Natural for z/OS

TP-Monitor-Schnittstellen

Version 9.2.5

October 2026

ADABAS & NATURAL

This document applies to Natural for z/OS Version 9.2.5 and all subsequent releases.

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

Copyright © 1979-2026 Software GmbH, Darmstadt, Germany and/or its subsidiaries and/or its affiliates and/or their licensors.

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

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

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

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

Document ID: ATMF-TPMON-925-20261013DE

Table of Contents

Preface ... ix
1 About this Documentation ... 1
Dokumentationskonventionen ... 2
Online-Informationen und Support ... 2
Datenschutz ... 3
I Verwendung von Natural mit TP-Monitoren ... 5
2 Verwendung von Natural mit TP-Monitoren ... 7
Von Natural unterstützte TP-Monitor-Systeme ... 8
Einsatz von Natural in einer Datenfernverarbeitungsumgebung (Teleprocessing, TP) ... 8
II Natural unter CICS ... 11
3 Unterstützung für zIIP unter CICS ... 13
4 Funktionsweise der Natural CICS-Schnittstelle ... 15
Natural CICS-Schnittstelle ... 16
Umgebungsabhängiger Natural Nukleus für CICS ... 16
Systemsteuerung unter CICS ... 17
Natural-System-Komponenten im dynamischen CICS-Speicher ... 18
Natural-Speicher-Threads unter CICS ... 21
Natural Roll Facilities unter CICS ... 22
CICS Roll Facilities ... 22
Lokaler Natural Buffer Pool unter CICS ... 23
Systemsteuerungssätze der Natural CICS-Schnittstelle im CICS-Zwischenspeicher ... 23
NCIDIREX - Systemverzeichnismodulnamen-Exit-Schnittstelle ... 24
NCIDTPEX - DTP-Terminal-Ein-/Ausgabe-Exit-Schnittstelle ... 25
NCITIDEX - Terminalkennung-Exit-Schnittstelle ... 26
NCIUIDEX - Benutzerkennung-Exit-Schnittstelle ... 27
NCIXIDEX - Transaktionskennung-Exit-Schnittstelle ... 27
Debugging-Möglichkeiten bei der Natural CICS-Schnittstelle ... 28
TWA-Nutzung bei der Natural CICS-Schnittstelle ... 29
5 Natural-CICS-Generierungsparameter ... 31
NCISCPCB-Generierungsparameter ... 32
NCMDIR-Makro-Parameter ... 32
NCMTGD-Makro-Parameter ... 38
NTCICSP-Makro-Parameter ... 44
6 Anpassung von VSAM RRDS Roll Files ... 45
Anzahl der VSAM RRDS Roll Files erhöhen ... 46
Anzahl der VSAM RRDS Roll Files verringern ... 46
Merkmale der VSAM RRDS Roll Files ändern ... 47
7 Natural in CICS MRO-Umgebungen ... 49
Subparameter COMARET im Makro NTCICSP auf ON gesetzt ... 50
Subparameter COMARET im Makro NTCICSP auf OFF gesetzt ... 50

8 CICS-Knotenfehlerprogramm und Freigabe aller Ressourcen bei Zeitüberschreitung einer Natural-Sitzung ... 51
Sitzungsbeendigung erfolgt nicht unter Kontrolle der Natural CICS-Schnittstelle ... 52
Wiederherstellungsmechanismen ... 53
NCIZNEP einrichten ... 54
NCIZNEP-Funktionalität ... 54
Behandlung von Zeitüberschreitungen (Timeouts) ... 56
API USR4006N zum Abrufen oder Abbrechen von Sitzungen ... 57
9 Unterstützung der CICS 3270 Bridge ... 59
Standardmäßige Unterstützung der CICS 3270 Bridge ... 60
Vollständige Unterstützung der CICS 3270 Bridge ... 60
NCIXFATU - NCI-Lademodul ... 60
Profilparameter DSC=OFF empfohlen ... 60
10 Überlegungen zur Thread-Sicherheit ... 61
11 Überlegungen zur CICS Open Transaction-Umgebung ... 63
12 Unterstützung von CICS-Channels und -Containern ... 65
13 Natural CICS-Schnittstelle und IBM Language Environment (LE) ... 67
14 Spezielle Natural CICS-Funktionalität ... 69
Aufruf von Nicht-Natural-Programmen ... 70
Dummy-Bildschirm-Ein-/Ausgabe bei Natural unter CICS ... 71
15 Natural CICS-Beispielprogramme ... 73
Frontend-Programme ... 74
Backend-Programme ... 75
User Exits ... 75
Subprogramm-Aufrufe (Calls) ... 76
Sonstige Programme ... 76
16 Aufruf von Natural aus Benutzerprogrammen ... 77
Kommandos zum Aktivieren einer Natural-Sitzung ... 78
Frontend-Parameter ... 80
Frontend-Aufruf über LINK ... 81
Frontend-Aufruf über RETURN IMMEDIATE ... 82
Frontend-Aufruf über START ... 82
Frontend-Aufruf über XCTL ... 82
Frontend-Aufruf über Distributed Program Link (DPL) ... 83
Aufruf eines Frontend-Programms als Backend ... 83
17 Asynchrone Natural-Verarbeitung unter CICS ... 85
Asynchrone Natural-Verarbeitung ... 86
Asynchrone Natural-Sitzungen unter CICS ... 86
Testen und Debuggen ... 87
18 Protokollieren von Natural-Sitzungen unter CICS ... 89
Möglichkeiten der Protokollierung ... 90
Definition der Natural-Protokolldatei ... 90
Natural-Protokollsätze (Log Records) ... 91
19 Natural CICS Performance-Überlegungen ... 95

Wahl zwischen Roll Server und Roll Facilities ... 96
Auswahl der Roll Facility ... 97
Shared Storage Threads im Vergleich zu GETMAINed Threads ... 100
CICS-Parameter-Einstellungen ... 102
Zeilenkompromierungssysteme (Line Compression) ... 103
Pseudokonversationelle im Vergleich zu konversationellen Transaktionen ... 103
Natural und Adabas ... 104
CICS-Monitoring-Produkte ... 104
20 Natural-Druck- und Arbeitsdateien unter CICS ... 105
Angepasste Nutzung von Druck- und Arbeitsdateien ... 106
CICS-Zwischenspeicher für Druck- und Arbeitsdateien ... 106
Transiente CICS-Daten bei Druck- und Arbeitsdateien ... 107
III Natural unter Com-plete/SMARTS ... 109
21 Natural unter Com-plete/SMARTS ... 111
Treiberparameter für die Natural Com-plete/SMARTS-Schnittstelle ... 112
Verwendung der Abend-Exits ... 112
Speichernutzung ... 113
Unterstützung für Backend-Programme ... 114
Unterstützung von Com-plete in Natural-Batch-Läufen ... 114
Asynchrone Natural-Verarbeitung unter Com-plete/SMARTS ... 114
Aufrufen von Natural aus Benutzerprogrammen ... 115
Behandlung von Speicherschutzschlüsseln ... 115
Unterstützung für die Behandlung von User Exits während der Sitzungsinitialisierung ... 116
Verwendung der SMARTS-Server-Umgebung ... 116
Unterstützung für das Recoverable Session Handling von Com-plete ... 119
Unterstützung von Natural for zIIP unter Com-plete ... 119
IV Natural unter IMS TM ... 121
22 Natural unter IMS TM - Umgebungen ... 123
Übersicht über die IMS TM-Schnittstelle ... 124
IMS TM-Umgebungen ... 125
Dialogorientierte Umgebungen ... 126
Nachrichtenorientierte Umgebung ... 128
Batch Message Processing-Umgebung ... 131
Unterstützung des Natural-Statements WRITE (n) ... 132
SET CONTROL 'N' (Terminalkommando %N) ... 135
Unterstützung von TS=ON bei Natural unter IMS TM-Nachrichten ... 135
SENDER-Ziel ... 135
Unterstützung des Natural-Profilparameters PROGRAM ... 136
Natural Development Server- bzw. Natural Web I/O Server-Umgebung ... 136
23 Natural unter IMS TM - Komponenten ... 139
Frontend-Modul ... 140
Natural IMS TM-Schnittstellenmodul NIIINTFM ... 141

Editierroutine für physische Eingaben (Physical Input Edit Routine) 142
Benutzernachrichtentabelle (User Message Table) DFSCMTU0 142
Roll File und Roll Server 143
Authorized Services Manager 144
Gemeinsam genutzter Natural-Nukleus 145
Natural Buffer Pool 145
Adabas-Schnittstelle 145
Preload-Liste 146
24 Natural unter IMS TM - Konfiguration 147
NIMMSGT-Makro-Parameter 148
NIMPIXT-Makro-Parameter 149
NIMBOOT-Makro-Parameter 150
25 Natural unter IMS TM - Serviceprogramme 153
Einführung in die Serviceprogramme der Natural IMS TM-Schnittstelle 154
Beschreibung der Serviceprogramme der Natural IMS TM-Schnittstelle 155
NIIBRCST - Übergebene Nachricht an Terminal senden 155
NIICMD - IMS TM-Kommando an IMS TM übergeben 155
NIIDEFT - Aufgeschobenen Wechsel zu Natural-Transaktionscode vorbereiten 156
NIIDEFTX - Aufgeschobenen Wechsel zu nicht-Natural-Transaktionscode vorbereiten 156
NIIDIRT - Direkten Wechsel zu Natural-Transaktionscode vorbereiten 157
NIIDIRTX - Direkten Wechsel zum Transaktionscode vorbereiten 158
NIIEMOD - Einstellung des Modul-Ausgabe-Deskriptors ändern 159
NIIGCMD - Nächstes Antwortsegment des vorherigen IMS TM-Kommandos abrufen 159
NIIGMSG - Erstes Segment der nächsten Nachricht abrufen 160
NIIGSEG - Nächstes Segment der Eingabenachricht abrufen 160
NIIGSPA - Daten ab SPA-Beginn abrufen 161
NIIIMSIN - IMS TM-Umgebungsinformationen abrufen 161
NIIISRTF - Mehrere Segmente enthaltende Nachrichten erstellen 162
NIIISRTM - Nachrichtensegment in Nachrichtenwarteschlange einfügen 163
NIIPCBAD - PSB-Name und PCB-Adresse zurückgeben 163
NIIPCOM - Daten in Antwortbereich verschieben 164
NIIPMSG - Nachricht senden 164
NIIPSBAD - PSB-Adresse zurückgeben 165
NIIPSPA - Daten in der SPA ersetzen 165
NIIPURG - PURG-Aufruf ausgeben 166
NIIRETRM - Daten in den Nachrichtenbereich verschieben 166
NIISASD - SENDER- und OUTDEST-Einstellungen ändern 167
NIIU3962 - Sitzung beenden 167
26 Natural unter IMS TM - Servicemodule 169
Zweck der Servicemodule 170
Beschreibungen der Servicemodule 170

CMCMMND - IMS TM-Operator-Kommandos ausgeben 170
CMDEFSW - Aufgeschobenen Transaktionswechsel zu einem Natural-Transaktionscode ausführen 171
CMDEFSWX - Aufgeschobenen Transaktionswechsel zu einem Nicht-Natural-Transaktionscode durchführen 171
CMDIRNMX - Wechsel zu einer anderen konversationellen Transaktion ohne Nachricht durchführen 172
CMDIRNMZ - Wechsel zu einer anderen konversationellen Transaktion ohne Nachricht durchführen 172
CMDIRSWX - Wechsel zu einer anderen konversationellen Transaktion mit Nachricht durchführen 173
CMDIRSWZ - Wechsel zu einer anderen konversationellen Transaktion mit Nachricht durchführen 174
CMDISPCB - PCB-Inhalt holen 175
CMEMOD - MOD-Namen dynamisch ändern 175
CMGETMSG - Nächste Nachricht lesen 176
CMGETSEG - Nächstes Segment lesen 176
CMGETSPA - Daten aus SPA übertragen 177
CMGSEGO - Nächstes Segment lesen 177
CMIMSID - MVS-Subsystemkennung abfragen 178
CMIMSINF - Informationen zur Systemumgebung abfragen 178
CMPCBADR - PCB-Adresse zurückgeben 179
CMPRNTR - Standard-Hardcopy-Ziel ändern 179
CMPUTMSG - Ausgabemeldung in den IO-PCB einfügen 180
CMPUTSPA - Daten in die SPA verschieben 180
CMQTRAN - Inhalt des aktuellen Transaktionscode-Tabelleneintrags zurückgeben 181
CMQUEUE - Nachricht in alternativen PCB einfügen 181
CMQUEUEX - Vollständige Kontrolle über den Nachrichteninhalt erlangen 182
CMSNFPRT - Logischen Gerätenamen setzen 183
CMSVC13D - Natural-Sitzung abbrechen 183
CMTRNSET - SPA über den alternativen PCB einfügen 183
NIIDDEFS - Aufgeschobenen Wechsel zu einer Fremdtransaktion durchführen 184
NIIDPURG - Multisegment-Nachricht versenden 184
NIIDQUMS - Multisegment-Nachrichten erstellen 185
NIIDSETT - Fremdtransaktionscode abfragen 185
27 Natural unter IMS TM - User Exits 187
NIIXACCT 188
NIIXSTAR 188
NIIXSSTA 188
NIIXISRM 189
NIIXISRT 189
NIIXTGU0 189

NIIXJESA 189
NIIXPRT0 189
NIIXRFNU 190
NIIXTGN0 190
28 Natural unter IMS TM - Spezielle Funktionen 191
Voraussetzungen 192
Accounting (Abrechnung) 192
Monitoring (Überwachung) 194
Broadcasting (Rundsenden) 194
Server-Umgebung 195
29 Natural unter IMS TM - Wiederherstellungsbehandlung 199
System- und Benutzerabbrüche 200
Nicht behebbare Fehler 200
Behebbare Fehler 201
V Natural unter TSO 203
30 Natural unter TSO 205
Allgemeine Informationen über die Natural-TSO-Schnittstelle 206
Treiberparameter für die Natural TSO-Schnittstelle 206
Von Natural unter TSO verwendete Datasets 206
TSO-Kommandos aus Natural unter TSO absetzen 210

Preface

Diese Dokumentation enthält detaillierte Informationen über den Betrieb von Natural mit den unterstützten TP-Monitor-Systemen.

