  
 FUSER  
 Parameter, 377

**G**

GC  
 parameter, 381

**H**

HC  
 parameter, 383  
 HCAM  
 parameter, 385  
 HCDEST  
 parameter, 387  
 HD  
 parameter, 389  
 HE  
 parameter, 391  
 HI  
 parameter, 397  
 HW  
 parameter, 399

**I**

IA  
 parameter, 401  
 IC  
 Parameter, 405  
 ICU  
 Parameter, 407  
 ID  
 parameter, 409  
 IKEY  
 parameter, 411  
 IM  
 parameter, 413  
 IMSG  
 parameter, 415  
 IMSP  
 parameter, 417  
 IMSPE  
 parameter, 421  
 IMSPT  
 parameter, 435  
 INTENS  
 parameter, 441  
 IP  
 parameter, 443  
 IS  
 parameter, 445  
 ISIZE  
 parameter, 447  
 ITERM  
 parameter, 449  
 ITRACE

parameter, 451

**K**

KD  
 parameter, 453  
 KEY  
 parameter, 455

**L**

LC  
 parameter, 457, 459  
 LCU  
 parameter, 461  
 LE  
 parameter, 463  
 LFILE  
 parameter, 465  
 LIBNAM  
 parameter, 469  
 LOG  
 parameter, 471  
 LS  
 parameter, 473  
 LT  
 parameter, 477

**M**

MADIO  
 parameter, 479  
 MAINPR  
 parameter, 481  
 MAXCL  
 parameter, 483  
 MAXROLL  
 parameter, 485  
 MAXYEAR  
 parameter, 487  
 MC  
 parameter, 489  
 MENU  
 parameter, 491  
 ML  
 parameter, 493  
 MONSIZE  
 parameter, 495  
 MP  
 parameter, 497  
 MS  
 parameter, 499  
 MSGSF  
 parameter, 501  
 MT  
 parameter, 503

**N**

NAFSIZE  
 parameter, 505  
 NAFUPF  
 parameter, 507  
 NC

parameter, 509  
 NISN  
 parameter, 511  
 NL  
 parameter, 513  
 NTCICSP  
 macro, 121  
 NTIMSP  
 macro, 417  
 NTIMSPE  
 macro, 421  
 NTIMSPT  
 macro, 435  
 NUCNAME  
 parameter, 515

## O

OBJIN  
 parameter, 517  
 OPF  
 parameter, 519  
 OPRB  
 Parameter, 521  
 OPT  
 parameter, 527  
 OSP  
 parameter, 529  
 OUTDEST  
 parameter, 535  
 OVSIZE  
 parameter, 537

## P

parameter  
 overview, xvii  
 PARM  
 parameter, 539  
 PC  
 parameter, 541, 543  
 PCNTRL  
 parameter, 545  
 PD  
 parameter, 547  
 PDPSIZE  
 parameter, 549  
 PECK  
 parameter, 551  
 PGP  
 parameter, 553  
 PLOG  
 parameter, 559  
 PM  
 parameter, 561  
 POS22  
 parameter, 565  
 PRINT  
 parameter, 567  
 PROFILE  
 parameter, 587  
 profile parameter  
 introduction, 5  
 overview, xvii

PROGRAM  
 parameter, 593  
 PS  
 parameter, 597  
 PSEUDO  
 parameter, 599

## R

RCA  
 parameter, 601  
 RCALIAS  
 parameter, 603  
 RCFIND  
 parameter, 607  
 RCGET  
 parameter, 609  
 RDACT  
 parameter, 611  
 RDC  
 parameter, 613  
 RDCEXIT  
 parameter, 619  
 RDCSIZE  
 parameter, 621  
 RDNODE  
 parameter, 623  
 RDPORT  
 parameter, 625  
 RECAT  
 parameter, 627  
 REINP  
 parameter, 629  
 RELO  
 parameter, 631  
 RFILE  
 parameter, 633  
 RI  
 parameter, 635  
 RJESIZE  
 parameter, 637  
 RM  
 parameter, 639  
 ROSY  
 parameter, 643  
 RPC  
 parameter, 645  
 RUNSIZE  
 parameter, 667

## S

SA  
 parameter, 669  
 SB  
 parameter, 671  
 SCTAB  
 parameter, 677  
 SELUNIT  
 parameter, 681  
 SENDER  
 parameter, 685  
 session parameter  
 introduction, 7

overview, xvii  
 SF  
 parameter, 687  
 SG  
 parameter, 689  
 SHAPED  
 parameter, 691  
 SKEY  
 parameter, 693  
 SL  
 parameter, 695  
 SLOCK  
 parameter, 697  
 SM  
 parameter, 699  
 SO  
 parameter, 709  
 SORT  
 parameter, 701  
 SRETAIN  
 Parameter, 713  
 SSIZE  
 parameter, 715  
 STACK  
 parameter, 717  
 STACKD  
 parameter, 719  
 STEPLIB  
 parameter, 721  
 SUBSID  
 parameter, 723  
 SYNERR  
 parameter, 725  
 SYS  
 parameter, 727  
 SYSCIP  
 parameter, 731  
 SYSPSW  
 parameter, 733

**T**

TAB  
 parameter, 735  
 TAB1  
 parameter, 739  
 TAB2 - Alternative Input Translation  
 parameter, 743  
 TABA1  
 parameter, 747  
 TABA2  
 parameter, 751  
 TABL  
 parameter, 755  
 TC  
 Parameter, 759  
 TCU  
 Parameter, 761  
 TD  
 parameter, 763  
 TF  
 parameter, 765  
 THPINIT  
 parameter, 769

THSEPCH  
 parameter, 771  
 THSIZE  
 parameter, 773  
 TMODEL  
 parameter, 775  
 TPF  
 parameter, 777  
 TQ  
 parameter, 779  
 TRACE  
 parameter, 781  
 TS  
 parameter, 785  
 TSIZE  
 parameter, 787  
 TSOP  
 parameter, 789  
 TTYPE  
 parameter, 797

**U**

UC  
 parameter, 799  
 UCONMAX  
 parameter, 801  
 UDB  
 parameter, 803  
 ULANG  
 parameter, 805  
 USER  
 parameter, 809  
 USERBUF  
 parameter, 813  
 UTAB1  
 parameter, 815  
 UTAB2  
 parameter, 819

**V**

VSAM  
 parameter, 823  
 VSIZE  
 parameter, 843

**W**

WEBIO  
 parameter, 845  
 WH  
 parameter, 849  
 WORK  
 parameter, 851  
 WPSIZE  
 parameter, 869  
 WSISIZE  
 parameter, 871

**X**

XML  
 parameter, 873

XREF  
parameter, 883

## Y

YD  
parameter, 887  
YSLW  
parameter, 889

## Z

ZD  
parameter, 893  
ZP  
parameter, 905  
ZSIZE  
parameter, 907