Betrieb von Natural mit TP-Monitoren	
Verwendung von Natural mit TP-Monitoren	Enthält allgemeine Informationen zum Betrieb von Natural mit TP-Monitoren.
TP-Monitor-Schnittstellen	
Natural unter CICS	Beschreibt die Funktionalität der Natural-CICS-Schnittstelle (Produktcode: NCI) sowie den Betrieb und die einzelnen Komponenten von Natural in einer CICS-Umgebung.
Natural unter Com-plete/SMARTS	Beschreibt die Funktionalität der Natural Com-plete/SMARTS-Schnittstelle (Produktcode: NCF) sowie den Betrieb und die einzelnen Komponenten von Natural in einer Com-plete-Umgebung.
Natural unter IMS TM	Beschreibt die Funktionalität der Natural IMS TM-Schnittstelle (Produktcode: NII) sowie den Betrieb und die einzelnen Komponenten von Natural in einer IMS TM-Umgebung.
Natural unter TSO	Beschreibt die Funktionalität der Natural TSO-Schnittstelle sowie den Betrieb und die einzelnen Komponenten von Natural unter TSO (Produktcode NTI) im Betriebssystem z/OS.

1 About this Documentation

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

Dokumentationskonventionen

Konvention	Beschreibung
Fettschrift	>Kennzeichnet Elemente auf einem Bildschirm.
`Nichtproportionale Schrift`	Kennzeichnet Namen und Orte von Diensten im Format *`Ordner.Unterordner.Dienst`*, Programmierschnittstellen (APIs), Namen von Klassen, Methoden und Properties in Java.
`Kursivschrift`	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 Alarme 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.

I Verwendung von Natural mit TP-Monitoren

2 Verwendung von Natural mit TP-Monitoren

- Von Natural unterstützte TP-Monitor-Systeme 8
- Einsatz von Natural in einer Datenfernverarbeitungsumgebung (Teleprocessing, TP) 8

Natural bietet Schnittstellen, über die der Natural-Nukleus auf einen TP-Monitor für die Online-Transaktionsverarbeitung und auf ein Betriebssystem (OS) für die Batch-Verarbeitung zugreifen kann.

Allgemeine Informationen zu den TP-Monitor-Schnittstellen finden Sie auch im Abschnitt *TP/OS-Schnittstelle* in der *Natural-System-Architektur*-Dokumentation.

Listen und Beschreibungen der spezifischen Meldungen und Codes, die bei der Verwendung der TP-Monitor-Schnittstellen auftreten können, finden Sie unter *TP-Monitor-Schnittstellen Meldungen und Codes* in der *Meldungen und Codes*-Dokumentation.

Notation `vrs` oder `vr`

In dieser Dokumentation steht die Schreibweise `vrs` oder `vr` für die jeweilige Produktversion (siehe auch *Version* im *Glossar*).

Von Natural unterstützte TP-Monitor-Systeme

Natural unterstützt die folgenden Teleprocessing-Monitorsysteme:

- **CICS**
- **Com-plete**
- **IMS TM**
- **TSO**

Einsatz von Natural in einer Datenfernverarbeitungsumgebung (Teleprocessing, TP)

In diesem Abschnitt werden die folgenden Themen behandelt:

- Einbettung von Natural in eine TP-Umgebung
- Relevante Natural-Profilparameter
- Aufrufen von Natural-Transaktionen unter einem TP-Monitor
- Überwachung und Steuerung TP-Monitor-spezifischer Natural-Merkmale
- Beenden einer Natural-Sitzung

- Beispiel-Programme

Einbettung von Natural in eine TP-Umgebung

In einer Teleprocessing-Monitor-Umgebung arbeitet Natural wie ein Standard-TP-Programm und befolgt die Regeln, die für Programme gelten, die unter der Kontrolle dieses TP-Monitors ausgeführt werden.

Da der Natural-Code vollständig reentrant ist, wird er von allen Natural-Benutzern gemeinsam genutzt und es existiert nur ein Arbeitsbereich pro Benutzer (und nur für die Dauer der Natural-Sitzung dieses Benutzers).

Natural-Anwenderprogramme ("Transaktionen") können zusammen mit nativen TP-Programmen ausgeführt werden, um ein integriertes System zu bilden, das sowohl Natural-Programme als auch herkömmliche Programme einbezieht.

Relevante Natural-Profilparameter

Es gibt spezifische Natural-Profilparameter, die gelten, wenn Natural mit einem TP-Monitor verwendet wird.

Einen Überblick über diese Parameter finden Sie unter *TP-Monitor-Schnittstellen* im Kapitel *Profilparameter sortiert nach Kategorien* in der Natural-*Parameter-Referenz*-Dokumentation.

Aufrufen von Natural-Transaktionen unter einem TP-Monitor

Natural-Transaktionen können aufgerufen werden, indem Sie das TP-Programm mit dem Namen `Natural` aufrufen und das Systemkommando `LOGON` sowie den Namen der auszuführenden Natural-Transaktion im Stack angeben.

Beim Aufruf von Natural können mehrere Kommandos/Transaktionen und Eingabedaten für die Kommandos/Transaktionen über den Stack übergeben werden.

Überwachung und Steuerung TP-Monitor-spezifischer Natural-Merkmale

Das Natural-Dienstprogramm SYSTP bietet verschiedene Funktionen, mit denen Sie TP-Monitor-spezifische Eigenschaften von Natural überwachen und steuern können.

SYSTP ist für die TP-Monitore Com-plete, CICS, IMS TM und TSO verfügbar.

Weitere Informationen siehe SYSTP Utility in der *Debugger und Dienstprogramme*-Dokumentation.

Beenden einer Natural-Sitzung

Die Natural-Sitzung kann durch Ausführen des Natural-Statements `TERMINATE` oder des Systemkommandos `FIN` beendet werden.

Beispiel-Programme

Die Natural-Library `SYSEXTP` enthält mehrere Beispielprogramme für spezifische Funktionen, die nur unter bestimmten TP-Monitoren gelten.

II Natural unter CICS

Dieser Teil beschreibt die Funktionalität der Natural CICS-Schnittstelle (Produktcode NCI) sowie den Betrieb und die einzelnen Komponenten von Natural in einer CICS-Umgebung.

- **Unterstützung für zIIP unter CICS**
- **Funktionsweise der Natural CICS-Schnittstelle**
- **Natural-CICS-Generierungsparameter**
- **Anpassung von VSAM RRDS Roll Files**
- **Natural in CICS MRO-Umgebungen**
- **CICS-Knotenfehlerprogramm und Freigabe aller Ressourcen bei Zeitüberschreitung einer Natural-Sitzung**
- **Unterstützung der CICS 3270 Bridge**
- **Überlegungen zur Thread-Sicherheit**
- **Überlegungen zur CICS Open Transaction-Umgebung**
- **Unterstützung von CICS-Channels und -Containern**
- **Natural CICS-Schnittstelle und IBM Language Environment (LE)**
- **Spezielle Natural CICS-Funktionalität**
- **Natural CICS-Beispielprogramme**
- **Aufruf von Natural aus Benutzerprogrammen**
- **Asynchrone Natural-Verarbeitung unter CICS**
- **Protokollieren von Natural-Sitzungen unter CICS**
- **Natural CICS Performance-Überlegungen**
- **Natural-Druck- und Arbeitsdateien unter CICS**

Note: Die Notation `vrs` oder `vr` steht für die jeweilige Produktversion (siehe auch *Version* im *Glossar*).

Verwandte Themen:

- *Natural CICS-Schnittstelle auf z/OS installieren* in der Natural-*Installation für z/OS*-Dokumentation
- *CICSP - Umgebungsparameter für Natural CICS-Interface*
- *Natural unter CICS-Meldungen* in der *Meldungen und Codes*-Dokumentation
- *SYSTP Utility* - dieses Natural-Dienstprogramm bietet verschiedene TP-Monitor-spezifische Funktionen.
- *Natural als Server unter CICS* in der Natural-*Operations*-Dokumentation.

3 Unterstützung für zIIP unter CICS

Die Natural CICS-Schnittstelle unterstützt IBMs System z Integrated Information Processors (zIIPs) in einer CICS-Umgebung unter z/OS.

Informationen zur Unterstützung von Natural for zIIP und den erforderlichen Voraussetzungen finden Sie in der *Natural for zIIP*-Dokumentation.

Informationen zu den Änderungen bei der Installation finden Sie unter *Natural CICS-Schnittstelle auf z/OS installieren* und *Natural for zIIP auf z/OS installieren*.

4 Funktionsweise der Natural CICS-Schnittstelle

- Natural CICS-Schnittstelle 16
- Umgebungsabhängiger Natural Nukleus für CICS 16
- Systemsteuerung unter CICS 17
- Natural-System-Komponenten im dynamischen CICS-Speicher 18
- Natural-Speicher-Threads unter CICS 21
- Natural Roll Facilities unter CICS 22
- CICS Roll Facilities 22
- Lokaler Natural Buffer Pool unter CICS 23
- Systemsteuerungssätze der Natural CICS-Schnittstelle im CICS-Zwischenspeicher 23
- NCIDIREX - Systemverzeichnismodulnamen-Exit-Schnittstelle 24
- NCIDTPEX - DTP-Terminal-Ein-/Ausgabe-Exit-Schnittstelle 25
- NCITIDEX - Terminalkennung-Exit-Schnittstelle 26
- NCIUIDEX - Benutzerkennung-Exit-Schnittstelle 27
- NCIXIDEX - Transaktionskennung-Exit-Schnittstelle 27
- Debugging-Möglichkeiten bei der Natural CICS-Schnittstelle 28
- TWA-Nutzung bei der Natural CICS-Schnittstelle 29

Dieses Kapitel beschreibt die Funktionsweise der Natural CICS-Schnittstelle. Folgende Themen werden behandelt:

Natural CICS-Schnittstelle

Die Natural CICS-Schnittstelle steuert die Sitzungsinitialisierung, den Roll-In-Neustart (im pseudokonversationellen Modus), die Terminal-Ein-/Ausgaben, den Datenbankzugriff, die Abbruchverarbeitung (Abend), die Aufrufe des Natural Local Buffer Pools sowie das Laden, Verlinken und Freigeben von externen Subprogrammen. Darüber hinaus werden alle Aus-/Einspeicherungsoperationen (Roll Out/Roll In) von der Natural CICS-Schnittstelle ausgeführt.

Umgebungsabhängiger Natural Nukleus für CICS

Der umgebungsabhängige Natural-Nukleuss (Beschreibung siehe *Umgebungsabhängiger Nukleus* in der Natural-*Installation für z/OS*-Dokumentation) für CICS besteht aus den folgenden Komponenten:

- **Objektmodul** `NCINUC`
  Dieses Modul enthält die Systemsteuerungslogik der Natural CICS-Schnittstelle und die Logik, die für den Aufruf von Betriebssystem- und CICS-Diensten erforderlich ist.

  Dieses Modul enthält auch die Einstiegsroutine, die insbesondere die Verlinkung der Natural CICS-Schnittstelle mit der LE-Umgebung (Language Environment) vorbereitet. Siehe ***Natural CICS-Schnittstelle und IBM Language Environment (LE)***.
- **Makro NTCICSP**
  Das Makro `NTCICSP` enthält Natural CICS-Schnittstellenparameter, die für Laufzeit- und Systemumgebungsgenerierungsoptionen erforderlich sind. Das Modul ist nicht von der CICS-Version abhängig, obwohl einige der Parameter je nach CICS-Version gesetzt werden sollten.

  Das Makro `NTCICSP` befindet sich im Natural-Parametermodul. Siehe *CICSP - Umgebungsparameter für Natural CICS-Schnittstelle* in der *Parameter-Referenz*-Dokumentation).
- **Objektmodul** `NCIXCALL`**.**
  Dieses Modul ist ein separates Programm in CICS, d.h. es ist nicht mit dem Natural-Nukleus verlinkt, da es über `EXEC CICS LINK` von 3GL-Programmen aufgerufen wird, die von Natural aufgerufen werden. Siehe *Natural 3GL CALLNAT-Schnittstelle verwenden* in der *Operations*-Dokumentation. Das Modul ist unabhängig von der CICS-Version.

CICS Shutdown unter Natural

Der umgebungsabhängige Natural-Nukleus kann in den CICS PLTSD für die Ausführung von CICS Quiesce Stage 1 oder 2 aufgenommen werden.

- Bei Ausführung in der Quiesce Stage 1 erzwingt die Natural-CICS-Schnittstelle die Beendigung aller aktiven Natural-Sitzungen, bevor die SYSTP-Snapshot-Funktion ausgeführt wird (Beschreibung siehe *Debugger und Dienstprogramme*-Dokumentation).
- Bei Ausführung in der Quiesce Stage 2 führt die Natural-CICS-Schnittstelle die SYSTP-Snapshot-Funktion aus.

Die Natural CICS-Schnittstelle enthält eine Logik, die (über einen CICS LINK) von einem Knotenfehlerprogramm mit der entsprechenden CICS-Terminaleingabeadresse auch in der CICS COMMAREA aufgerufen wird.

Systemsteuerung unter CICS

Zu den CICS-spezifischen Natural-Merkmalen gehören die Organisation des dynamischen Speichers in Threads und die zusätzliche Fähigkeit, diese Threads so zu behandeln, dass das Natural CICS System Control Program den dynamischen Speicher effizienter verwalten kann.

Das Natural CICS System Control Program wurde ursprünglich entwickelt, um die 64 KB GETMAIN-Grenze unter CICS zu überwinden. Es bietet vollständige Speicherzuordnungs- und -verwaltungsfunktionen, einschließlich Roll File I/O-Operationen und Verschiebungsfunktionen für pseudo-konversationelle Benutzer.

Um die pseudokonversationellen Verarbeitungsmöglichkeiten von Natural mit CICS zu verbessern, verwendet das System Control Program Threads, d.h. eine zusammenhängende Speicherplatzmenge, der für jeden Benutzer eingerichtet wird. Dank dieser Struktur kann Natural den dynamischen Speicher unter minimaler CICS-Beteiligung verwalten.

Die folgenden Ausführungen zu Struktur und Funktionsweise des System Control Program vermitteln Ihnen ein umfassendes Verständnis der Systemsteuerung. Sie sollten diesen Mechanismus verstehen, bevor Sie mit der Installation von Natural unter CICS beginnen.

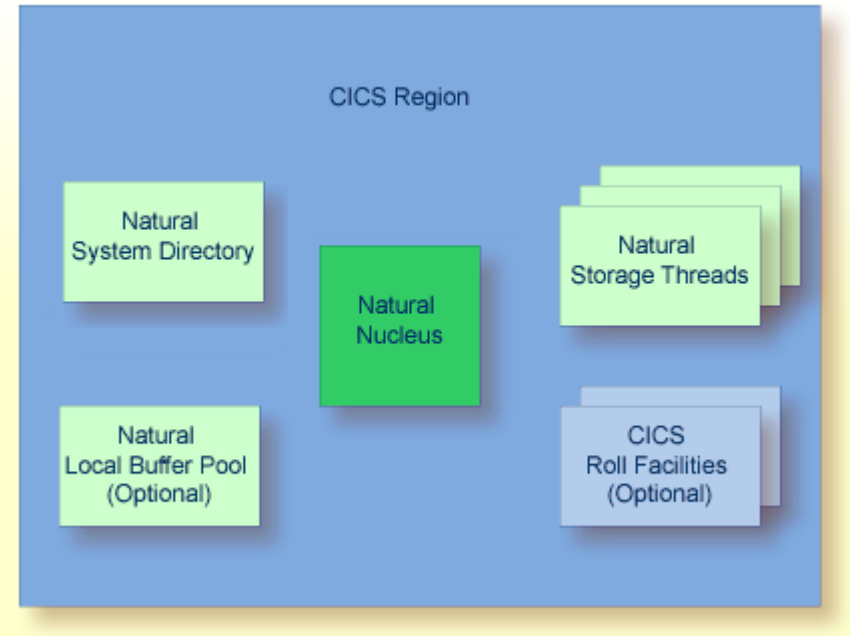
Natural-System-Komponenten im dynamischen CICS-Speicher

Scenario 1:

Einzelne CICS-Region

Das folgende Diagramm zeigt die Natural-System-Komponenten des Natural-Systems, die sich im dynamischen CICS-Speicher befinden. Diese Komponenten werden unter den folgenden Überschriften erläutert:

- *Natural-Speicher-Threads unter CICS*
- *Lokaler Natural-Buffer-Pool unter CICS*
- *Natural Roll Facilities*

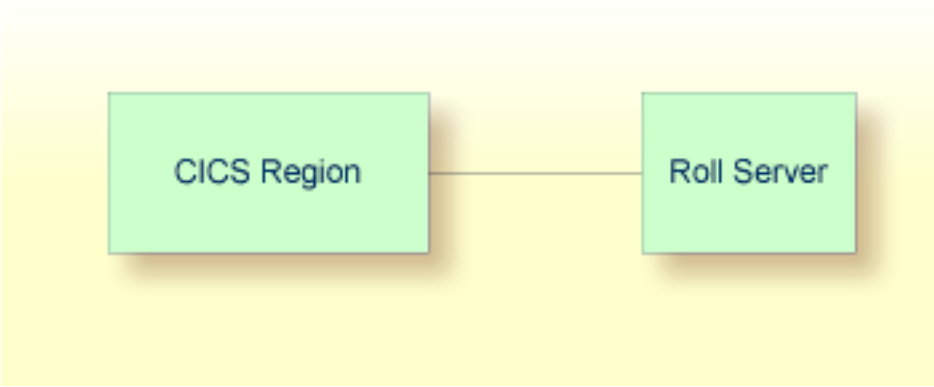


Szenario 1 gilt für den lokalen Betrieb von Natural in einer einzelnen CICS-Anwendungsregion.

Weitere Szenarien sind möglich. Die folgenden drei Diagramme zeigen Kombinationen von z/OS-Systemen, CICS-Regionen, dem *Natural Roll Server* und dem *Natural Authorized Services Manager* (Beschreibung siehe *Operations*-Dokumentation).

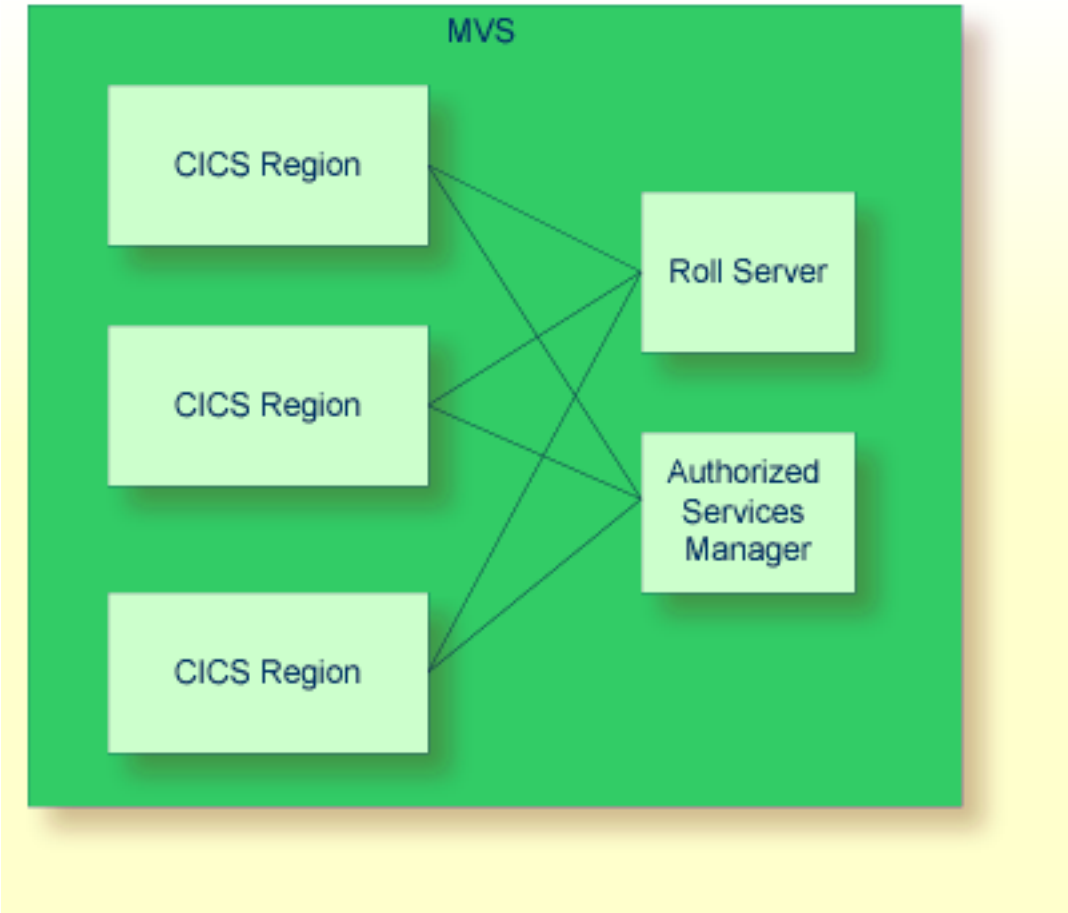
Scenario 2:

Einzelnes z/OS mit einzelner CICS-Region, einzelner Roll Server



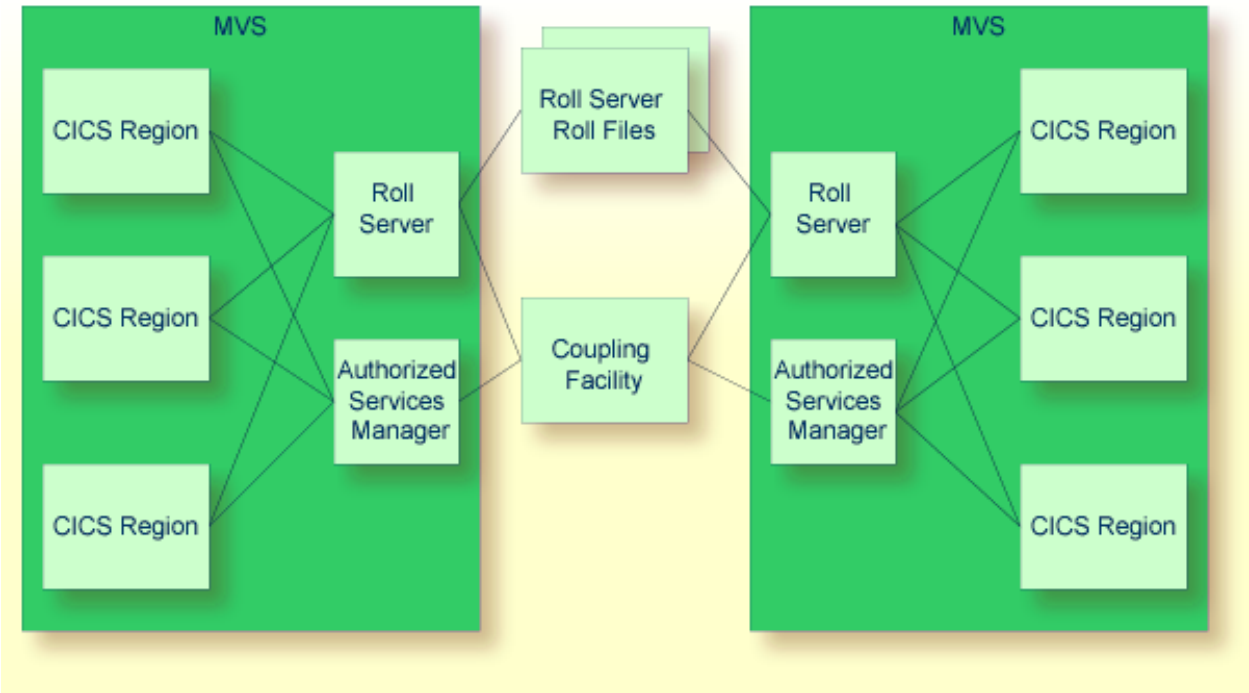
Scenario 3:

Einzelnes z/OS mit mehreren CICS-Regionen, einzelnem Roll Server und (optional) Authorized Services Manager



Scenario 4:

Mehrere z/OS mit mehreren CICS-Regionen, mehreren Roll Servern/Authorized Services Managern



Erforderliche Parametereinstellungen für die obigen Szenarien

Modul	Scenario 1	Scenario 2	Scenario 3	Scenario 4
NTBPI (BPI)	TYPE=NAT,SIZE=*nnn*	nicht zutreffend	nicht zutreffend	nicht zutreffend
NCMDIR CICSPLX	NO	NO	YES/MODE	YES/MODE
NCMDIR SIPSERV	NO	NO/YES	YES	YES
NCMDIR ROLLSRV	NO	YES	YES	YES
Roll Server CF-Strukturname	nicht zutreffend	keiner	keiner	*name*
Authorized Services Manager/SIP	nicht zutreffend	nicht zutreffend	SIP-Slot-Nummer/Größe	XCF-Gruppenname/CF-Strukturname

Die Natural CICS-Schnittstelle erfordert eine SIP-Slotgröße von 256 Bytes.

Note: Für die Szenarien 2, 3 und 4 muss bei der allerersten Natural-Sitzung, die die NCI-Umgebung initialisiert, der Profilparameter SUBSID auf den Wert des entsprechenden *Roll Server* und/oder *Authorized Services Manager* gesetzt werden.

Natural-Speicher-Threads unter CICS

Ein Thread ist ein zusammenhängender Speicherbereich, aus dem Natural seinen gesamten benötigten Speicher anfordert. Dabei kann es sich entweder um Speicher handeln, der von mehreren Natural-Benutzern gemeinsam genutzt wird, oder, in 31-Bit-Modus-Umgebungen, um CICS-Benutzerspeicher oberhalb der 16 MB-Grenze, der für eine bestimmte Aufgabe vorgesehen ist.

Jeder Speicher-Thread kann als der "Adressraum" für einen Natural-Benutzer angesehen werden. Jede vom Natural-Nukleus ausgegebene Speicherzuweisungsanforderung wird an das **Systemsteuerungsprogramm** weitergeleitet, damit sie vom Speicher-Thread erfüllt wird.

Speicher-Threads werden bei der Initialisierung der Natural-CICS-Schnittstelle zugeordnet. Sie werden in einer CICS-Region oder in einer CICS-Partition zugewiesen; in diesem Fall handelt es sich um permanente (gemeinsam genutzte) Threads, oder sie werden beim Start einer Natural-CICS-Task zugewiesen. In diesem Fall handelt es sich um exklusive Threads (task-abhängiger Benutzerspeicher).

Die Technik der Speicher-Threads wurde in Natural aus folgenden Gründen eingeführt:

- Um die 64-KB-Beschränkung von CICS für Benutzerspeicher in Systemen, die nicht im 31-Bit-Modus arbeiten, zu überwinden.
- Um das Aus-/Einspeichern (Rolling) zu optimieren (früher musste jeder Teil des Benutzerspeichers auf das Roll-Medium geschrieben werden. Jetzt, da es einen zusammenhängenden Speicherbereich gibt, wird dieser Bereich komprimiert, indem die betreffenden Teile vor dem Rolling aneinandergefügt werden).
- Die Natural-CICS-Schnittstelle versucht, alle `GETMAIN`-Anforderungen einer Natural-Sitzung von ihrem Thread aus zu erfüllen. Dies ist schneller als `GETMAIN`-Anforderungen durch CICS-Dienstaufrufe. Dies gilt insbesondere für CICS-Kommandoebenenaufrufe, da auch das CICS `EXEC` Interface Program (`EIP`) beteiligt ist.

Bei jeder Bildschirm-E/A wird ein Thread von der besitzenden Task freigegeben. Dies gilt sowohl für konversationelle als auch für pseudokonversationelle Tasks. Wenn eine Sitzung wieder aufgenommen wird, wird ihr Speicher wieder in einen Thread verlagert (Roll In), es sei denn, der Speicher ist noch vorhanden, d.h. keine andere Task hat den Thread zwischenzeitlich benutzt.

Der Natural-Thread-Auswahlalgorithmus gleicht die Thread-Nutzung so aus, dass die Roll I/O-Operationen minimiert werden. Das heißt, je mehr Threads es gibt, desto größer ist die Chance, die alten Daten zu finden und damit ein Roll In zu verhindern. Je mehr Threads es jedoch gibt, desto mehr Paging muss das Betriebssystem durchführen, um alle Threads effizient im realen Speicher zu halten.

Threads werden je nach ihrer Größe und ihrem Typ gruppiert, d.h. je nachdem, ob sie als permanent gemeinsam genutzter Speicher oder über eine `GETMAIN`-Anforderung vorab zugewiesen wurden. Die Entscheidung, welche Art von Thread-Gruppe zu verwenden ist, wird durch den CICS-

Transaktionscode bei der Sitzungsinitialisierung gesteuert. Alle Speicher-Threads, die zur gleichen Gruppe gehören, haben die gleiche Größe.

Der Thread sollte so klein wie möglich definiert werden; siehe auch die Funktion *Buffer Usage Statistics* des Dienstprogramms SYSTP (Beschreibung siehe *Debugger und Dienstprogramme*-Dokumentation). Der Thread muss jedoch immer noch groß genug sein, um die Sitzung mit den größten Größen zu halten.

Wenn Sie über getrennte Natural-Entwicklungs- und -Produktionsumgebungen verfügen, sollten Sie in der Regel mehr kleinere Threads in der Produktionsumgebung (um Produktionsanforderungen so schnell wie möglich zu bedienen) und eine geringere Anzahl größere Threads in der Entwicklungsumgebung haben (da Natural-Programmierer normalerweise größere Natural-Größen benötigen und längere "Denkpausen" einlegen).

Bei der allerersten Natural-Sitzung werden alle permanenten (gemeinsam genutzten) Threads zugeordnet.

Natural Roll Facilities unter CICS

Da permanente Speicher-Threads von mehreren Benutzern gemeinsam genutzt werden und größere, über `GETMAIN` zugeordnete Threads nicht zu lange gehalten werden sollten, gibt eine Natural-Task ihren Thread bei jeder Terminal-Ein-/Ausgabe frei. Zuvor müssen jedoch die Benutzerdaten gespeichert werden, um die Natural-Sitzung nach erfolgter Terminal-E/A wieder starten zu können.

Die Sitzungsdaten können gespeichert werden, unter Verwendung

- des Natural Roll Server mit seinem lokalen Roll Buffer und den Roll Files,
- der CICS Roll Facilities.

Siehe auch die verschiedenen **Komponentenszenarien**. Weitere Informationen finden Sie unter *Roll Server* in der Natural-*Operations*-Dokumentation.

CICS Roll Facilities

Die CICS Roll Facilities sind lokale CICS-Speichereinrichtungen. Dabei kann es sich entweder um CICS-Haupt- oder Hilfsspeicher oder um VSAM Relative Record Data Sets (RRDS) handeln, die der Benutzer zuvor für CICS definiert hat. Diese Dateien ermöglichen Natural die Speicherung des komprimierten dynamischen Speichers eines Benutzers, wenn eine Ausspeicherung (Roll Out) stattfindet.

Jede CICS-Dienstanforderung verursacht einen CICS-System-Overhead. Je größer also die `CISIZE`/Datensatzgröße für die Roll Facility ist, desto weniger CPU-Overhead entsteht, da weniger CICS-Dienstaufrufe zum Auslagern einer Natural-Sitzung erforderlich sind. Andererseits bedeutet eine größere `CISIZE`-/Datensatzgröße auch, dass mehr VSAM-Pufferspeicher für die Roll Facility zugewiesen wird.

Weitere Informationen zu Roll Facilities finden Sie unter ***Natural CICS Performance-Überlegungen***.

Caution: Wenn der Roll Server verwendet wird, sind die CICS Roll Facilities nicht verfügbar.

Lokaler Natural Buffer Pool unter CICS

Der lokale Natural Buffer Pool enthält alle Natural-Module während der Ausführung und Kopien von Natural-Modulen, nachdem diese aus der Adabas- oder VSAM-Systemdatei geladen wurden.

Der lokale Buffer Pool muss groß genug sein, um die Anzahl der Natural-Programmladevorgänge zu minimieren. Ist der lokale Buffer Pool jedoch zu groß, bedeutet dies eine Verschwendung von Speicherplatz und kann zu Paging-Overhead führen.

Der lokale Buffer Pool wird als `GETMAIN`-Speicher zugewiesen, d.h. `EXEC CICS GETMAIN SHARED` bei allen CICS Transaction Server-Versionen. In der Partition oder im entsprechenden CICS-DSA muss ausreichend Speicherplatz vorhanden sein.

Ein lokaler Buffer Pool ist optional, da Natural auch mit einem globalen Buffer Pool laufen kann, der mit anderen Natural-Umgebungen wie *Natural im Batch-Modus*, ***Natural unter TSO*** oder ***Natural unter IMS*** gemeinsam genutzt werden kann.

Systemsteuerungssätze der Natural CICS-Schnittstelle im CICS-Zwischenspeicher

Die Natural CICS-Schnittstelle vermerkt ihre permanenten `GETMAIN`-Speicher, d.h. Speicher, die über `EXEC CICS GETMAIN SHARED` in NCI-Systemsteuerungsdatensätzen im CICS-Hauptzwischenspeicher angelegt wurden.

Diese Systemsteuerungssätze werden aus zwei Gründen aufbewahrt:

1. **Systemwiederherstellung**
   Da alle NCI-bezogenen Speicher mit dem NCI-Systemverzeichnis verkettet sind, können die Systemsteuerungssätze verwendet werden, um die Speicherketten im Falle einer Speicherbeschädigung wiederherzustellen.

2. **Bereinigung**
   Zur Bereinigung des alten NCI-Systems nach CICS `NEWCOPY` des NCI-Systemverzeichnismoduls.

   Bei der Initialisierung der NCI-Systemumgebung prüft NCI, ob Systemsteuerungsdatensätze vorhanden sind, und gibt, falls welche gefunden wurden, die zugehörigen permanenten Speicher vor der Installation der neuen Umgebung frei.

Die Namen der temporären CICS-Speicherwarteschlangen dieser Steuerungsdatensätze (Control Record, `CR`) lauten *prefixX*`CR`, wobei *prefix* das gemeinsame Präfix für Natural CICS-Komponenten ist (siehe Schlüsselwort-Subparameter `PREFIX` des Makros `NTCICSP`) und *X* ein hexadezimaler Wert ist, nämlich

Wert	Erläuterung
`x'01'`	für den Hauptsystemsteuerungsdatensatz, der Informationen über die NCI-Systemverzeichniserweiterung, gemeinsam genutzte (Shared) Threads (`TYPE=SHR`) und sekundäre SIR-Blöcke enthält (siehe `NCMDIR`-Generierungsparameter `USERS`).
`x'02'`	für den `parms`-Systemsteuerungsdatensatz mit Informationen über die Parameter des gemeinsam genutzten NCI-Profils, die per Dateieingabe abgerufen werden (siehe den Parameter `PRMDEST` des Makros `NTCICSP`).
`x'03'`	für den `pools`-Systemsteuerungsdatensatz, der Informationen über alle zur NCI-Umgebung gehörenden lokalen Pools enthält.

Important: Da die NCI-Systemsteuerungssätze eine lokale NCI-Umgebung beschreiben, müssen diese CICS-Hauptzwischenspeicherwarteschlangen auch im CICS-AOR aufbewahrt werden. Dies gilt insbesondere für die Ausführung von Natural in CICSPlex.

NCIDIREX - Systemverzeichnismodulnamen-Exit-Schnittstelle

Der Name des Systemverzeichnismoduls der Natural CICS-Schnittstelle lautet standardmäßig *prefix*`CB` (siehe Schlüsselwort-Subparameter `PREFIX` des Makros `NTCICSP`), sofern er nicht explizit über den Schlüsselwort-Subparameter `DIRNAME` des Makros `NTCICSP` angegeben wird.

Die Exit-Schnittstelle `NCIDIREX` dient zum Setzen/Ändern des Namens des Natural CICS-Schnittstellen-Systemverzeichnismoduls zur Laufzeit. Dadurch ist es möglich, denselben NCI-Treiber bzw. dasselbe Natural-Parametermodul zu verwenden, aber verschiedene NCI-Umgebungen (Thread-Gruppen/Thread-Größen usw.) zu nutzen, indem auf verschiedene Systemverzeichnismodule zugegriffen wird, die z.B. von der CICS-Systemkennung oder der Transaktionskennung abhängen.

Die ersten 5 Zeichen des Verzeichnismodulnamens werden auch als Teil der CICS-Warteschlangennamen für den Zwischenspeicher verwendet, die sich auf die jeweilige NCI-Umgebung beziehen. Wenn also mehr als eine Natural CICS-Umgebung in einer CICS-Region betrieben wird, müssen sich die entsprechenden Systemverzeichnismodulnamen in den ersten 5 Zeichen unterscheiden.

Die Exit-Schnittstelle NCIDIREX wird unter Verwendung der Standard-Linkage-Konventionen (Register 13, 14, 15 und 1) aufgerufen, aber zusätzlich mit den Registern 4 und 5, die CICS-EIB- und EISTG-Adressen enthalten, damit der Exit CICS-Dienste aufrufen kann.

Das Quellcodemodul XNCIDIRX enthält ein Beispiel für einen Systemverzeichnismodulnamen-Exit.

NCIDTPEX - DTP-Terminal-Ein-/Ausgabe-Exit-Schnittstelle

Natural Sessions können auch über Distributed Transaction Processing (DTP), d.h. über APPC- oder MRO-Conversations, ausgeführt werden. Formal ist solchen Natural-Sitzungen ein Terminal zugeordnet (CICS TCTTE), allerdings handelt es sich dabei um ein Terminal aus einem Pool (siehe CICS SESSIONs / CONNECTIONs), und das "Terminal" kann sich von einem Natural-Dialogschritt zum anderen ändern, d.h. solche "Terminals" können nicht als Schlüssel (Key) verwendet werden, um den Kontext einer Sitzung über eine "Bildschirm-Ein-/Ausgabe" zu speichern. Aus diesem Grund werden solche Natural-Sitzungen standardmäßig als asynchrone Sitzungen (TTYPE=ASYN/ASYL) behandelt, und Natural kommuniziert nicht mit diesen Terminals, da sie keine 3270-Geräte sind.

Mit NCIDTPEX steht jedoch eine Exit-Schnittstelle zur Verfügung, über die Sie die Natural-Sitzung in konversationeller Form ausführen können:

- Wenn dieser Exit verfügbar ist, baut Natural eine Terminalsitzung auf (TTYPE=3270).
- Die Natural-Terminal-Ein- und Ausgabeoperationen (RECEIVE/SEND/CONVERSE) werden nicht von Natural verarbeitet, sondern zur Weiterverarbeitung an den Exit übergeben.

Die Quellcode-Module XNCIDTPX und XNCITIOX enthalten Beispiele für DTP-Terminal-Exits.

Verwendung von NCIDTPEX

Sie können den Generierungsparameter FDTPX des Makros NTCICSP auf ON setzen, um zu bewirken, dass ein potenzieller DTP-Exit für alle Terminaltypen aufgerufen wird. Dies kann z.B. hilfreich sein, wenn Sie die Terminalausgabe analysieren wollen, bevor eine EXEC CICS SEND-Operation ausgeführt wird, oder wenn Sie Bildschirm-Ein-/Ausgaben unterdrücken wollen.

NCITIDEX - Terminalkennung-Exit-Schnittstelle

Die aus 4 Zeichen bestehende CICS-Terminalkennung, die pro CICS-Region eindeutig ist, wird von der Natural CICS-Schnittstelle als Teil des Sitzungsschlüssels (Session Key) verwendet (SIP-Server, Roll Server, CICS-Zwischenspeicherwarteschlangen). Aus Gründen der Kompatibilität mit Natural wird für die Natural CICS-Schnittstelle (NCI) ein 8-Zeichen-Feld verwendet. Diese NCI-Terminalkennung kann über mehrere CICS-Regionen hinweg eindeutig gemacht werden, indem die CICS-Systemkennung an die CICS-Terminalkennung angehängt wird (siehe Schlüsselwort-Subparameter `UNITID` des Makros `NTCICSP`).

Alternativ kann die Terminalkennung-Exit-Schnittstelle NCITIDEX verwendet werden, um diese NCI-Terminalkennung zu setzen. Es ist zu beachten, dass für CICS-Zwecke (z.B. Namen von Zwischenspeicherwarteschlangen usw.) nur die ersten vier Zeichen der NCI-Terminalkennung verwendet werden. Daher müssen diese 4-Zeichen-Strings eindeutig sein.

Die Exit-Schnittstelle `NCITIDEX` ist vor allem für Session-Manager unter CICS interessant, um mehrere Natural-Sessions zu unterscheiden, die auf demselben physischen Terminal laufen.

Die durch eine Exit-Schnittstelle `NCITIDEX` gesetzte Terminalkennung wird von der Natural CICS-Schnittstelle “extern” verwendet und ist der Standardwert für die Natural-Systemvariable `*INIT-ID` zur Verwendung in Natural. (Die Systemvariable `*INIT-ID` kann nachträglich durch den User Exit `NCIUIDEX` / `NATUEX1` geändert werden.)

Der Aufruf des Schnittstellen-Exits `NCITIDEX` erfolgt unter Verwendung der Standard-Linkage-Konventionen (Register 13, 14, 15 und 1), aber zusätzlich unter Verwendung der Register 4 und 5, die CICS-EIB- und EISTG-Adressen enthalten, damit dieser Exit CICS-Dienste aufrufen kann.

Das Quellcodemodul `XNCITIDX` enthält ein Beispiel für eine Terminalkennung-Exit-Schnittstelle.

Einschränkungen

Bestimmte Funktionen der Natural CICS-Schnittstelle funktionieren nicht, wenn die ersten vier Zeichen der logischen Terminalkennung nicht mit dem physischen Terminal übereinstimmen.

Dies hat zur Folge:

- Sie können keine Nachricht an ein logisches Terminal via Message Switching senden.
- Sie können das Dienstprogramm SYSTP oder das Knotenfehlerprogramm (Node Error Program, NEP) nicht zum Abbau (Flush) einer Sitzung an einem logischen Terminal verwenden.

NCIUIDEX - Benutzerkennung-Exit-Schnittstelle

Natural stellt die User Exit-Schnittstelle NATUEX1 bereit, mit der Sie feststellen können, ob ein Benutzer berechtigt ist, Natural zu benutzen, und mit der Sie verschiedene Natural-Systemvariablen setzen können.

Immer wenn eine Natural-Benutzersitzung gestartet wird, wird der Schnittstellen-Exit NATUEX1 über die Standard-Linkage-Konventionen aufgerufen (Register 13, 14, 15 und 1).

In einer CICS-Umgebung reichen die Standard-Linkage-Konventionen nicht aus, um CICS-Dienstaufrufe abzusetzen und die Adressierbarkeit von CICS-Kontrollblöcken zu erreichen.

Daher stellt die Natural CICS-Schnittstelle das Lademodul NCIUEX1 als NATUEX1-Exit-Schnittstelle in einer CICS-Umgebung zur Verfügung. Dieses Modul stellt lediglich die Adressierbarkeit in CICS her und ruft den NCIUIDEX-Schnittstellen-Exit auf, indem es die Standard-Linkage-Konventionen (Register 13, 14, 15 und 1) verwendet, darüber hinaus aber auch CICS-bezogene Adressen in anderen Registern übergibt: R4 (EIB), R5 (EISTG), R6 (TCTTE).

Wenn Sie also Anfragen stellen wollen, die die Adressierbarkeit der CICS-Umgebung voraussetzen, sollten Sie die NCIUIDEX-Benutzerkennung-Exit-Schnittstelle und nicht die NATUEX1-Standard-Schnittstelle verwenden.

Das Quellcodemodul XNCIUIDX enthält ein Beispiel für einen Benutzerkennung-Exit.

Important: Bei jeder Installation eines neuen CICS-Releases muss die Benutzerkennung-Exit-Schnittstelle NCIUIDEX neu zusammengesetzt und verlinkt werden.

NCIXIDEX - Transaktionskennung-Exit-Schnittstelle

Standardmäßig verwendet Natural immer die Transaktionskennung, mit der die pseudokonversationelle Sitzung gestartet wurde. Diese Transaktionskennung kann in Natural mit CALLNAT CMTRNSET (Library SYSEXTP) geändert werden. Die Transaktionskennung-Exit-Schnittstelle NCIXIDEX kann ebenfalls verwendet werden, um die Kennung der pseudokonversationellen Natural-Transaktion zu ändern.

Der Aufruf der Transaktionskennung-Exit-Schnittstelle NCIXIDEX erfolgt über die Standard-Linkage-Konventionen (Register 13, 14, 15 und 1), aber zusätzlich über die Register 4 und 5, die CICS-EIB- und EISTG-Adressen enthalten, damit der Exit CICS-Dienste aufrufen kann.

Das Quellcode-Modul XNCIXIDX enthält ein Beispiel für einen Transaktionskennung-Exit.

Note: Die Transaktionskennung-Exit-Schnittstelle wird nur vor pseudokonversationellen Bildschirm-Ein-/Ausgaben unter der Kontrolle der Natural CICS-Schnittstelle aufgerufen,

d.h. der Exit wird nicht bei konversationellen Bildschirm-E/A (z.B. `SET CONTROL 'N'`) oder wenn Natural von einem Frontend-Programm über `EXEC CICS LINK` aufgerufen wird, aufgerufen.

Debugging-Möglichkeiten bei der Natural CICS-Schnittstelle

Die folgenden Themen werden behandelt:

- Verwendung des Parameters TPF
- Asynchrone Natural-Sitzungen verwenden

Verwendung des Parameters TPF

Der dynamische Parameter `TPF=(TPF1,TPF2,TPF3,TPF4,TPF5,TPF6,TPF7,TPF8)` kann für treiberspezifische Optionen durch positionsgebundene Angabe von `1` bei der entsprechenden Option gesetzt werden.

Option	Erläuterung
`TPF1`	Aufruf des Adabas-Linkage-Moduls über `EXEC CICS LINK` mit Adabas-Parameter in TWA und CICS COMMAREA statt über DCI. Ermöglicht das Debugging von Problemen im Zusammenhang mit Adabas über CEDF.
`TPF2`	Bei dieser Parametereinstellung werden alle `CMTRACE`-Trace-Sätze zusätzlich zum CICS-Trace (Meldungsnummer NCI0110) in die Natural CICS Interface Message Destination (siehe Schlüsselwort-Subparameter `MSGDEST` des Makros `NTCICSP`) geschrieben. **Note:** Zum Schreiben von Trace-Sätzen ist der Profilparameter `ETRACE=ON` oder `ETRACE=(ON,NOGTF)` erforderlich.
`TPF3`	Dump des gesamten Natural Buffer Pool. Mit dieser Parametereinstellung wird der gesamte Natural Buffer Pool in einen CICS-Transaktions-Dump aufgenommen. **Note:** Normalerweise wird der Natural Buffer Pool in einem Dump nicht benötigt, da alle für eine Sitzung relevanten Objekte aus dem Buffer Pool ohnehin ausgewertet werden. Daher kann diese Option nur im Falle eines Buffer Pool-Problems erforderlich sein.
`TPF4`	Dump des gesamten Editor Buffer Pool. Mit dieser Parametereinstellung wird der Editor Buffer Pool in einen CICS-Transaktions-Dump einbezogen.
`TPF6`	Behandlung von Terminal-E/A-Fehlern durch NCI. Mit dieser Parametereinstellung gibt NCI bei Terminal-Ein-/Ausgabe-Fehlern die Kontrolle nicht an Natural zurück, sondern behandelt sie selbst, was zu einer der Fehlermeldungen `NT06` bis `NT13` führt.

Option	Erläuterung
TPF7	Zwangsweiser Abbruch (Force Abend) im Falle von NCI-Systemfehlern. Mit dieser Parametereinstellung wird bei den Fehlermeldungen NS*xx*, NI*xx*, NR*xx* oder NUS*nnnn* ein Programm-Check erzwungen. Dies ist besonders hilfreich, wenn ein Debugging-Tool zum Abfangen von Abbrüchen aktiv ist. Dann kann der Fehler direkt online ausgewertet werden.

Bei der Angabe von 0 (die auch weggelassen werden kann) wird die entsprechende Option nicht gesetzt, z.B.:

TPF=(0,0,0,1) was gleichbedeutend mit TPF=(,,,1) ist.

Asynchrone Natural-Sitzungen verwenden

Wenn die ersten 5 Zeichen im dynamischen Parameterstring zum Starten von Natural ASYN, sind, baut die Natural CICS-Schnittstelle immer eine asynchrone Natural-Sitzung auf, unabhängig davon, ob eine Terminal- oder eine Nicht-Terminal-Sitzung gestartet wird.

Dies kann zu Testzwecken hilfreich sein, insbesondere mit EDF oder mit anderen installierten Debugging-Tools.

TWA-Nutzung bei der Natural CICS-Schnittstelle

Die Natural-Transaktionen sind alle mit einer TWA-Größe von 128 Byte definiert, obwohl die Natural CICS-Schnittstelle nur die ersten 88 Byte der CICS-Transaktions-Work-Area (TWA) für die Natural-Verarbeitung der folgenden Funktionen verwendet:

- Wenn Adabas wegen der Adabas-Parameterliste (bis zu 32 Byte) aufgerufen wird, speichert die Natural CICS-Schnittstelle den TWA-Inhalt vor dem Aufruf von Adabas und stellt ihn nach dem Adabas-Aufruf wieder her.
- Wenn externe Programme für die Adresszeiger der Parameterliste (bis zu 20 Byte, siehe Natural-Statement CALL) aufgerufen werden, speichert die Natural CICS-Schnittstelle den TWA-Inhalt vor dem Aufruf des externen Programms und stellt den TWA-Aufrufanteil nach dem Aufruf des externen Programms wieder her.
- Wenn ein Backend-Programm für die Beendigungsnachricht und eventuelle Beendigungsdaten (80 Bytes, siehe *Konventionen für den Aufruf von Backend-Programmen* in der Natural-*Operations*-Dokumentation) aufgerufen wird
- Bei der Rückgabe der Kontrolle an einen “LINK”-Front-End-Aufrufer für die Beendigungsmeldung und die eventuellen Beendigungsdaten am Ende der Sitzung und beim vollständigen Zurücksetzen des Beendigungsnachrichtenbereichs auf einen niedrigen Wert am Ende des Natural-Dialogschritts, d.h. 80 Bytes am Ende der Sitzung und des Dialogschritts.

- Für die Übergabe von LE-Informationen beim Start der CICS-Task (bis zu 88 Bytes, nur beim Start der Task).

Benutzerprogramme (Frontend-, Backend-, so genannte externe Programme) können ebenfalls die Vorteile der CICS TWA nutzen, um neben Natural zu kommunizieren, aber sie sollten nicht den von Natural verwendeten TWA-Teil verwenden. In solchen Fällen wird dringend empfohlen, die TWA-Größe der Natural-Transaktionen zu erhöhen und TWA-Teile außerhalb der ersten 128 Bytes zu verwenden.

5 Natural-CICS-Generierungsparameter

- NCISCPCB-Generierungsparameter 32
- NCMDIR-Makro-Parameter 32
- NCMTGD-Makro-Parameter 38
- NTCICSP-Makro-Parameter 44

In diesem Kapitel werden die Natural CICS-Generierungsparameter beschrieben. Folgende Themen werden behandelt:

Die zum Starten von Natural CICS verwendeten Parameter finden Sie im Abschnitt *CICS Startup Parameters* in *Installing Natural CICS Interface on z/OS* in der Natural *Installation*-Dokumentation.

Verwandte Themen:

- *Installing Natural CICS Interface on z/OS* in der Natural-*Installation*-Dokumentation.
- *SYSTP Utility* - ein Natural-Dienstprogramm, das verschiedene TP-Monitor-spezifische Funktionen bereitstellt.
- Informationen zum Betrieb und zu den einzelnen Komponenten von Natural in einer CICS-Umgebung finden Sie in den folgenden Kapiteln der vorliegenden *Natural unter CICS*-Dokumentation:

NCISCPCB-Generierungsparameter

Das Systemverzeichnis der Natural CICS-Schnittstelle wird durch Assemblieren und Verlinken des Quellcodemoduls `NCISCPCB` erzeugt; siehe den entsprechenden Schritt in *Installing Natural CICS Interface on z/OS* in der Natural-*Installation*-Dokumentation.

`NCISCPCB` enthält die folgenden Makros:

- `NCMDIR`
- `NCMTGD`

Der Zweck dieser Makros und die einzelnen Parameter, die in diesen Makros angegeben werden können, werden in den folgenden Abschnitten beschrieben.

NCMDIR-Makro-Parameter

Das Makro `NCMDIR` ist zwingend erforderlich und muss als erstes Makro im `NCISCPCB`-Quellcodemodul angegeben werden. Es enthält verschiedene Optionen für das System. Die einzelnen Parameter, die im `NCMDIR`-Makro angegeben werden können, werden im Folgenden beschrieben.

`CICSPLX` | `ROLLSRV` | `SIPSERV` | `SUBSID` | `TSKEY` | `TSRECSZ` | `USERS`

CICSPLX - Wechsel der CICS-Anwendungsregion

Mögliche Werte:

Wert	Erläuterung
YES	Die Natural CICS-Schnittstelle bewahrt alle sitzungsrelevanten Daten wie die Session Information Records (SIRs) und die Sitzungsdaten über eine pseudokonversationelle Bildschirm-Ein-/Ausgabe außerhalb einer lokalen CICS Application Owning Region (AOR) und ermöglicht so den Wechsel von CICS AORs. Wird dieser Parameter auf YES gesetzt, muss außerdem der Profilparameter ADAMODE auf einen Wert größer 0 gesetzt werden.
MODE	Diese Einstellung hat nahezu die gleiche Bedeutung wie YES. Die einzige Ausnahme ist, dass CICSPLX=MODE die Angabe des Profilparameters ADAMODE=0 erlaubt, d.h. ein CICS-AOR-Wechsel ist nicht möglich, aber eine Natural-Session kann den Neustart eines CICS-AOR in einer MRO-Umgebung überdauern.
NO	Kritische Natural-Sitzungsdaten werden im lokalen CICS-AOR aufbewahrt, wodurch der CICS-AOR-Wechsel faktisch deaktiviert wird. Dies ist der Standardwert.

Natural-PLEX-Unterstützung bedeutet, dass eine Natural CICS-Sitzung am Ende einer CICS-Task alle in einer CICS-Anwendungsregion von ihr hinterlassenen Spuren beseitigt, da sie möglicherweise nie wieder in diese Region zurückkehrt. Daher müssen alle für eine Natural CICS-Sitzung relevanten Daten außerhalb einer CICS-Anwendungsregion gehalten werden, d.h. Natural unter CICS übergibt am Ende der CICS-Task seine Sitzungsinformationsdatensätze (SIRs) an den SIP-Handler des Authorized Services Manager und die Sitzungsdaten an den Natural Roll Server. Darüber hinaus müssen alle "gehaltenen" Module, d.h. Module, die nicht mit Natural verlinkt sind, sondern direkt über Standard-Linkage-Konventionen wie RCA-Module oder das Adabas-Linkage-Modul aufgerufen werden, am CICS-Task-Ende freigegeben werden. Außerdem muss die Neustart-Information im Falle von COMARET=YES in einer CICS Terminal Owning Region (TOR) oder im Falle von COMARET=NO in einer CICS Data Owning Region (DOR), die von allen teilnehmenden CICS-AORs gemeinsam genutzt wird, aufbewahrt werden. Einzelheiten siehe Parameter COMARET.

Wurde YES oder MODE angegeben und der Parameter SUBSID im Makro NCMDIR nicht gesetzt wurde, wird der Wert des Natural-Profilparameters SUBSID genommen, der für die Natural-Sitzung gilt, die die NCI-Umgebung initialisiert.

Caution: Wird dieser Parameter auf YES oder MODE setzen, so werden automatisch die Parameter SIPSERV und ROLLSRV auf YES gesetzt.

ROLLSRV - Roll Server als Roll Facility

Mögliche Werte:

Wert	Erläuterung
NO	Dies ist der Standardwert, wenn CICSPLX=NO und SIPSERV=NO. Wenn CICSPLX oder SIPSERV YES ist, wird ROLLSRV=YES erzwungen.
YES	Die Angabe von YES bewirkt, dass die Natural CICS-Schnittstelle als Roll Facility nur den *Natural Roll Server* verwendet.

Wenn der Natural Roll Server zum Sichern und Wiederherstellen der Natural-Sitzungsdaten über eine Bildschirm-Ein-/Ausgabe hinweg verwendet werden soll, muss dieser Parameter auf YES gesetzt werden, wenn die Parameter CICSPLX und SIPSERV beide auf NO gesetzt sind. Wenn YES angegeben (oder erzwungen) wurde und der Parameter SUBSID im Makro NCMDIR nicht gesetzt wurde, wird der Wert des Natural-Profilparameters SUBSID übernommen, der für die die NCI-Umgebung initialisierende Natural-Sitzung gilt.

Beachten Sie, dass der Natural-Profilparameter SUBSID für die Zwecke der Natural CICS-Schnittstelle nur berücksichtigt wird, wenn er dynamisch oder im Natural-Parametermodul angegeben ist. Er wird ignoriert, wenn er in einem Parameterstring durch den Profilparameter SYS oder PROFILE oder in einem alternativen, mit dem Profilparameter PARM definierten Parametermodul angegeben ist.

Weitere Informationen zur Wahl zwischen Natural Roll Server und von der Natural CICS-Schnittstelle unterstützte Roll Files, siehe ***Wahl zwischen Roll Server und Roll Facilities*** und ***Auswahl der Roll Facility***.

SIPSERV - Session Information Pool des Authorized Services Manager

Mögliche Werte:

Wert	Erläuterung
NO	Bei CICSPLX=NO ist dies der Standardwert. Wenn CICSPLX nicht auf NO gesetzt ist, wird SIPSERV=YES erzwungen.
YES	Bewirkt, dass die Natural CICS-Schnittstelle ihre Session Information Records (SIRs) im Session Information Pool des *Authorized Services Manager* speichert.

Wenn dieser Parameter auf YES gesetzt ist oder wenn YES erzwungen wird, werden die Natural-Session-Information-Records außerhalb einer CICS-Region aufbewahrt, so dass Natural eine CICS-Anwendungsregion nach einer pseudo-konversationellen Bildschirm-Ein-/Ausgabe wechseln kann.

Wenn YES angegeben (oder erzwungen) wird und der Parameter SUBSID im Makro NCMDIR nicht gesetzt wurde, wird der Wert des Natural-Profilparameters SUBSID verwendet, der für die die NCI-Umgebung initialisierende Natural-Sitzung gilt.

Beachten Sie, dass für die Zwecke der Natural CICS-Schnittstelle der Natural-Profilparameter SUBSID nur dann berücksichtigt wird, wenn er dynamisch oder im Parametermodul angegeben ist. Er wird ignoriert, wenn er in einem Parameterstring durch den Profilparameter SYS oder PROFILE oder in einem (mit dem Profilparameter PARM definierten) alternativen Parametermodul angegeben ist.

Caution: Wenn YES für diesen Parameter wirksam ist, wird der ROLLSRV-Parameter zwangsweise auf YES gesetzt, sofern er nicht bereits entsprechend gesetzt ist.

SUBSID - Subsystemkennung

Mögliche Werte:

Wert	Erläuterung
xxxx	Definiert die Subsystemkennung für den Natural Roll Server und/oder für den Authorized Services Manager.

Dieser Parameter definiert die Natural-Subsystemkennung, die für den Natural Roll Server und/oder für den Authorized Services Manager verwendet werden soll. Wird dieser Parameter nicht angegeben, so wird der Wert des Natural-Profilparameters SUBSID genommen.

Beachten Sie, dass für die Zwecke der Natural CICS-Schnittstelle der Natural-Profilparameter SUBSID nur dann berücksichtigt wird, wenn er dynamisch oder im Parametermodul angegeben ist. Er wird ignoriert, wenn er in einem Parameterstring durch den Profilparameter SYS oder PROFILE oder in einem (mit dem Profilparameter PARM definierten) alternativen Parametermodul angegeben ist.

TSKEY - Präfixe für Natural CICS-Zwischenspeicherschlüssel

Dieser Parameter definiert die konstanten Präfixe der Zwischenspeicherwarteschlangen (Temporary Storage Queues). Erläuterung siehe unten.

Dieser Parameter hat die gleiche Bedeutung wie der Parameter TSKEY im Modul NCIZNEP (siehe Natural-*Installation*-Dokumentation) und muss identisch angegeben werden.

Mögliche Werte:

Wert	Erläuterung
(*xxxx*,*yyyy*)	*xxxx* definiert das Präfix für Roll-Daten, während *yyyy* das Präfix für pseudo-konversationelle Restart-Daten definiert.
(NAT2,NCOM)	Dies ist der Standardwert.

Wenn der CICS-Zwischenspeicher (Haupt- oder Hilfsspeicher) für die Roll Facility der Natural-CICS-Schnittstelle oder für den Kommunikationsbereich für pseudokonversationelle Natural-Tasks (wie beim Parameter COMARET des Makros NTCICSP beschrieben) verwendet werden soll, müssen Namen für Warteschlangen des Task-abhängigen spezifischen Zwischenspeichers angegeben werden.

Diese Warteschlangennamen bestehen aus einem konstanten 4-Byte-Schlüssel und einem Task-bezogenen Schlüssel. Bei terminalabhängigen Tasks entspricht dieser Task-bezogene Schlüssel der Terminalkennung, bei asynchronen Nicht-Terminal-Tasks entspricht er der eindeutigen CICS-Task-Nummer (CICS Unique Task Number). Das konstante Präfix der temporären Speicherwarteschlangennamen wird durch den Parameter TSKEY definiert.

Die Natural CICS-Schnittstelle erfordert zwei 4-Byte-Präfixe: eines für Ein-/Auspeicherungsdaten (Roll Data) und eines für pseudo-konversationelle Neustart-Daten (Restart Data). *xxxx* definiert das Präfix für Ein-/Auspeicherungsdaten, *yyyy* definiert das Präfix für pseudo-konversationelle Restart-Daten. Die beiden Präfixe müssen sich voneinander unterscheiden und für Natural unter CICS exklusiv sein.

Wenn Sie in einer CICSplex-Umgebung arbeiten, muss das CICS-Präfix für den temporären Speicher für Natural-Sitzungsneustartinformationen in einem CICS-TST als REMOTE/SHARED definiert werden, um in allen beteiligten CICS-Regionen zugänglich zu sein.

TSRECSZ - Satzgrößen für Haupt- und Hilfszwischenspeicher

Dieser Parameter legt die maximale Satzlänge für das Aus-/Einpeichern (Rolling) von Daten fest, wenn der CICS-Zwischenspeicher als Natural CICS Interface Roll Facility verwendet werden soll.

Mögliche Werte:

Wert	Erläuterung
(*nnnnn*,*mmmmm*)	Der erste Subparameter *nnnnn* gilt für den CICS-Hauptzwischenspeicher und muss im Bereich von 4096 bis 32763 oder 0 liegen oder eines der Schlüsselwörter MAX, YES oder NO sein: ■ Wenn der Wert numerisch und nicht Null ist, wird er ohne Bedingungen verwendet. ■ Wenn der Wert auf 0 oder NO gesetzt ist, kann der CICS-Hauptzwischenspeicher nicht für eine Natural Roll Facility verwendet werden. ■ Wenn der Wert auf MAX oder YES gesetzt ist, wird eine Datensatzgröße von 32763 verwendet.

Wert	Erläuterung
	Der zweite Subparameter *mmmmm* gilt für den CICS-Hilfszwischenspeicher und muss im Bereich von `3976` bis `32763` oder `0` liegen oder eines der Schlüsselwörter `MAX`, `YES` oder `NO` sein: ■ Wenn der Wert numerisch und nicht Null ist, wird er ohne Bedingungen verwendet. Wenn er auf `MAX` gesetzt ist, wird eine Datensatzgröße von `32763` genommen. ■ Wenn er auf `NO` gesetzt ist, kann der CICS-Zwischenspeicher nicht für eine Natural Roll Facility verwendet werden. ■ Wenn dieser Wert auf `0` oder `YES` gesetzt ist, legt die Natural CICS-Schnittstelle die Satzlänge fest, die in ein Kontrollintervall des Zwischenspeichers passt, d.h. CI-Größe minus VSAM-Kontrollinformationen minus CICS-Kontrollinformationen. Eine benutzerdefinierte Satzgröße, die größer ist als die CI-Größe, führt zu weniger (logischen) Ein-/Ausspeicherungs-Operationen (Roll In/Out) auf Kosten des zusätzlichen CICS-Aufwands durch das Schreiben von übergreifenden Sätzen.
`(32748,0)`	Dies ist der Standardwert.

USERS - Anzahl der Session Information Record Slots

Dieser Parameter gibt die Anzahl der Session Information Record Slots (SIRs) an.

Mögliche Werte:

Wert	Erläuterung
`(`*nnnnn*`,` *mmm*`)`	Der Subparameter *nnnnn* definiert die Anzahl der SIRs, die im Natural CICS-Verzeichnismodul selbst gehalten werden sollen. *nnnnn* muss im Bereich von `1` bis `32767` liegen. Wenn die SIR-Slots im Verzeichnis belegt sind, belegt die Natural CICS-Schnittstelle ein CICS-Shared-Storage-Segment, das groß genug ist, um die durch *mmm* definierte Anzahl von SIRs aufzunehmen, die im Bereich von `0` bis `255` liegen muss. Wenn der Subparameter *mmm* `0` ist oder weggelassen wird, belegt das System keinen zusätzlichen Speicher für SIRs, wenn kein freier SIR-Slot im Systemverzeichnis verfügbar ist. In diesem Fall ist das Natural CICS-System tatsächlich auf die durch den ersten Subparameter angegebene Anzahl von Benutzern beschränkt. Wenn für *mmm* ein anderer Wert als `0` angegeben wird, werden sekundäre Speichersegmente automatisch nach Bedarf zugewiesen. Zugewiesene Sekundärsegmente werden wieder freigegeben, wenn sie nicht mehr benötigt werden.
`(100,20)`	Dies ist der Standardwert.

Die Natural CICS-Schnittstelle speichert permanent Informationen über alle aktiven Natural-Sitzungen. Für jede Sitzung wird ein so genannter Session Information Record (SIR) geführt.

Diese SIRs werden aufbewahrt

■ in einer Coupling Facility, wenn sie in einer z/OS Parallel Sysplex-Umgebung laufen;

- in einem Datenbereich des Natural Authorized Services Manager, wenn er in mehreren CICS-Regionen innerhalb eines einzelnen z/OS-Systems läuft;
- im Hauptspeicher einer CICS-Region, wenn sie in einem einzelnen CICS-AOR (lokal) ausgeführt wird.

Wenn jedoch eine Natural-Sitzung in einer CICS-Region aktiv ist, belegt sie einen SIR-Slot in der aktuellen Anwendungsregion.

Bei der lokalen Ausführung in einem einzelnen CICS-AOR gilt der Parameter `USERS` für alle Natural-Sitzungen. Bei der Ausführung in einer CICSplex-Umgebung gilt `USERS` für die Teilmenge der Natural-Sitzungen, die derzeit in jedem der beteiligten CICS-AORs aktiv ist.

NCMTGD-Makro-Parameter

Das Makro `NCMTGD` ist zwingend erforderlich und muss für jede Thread-Gruppe angegeben werden. Bei der Natural CICS-Schnittstelle können Sie Gruppen von Threads definieren. Diese Gruppen werden bei der Sitzungsinitialisierung durch die CICS-Transaktionskennung gesteuert/ausgewählt. Die gemeinsame Thread-Größe für die verschiedenen Gruppen kann unterschiedlich sein und die Gruppen können unterschiedliche Optionen haben. Die Thread-Gruppendefinitionen sind Teil des Natural CICS-Systemverzeichnisses, da sie für das gesamte System relevant sind, nicht nur für eine einzelne Sitzung.

Die einzelnen Parameter, die in einem `NCMTGD`-Makro angegeben werden können, werden im Folgenden beschrieben.

`PFKEY` | `PRIMERF` | `THRDSZE` | `THREADS` | `TRAN` | `TYPE` | `XTRAN`

PFKEY - PF-/PA-Tasten für Thread-Gruppe

Dieser Parameter definiert eine einzelne CICS-Transaktion oder eine Liste von Transaktionen.

Mögliche Werte:

Wert	Erläuterung
xxx	Mögliche Werte für *xxx* sind: PF1 bis PF24, PA1 bis PA3.
(xxx,xxx,...)	Es kann auch eine Liste von Tasten angegeben werden. Diese muss in Klammern gesetzt werden, zum Beispiel `PFKEY=(PF12, PF14)`. Ein Standardwert ist nicht vorgesehen.

Ein Standardwert ist nicht vorgesehen.

Beim Starten einer Sitzung durchsucht die Natural CICS-Schnittstelle alle Thread-Gruppendefinitionen nach der aktuellen Transaktionskennung bzw. der PF- oder PA-Taste. Wenn sie nicht gefunden wird, wird die erste Thread-Gruppe als Standardwert verwendet.

Caution: Für alle Gruppen mit Ausnahme der ersten Gruppe, die als Standardgruppe verwendet wird, muss mindestens eine Transaktionskennung (im Zeichen- oder Hexadezimalformat) oder eine transaktionseinleitende Achtungskennung (Attention Identifier) angegeben werden.

PRIMERF - Primäre Roll Facility für die Natural CICS-Schnittstelle

Dieser Parameter definiert die primäre Roll Facility für die Natural CICS-Schnittstelle für alle in der zugehörigen Thread-Gruppe definierten Tasks. Daher gilt dieser Parameter nicht für Thread-Gruppen mit `TYPE=NONE`.

Mögliche Werte:

Wert	Erläuterung
`VSAM`	Als primäre Roll Facility werden VSAM Natural CICS Interface VSAM RRDS Roll Files verwendet. Wenn im CICS-System kein VSAM RRDS Roll File vorhanden ist, wird `PRIMERF=AUX` wirksam. Wenn die VSAM RRDS Roll Files voll werden oder nicht verfügbar sind, gilt Folgendes: ■ `PRIMERF=AUX` wird wirksam, wenn im CICS-System ein Hilfszwischenspeicher definiert ist. ■ `PRIMERF=MAIN` wird wirksam, wenn im CICS-System kein Hilfszwischenspeicher definiert ist.
`AUX`	CICS-Hilfszwischenspeicher wird als primäre Roll Facility der Natural CICS-Schnittstelle genommen. Falls im CICS-System kein Hilfszwischenspeicher definiert ist, wird die Einstellung `PRIMERF=MAIN` wirksam.
`MAIN`	CICS-Hauptzwischenspeicher wird als primäre Roll Facility der Natural CICS-Schnittstelle verwendet. Die Satzgröße wird durch den Parameter `TSRECSZ` definiert. Dabei gilt Folgendes: ■ CICS-Hauptzwischenspeicher ist Speicher, der oberhalb der Grenze zugewiesen wird. Dadurch werden VSAM-Ein-/Ausgabe-Aktivitäten und die Kommunikation mit einem Zwischenspeicher-Server vermieden. Infolgedessen ist `PRIMERF=MAIN` wesentlich schneller als `PRIMERF=AUX` und wird daher empfohlen, wenn für Natural-Sitzungen kein dynamisches Weiterleiten von Transaktionen (Transaction Routing) verwendet wird. Allerdings werden die Aus-/Einspeicherungsdaten (Roll Data) in Sätze mit einer maximalen Größe von 32 KB aufgeteilt. ■ Die Natural CICS-Schnittstelle kann auch CICS-Speicherobjekte als Roll Facility verwenden. Speicherobjekte werden oberhalb der Grenze zugeordnet. Im Gegensatz zu `PRIMERF=MAIN` müssen die Aus-/Einspeicherungsdaten jedoch nicht aufgeteilt werden, wenn sie in ein Speicherobjekt verschoben werden. ■ Wenn Sie Speicherobjekte verwenden möchten, geben Sie `PRIMERF=MAIN` und `MEMOBJR=ON` (Standardeinstellung) mit dem Makro `NTCICSP` an (siehe Parameter-Referenz-Dokumentation).

Wert	Erläuterung
NONE	Die zugehörigen Sitzungen benutzen überhaupt keine Aus-/Einspeicherungsfunktion. NONE ist nicht gültig für TYPE=SHR-Gruppen und für Gruppen mit TYPE=ALIAS, die TYPE=SHR-Gruppen neu definieren. Sitzungen, die mit Thread-Gruppen verbunden sind, die mit PRIMERF=NONE definiert sind, können nicht verschoben werden, da es keine Aus-/Einspeicherungsfunktion gibt, um diese Aufgabe auszuführen. Diese Sitzungen sind daher konversationell.

Ein Standardwert ist nicht vorgesehen.

Dieser Parameter wird ignoriert, wenn Sie den Natural Roll Server verwenden. Wenn Sie eine Natural-Sitzung mit Roll Server zwangsweise konversationell ohne Aus-/Einspeicherungsfunktion laufen lassen, ist der Wert NONE wirksam.

Weitere Informationen zur Wahl zwischen Natural Roll Server und von der Natural CICS-Schnittstelle unterstützte Roll Files, siehe ***Wahl zwischen Roll Server und Roll Facilities*** und ***Auswahl der Roll Facility***.

THRDSZE - Thread-Größe

Dieser Parameter definiert die gemeinsame Thread-Größe für die Gruppen mit TYPE=GETM und TYPE=SHR.

Mögliche Werte:

Wert	Erläuterung
nnnnn	Die Thread-Größe *nnnnn* kann im Bereich von 40 KB (Minimum) bis 65532 KB (Maximum) liegen.

Ein Standardwert ist nicht vorgesehen.

Beachten Sie, dass dieser Parameter die *logische* Thread-Größe definiert, die Natural zur Verfügung steht. Die Natural CICS-Schnittstelle fügt jedoch für interne Administrationszwecke weitere 2 KB zur logischen Thread-Größe hinzu. Das bedeutet, dass die *physische* Thread-Größe oder Länge der Thread-GETMAIN-Anforderung um 2 KB größer ist als der THRDSZE-Wert.

Bei TYPE=GETM sind zusätzlich 16 Bytes für die vor- und nachgelagerten CICS-Speicherabrechnungsbereiche (CICS Storage Accounting Areas, SAAs) zu berücksichtigen.

Wichtige Hinweise:

1. Bei GETMAINs mit mehr als 512 KB richtet CICS diese Speicherbereiche an MB-Grenzen aus.
2. Bei Verwendung von Transaction Isolation verwendet CICS intern 1 MB “Pages” in der EUDSA (Details finden Sie im *CICS Performance Guide*).

Diese beiden Tatsachen führen zu Speicherfragmentierung und sollten bei der Einstellung eines geeigneten EDSALIM in CICS berücksichtigt werden.

THREADS - Anzahl der Threads oder Tasks pro Thread-Gruppe

Dieser Parameter gibt die Anzahl der Threads oder Tasks wie unten beschrieben an.

Mögliche Werte:

Wert	Erläuterung
nnn	Die Anzahl der Threads kann gleich oder kleiner als 510 sein.

Ein Standardwert ist nicht vorgesehen.

Für Thread-Gruppen mit TYPE=SHR ist der Parameter THREADS obligatorisch und definiert die Anzahl der Threads, die während der Installation über GETMAIN (SVC oder SHARED, je nach CICS-Version) zugewiesen werden sollen.

Für die Thread-Gruppen mit TYPE=GETM und TYPE=NONE ist der Parameter THREADS optional und bestimmt die maximale Anzahl der gleichzeitig aktiven Natural-Tasks pro Thread-Gruppe. Für diese Thread-Gruppen-Typen steuert der THREADS-Parameter im Gegensatz zu Thread-Gruppen mit TYPE=SHR nicht die Speichernutzung (siehe auch ***Kontrolle über die Speichernutzung***).

Die Anzahl der Threads bzw. die Anzahl der Tasks pro Thread-Gruppe wird durch die Bereitstellung von Thread Control Blocks (TCBs) definiert.

Bei Thread-Gruppen mit TYPE=SHR ist jeder Thread eng mit seinem TCB verbunden. Die Threads werden gemeinsam genutzt, indem sie im zugehörigen TCB in eine Warteschlange gestellt werden. Thread-Gruppen mit TYPE=GETM und TYPE=NONE stellen sich nur bei einem TCB in die Warteschlange, um aktiv zu werden.

Während Sitzungen mit Thread-Gruppen mit TYPE=SHR miteinander um Threads konkurrieren, konkurrieren die anderen Sitzungstypen um TCBs mit einem bereits zugewiesenen Thread (TYPE=GETM) oder mit gar keinem zugewiesenen Thread (TYPE=NONE).

Wenn der Parameter THREADS ungleich Null ist, werden die Natural-Profilparameter DBROLL und MAXROLL sowie die Aufrufe von CMROLL bei TYPE=GETM/NONE-Thread-Gruppen anders behandelt: Da Threads nicht freigegeben werden können, wird die gehaltene TCB-Ressource freigegeben, wodurch die Sitzung mit den im Speicher gehaltenen Sitzungsdaten aktiviert wird.

TRAN - Transaktionskennungen für Thread-Gruppe

Dieser Parameter definiert eine einzelne CICS-Transaktion oder eine Liste mit Transaktionen.

Mögliche Werte:

Wert	Erläuterung
(siehe unten)	Ein oder mehrere CICS-Transaktionscodes, die im PCT für Natural definiert sind.

Ein Standardwert ist nicht vorgesehen.

Der Parameter `TRAN` erwartet Transaktionskennungen im Zeichenformat. Transaktionskennungen mit nicht alphanumerischen Zeichen müssen in Hochkommata eingeschlossen werden.

Beim Starten einer Sitzung durchsucht die Natural CICS-Schnittstelle alle Thread-Gruppendefinitionen nach der aktuellen Transaktionskennung bzw. der PF- oder PA-Taste. Wenn sie nicht gefunden wird, wird standardmäßig die erste Thread-Gruppe verwendet.

Eine Liste mit Transaktionskennungen muss in Hochkommata eingeschlossen werden, zum Beispiel `TRAN=(NATU, XYZ)`.

Caution: Für alle Gruppen mit Ausnahme der ersten Gruppe, die als Standardgruppe verwendet wird, muss mindestens eine Transaktionskennung (im Zeichen- oder Hexadezimalformat) oder eine transaktionseinleitende Achtungskennung angegeben werden.

TYPE - Thread-Typ für Gruppe

Dieser Parameter legt fest, welcher Thread-Typ für eine bestimmte Gruppe verwendet werden soll.

Mögliche Werte:

Wert	Erläuterung
`SHR`	Es werden CICS Shared Storage Threads verwendet. Die für eine Thread-Gruppe verfügbaren Threads werden von allen für diese Gruppe definierten CICS-Transaktionen gemeinsam genutzt. Die Thread-Auswahl beim Start einer CICS-Task erfolgt mittels einer ENQUEUE/DEQUEUE-Technik. Wenn derzeit kein Thread verfügbar ist, wird eine Warteschlange für diese Thread-Gruppe geführt. Dies ist der Standardwert. In einer z/OS-Parallel-Sysplex-Umgebung erzwingt der Natural-Profilparameter `RELO=OFF`, dass Sitzungen mit `TYPE=SHR`-Threads konversationell sind, um einen CICS-Regionswechsel zu verhindern.

Wert	Erläuterung
GETM	Es werden über GETMAIN zugewiesene Threads verwendet, was bedeutet, dass ein Thread tatsächlich aufgenommen wird, indem eine CICS-GETMAIN-Operation - EXEC CICS GETMAIN FLENGTH - mit der gemeinsamen Thread-Größe der Thread-Gruppe ausgeführt wird. Bei der Verwendung von Threads, die über GETMAIN zugewiesen werden, steht jeder Natural-Task bis zu ihrer Beendigung exklusiver Thread-Speicher zur Verfügung, d.h. für pseudo-konversationelle Tasks von einer Bildschirm-Ein-/Ausgabe zur nächsten. Wenn der Natural-Parameter RELO=OFF oder PSEUDO=OFF angegeben wird, sind Tasks, die über GETMAIN zugewiesene Threads verwenden, gezwungen, konversationell zu sein, da nicht gewährleistet ist, dass nach einem FREEMAIN des Threads ein nachfolgendes GETMAIN denselben Speicherplatz erhält. Da der über GETMAIN zugewiesene Thread-Speicher jedoch ausschließlich der besitzenden Task gehört, können solche Tasks als nicht per Aus-/Einspeicherung (Rolling) verlagerbar definiert werden (siehe den Parameter PRIMERF), was bedeutet, dass ein bestimmter Thread bis zum Ende der Natural-Sitzung zu einer bestimmten Task gehört. In diesem Fall ist die Task von vornherein konversationell und es findet keine Aus-/Einspeicherung statt.
NONE	Die in dieser Thread-Gruppe definierten Transaktionen verwenden keine Threads, und alle Natural-GETMAIN-Anfragen werden direkt an CICS für eine EXEC CICS-GETMAIN-FLENGTH-Anfrage weitergeleitet. Konzeptbedingt können derartige Aufgaben nicht per Aus-/Einspeicherung (Rolling) verlagert werden und sind daher konversationell.
ALIAS	Das aktuelle NCMTGD-Makro bietet verschiedene Optionen für die durch die vorherige NCMTGD-Makrospezifikation definierte Thread-Gruppe. Allerdings können nur Thread-Gruppen mit TYPE=GETM und TYPE=SHR durch eine oder mehrere NCMTGD-Makro-Angaben mit TYPE=ALIAS als Anfragen neu definiert werden. Es werden bis zu 99 Thread-Gruppen unterstützt, d.h. es werden bis zu 99 NCMTGD-Makro-Angaben mit einem anderen TYPE als ALIAS erkannt.

XTRAN - Hexadezimale Transaktionskennungen für Thread-Gruppe

Dieser Parameter entspricht dem TRAN-Parameter, erwartet aber eine Transaktionskennung im hexadezimalen Format.

Mögliche Werte:

Wert	Erläuterung
(siehe unten)	Mögliche Werte sind: Ein oder mehrere CICS-Transaktionscodes, die in der PCT für Natural definiert sind.

Ein Standardwert ist nicht vorgesehen.

Eine Liste von Transaktionskennungen im hexadezimalen Format muss in runde Klammern gesetzt werden, zum Beispiel: XTRAN=(D5C1E3E4, E7E8E9).

Caution: Für alle Gruppen mit Ausnahme der ersten Gruppe, die als Standardgruppe verwendet wird, muss mindestens eine Transaktionskennung (im Zeichen- oder Hexadezimalformat) oder eine transaktionseinleitende Achtungskennung (Attention Identifier) angegeben werden.

NTCICSP-Makro-Parameter

Die für die Natural-CICS-Schnittstelle erforderlichen Parameter werden durch Assemblierung des Natural-Parametermoduls erzeugt, das die erforderlichen `NTCICSP`-Makro-Definitionen enthält. Das Natural-Parametermodul wird in dem entsprechenden Installationsschritt in *Installing Natural CICS Interface on z/OS* (in der Natural *Installation*-Dokumentation) erstellt.

Das `NTCICSP`-Makro bestimmt alle Natural-Sitzungsoptionen, die in einer CICS-Umgebung relevant sind. Die einzelnen Parameter, die im Makro `NTCICSP` enthalten sind, werden unter *CICSP - Umgebungsparameter für Natural CICS-Schnittstelle* in der *Parameter-Referenz*-Dokumentation beschrieben.

6 Anpassung von VSAM RRDS Roll Files

- Anzahl der VSAM RRDS Roll Files erhöhen 46
- Anzahl der VSAM RRDS Roll Files verringern 46
- Merkmale der VSAM RRDS Roll Files ändern 47

Dieses Kapitel beschreibt die Anpassung von VSAM RRDS Roll Files.

Dieses Kapitel gilt nicht, wenn Sie den Natural Roll Server verwenden.

Anzahl der VSAM RRDS Roll Files erhöhen

Es können bis zu neun VSAM RRDS Roll Files zugewiesen werden. Jedes Roll File hat eine Kennung, die aus einem benutzerdefinierten Präfix, gefolgt von einem festen Suffix, besteht. Das Präfix kann 1 bis 9 Zeichen lang sein. Das Suffix besteht aus zwei Zeichen von `R1` bis `R9`.

Um ein neues VSAM Roll File hinzuzufügen, müssen Sie die folgenden Schritte ausführen:

1. Erstellen Sie einen leeren VSAM RRDS, der den Standards Ihres Standorts entspricht. Initialisieren Sie dann den Dataset mit dem Batch-Programm `NCISCPRI`, das während der Natural-Installation assembliert worden sein muss. Die Attribute `SPACE` und `RECORDSIZE` können sich zwischen verschiedenen Roll Files unterscheiden, so dass Sie sie nach Bedarf ändern können, um die besten Werte für Ihre Umgebung zu finden.
2. Legen Sie einen FCT-Eintrag an und ändern Sie die CICS JCL entsprechend, indem Sie das Präfix/Suffix für beide verwenden.

Das neue Roll File wird verfügbar, wenn die Natural CICS-Schnittstelle erneut initialisiert wird.

Anzahl der VSAM RRDS Roll Files verringern

Führen Sie die folgenden Schritte aus:

1. Vergewissern Sie sich, dass Natural nicht aktiv ist.
2. Löschen Sie entweder die FCT- und JCL-Definitionen oder löschen Sie die Datei.

Die Anzahl der Roll Files wird angepasst, wenn die Natural CICS-Schnittstelle erneut initialisiert wird.

Merkmale der VSAM RRDS Roll Files ändern

Führen Sie die folgenden Schritte aus:

1. Befolgen Sie die oben beschriebenen Vorgehensweise zum Verringern der Anzahl der Roll Files.
2. Führen Sie die Prozeduren zum Erhöhen der Anzahl der Roll Files aus.

7 Natural in CICS MRO-Umgebungen

- Subparameter COMARET im Makro NTCICSP auf ON gesetzt 50
- Subparameter COMARET im Makro NTCICSP auf OFF gesetzt 50

Dieses Kapitel beschreibt die Funktionalität von Natural in CICS Multi-Region (MRO)-Umgebungen.

Subparameter COMARET im Makro NTCICSP auf ON gesetzt

Wenn der Parameter `COMARET` im Makro `NTCICSP` auf `ON` gesetzt ist, werden die Natural-Sitzungsdaten in zwei verschiedenen CICS-Regionen gehalten:

- Die Informationen zum Sitzungsneustart werden in der COMMAREA gehalten, die mit dem Terminaleintrag in der CICS-Terminal Owning Region (TOR) verknüpft ist.
- Die eigentlichen Sitzungsdaten werden in der CICS Application Owning Region (AOR) gehalten, d.h. im Thread oder in der Roll Facility.

Dies kann zu Inkonsistenzen führen, wenn z.B. die AOR neu gestartet wird, die TOR aber noch alte "wartende" Natural-Sitzungen enthält. Die Wiederaufnahme einer solchen Sitzung führt zu einer entsprechenden Fehlermeldung.

Subparameter COMARET im Makro NTCICSP auf OFF gesetzt

Wenn der Parameter `COMARET` auf `OFF` gesetzt ist, bleiben alle Natural-Sitzungsdaten im AOR erhalten, wodurch die oben erwähnten Inkonsistenzen vermieden werden.

Es kann jedoch ein Sicherheitsproblem auftreten, wenn ein Terminal aus der TOR entfernt wird (entweder zurück zu VTAM oder durch Ausschalten des Session Managers oder der Stromversorgung) und ein anderes Terminal, das diese TOR anwählt, die Kennung des entfernten Terminals erhält und den Natural-Transaktionscode einstellt: Dann nimmt dieses Terminal die Sitzung des zuvor entfernten Terminals wieder auf, weil die Neustartinformationen im temporären Speicher des AOR die Terminalkennung als Teil des Warteschlangennamens enthalten.

Um eine solche Situation zu verhindern, kann ein Knotenfehlerprogramm (Node Error Program, NEP) installiert werden (siehe ***CICS-Knotenfehlerprogramm und Freigabe aller Ressourcen bei Zeitüberschreitung einer Natural-Sitzung*** und ***Natural CICS-Beispielprogramme***), das eine Natural-Sitzung beendet, wenn das zugehörige Terminal getrennt wird.

8 CICS-Knotenfehlerprogramm und Freigabe aller Ressourcen bei Zeitüberschreitung einer Natural-Sitzung

- Sitzungsbeendigung erfolgt nicht unter Kontrolle der Natural CICS-Schnittstelle 52
- Wiederherstellungsmechanismen 53
- NCIZNEP einrichten 54
- NCIZNEP-Funktionalität 54
- Behandlung von Zeitüberschreitungen (Timeouts) 56
- API USR4006N zum Abrufen oder Abbrechen von Sitzungen 57

Eine aktive Natural-Sitzung nutzt CICS-Ressourcen wie Thread-Speicher, Roll Facility-Einträge (d.h. Datensätze in einem VSAM RRDS File, in einer CICS-Zwischenspeicherwarteschlange oder im Hauptspeicher) sowie Roll Server Slots. Diese Ressourcen werden von der Natural CICS-Schnittstelle zugeordnet und korrekt freigegeben, sobald eine von der Natural CICS-Schnittstelle kontrollierte Sitzung normal oder abnormal beendet wird.

Falls eine Natural-Sitzung nicht von der Natural CICS-Schnittstelle kontrolliert wird, hilft ein Knotenfehlerprogramm (Node Error Program, NEP) bei der korrekten Freigabe der belegten Ressourcen.

In diesem Kapitel werden das CICS-Knotenfehlerprogramm und Aspekte von Zeitüberschreitungen (Timeouts) behandelt.

Informationen zur Installation eines CICS-Knotenfehlerprogramms (Node Error Program) finden Sie im entsprechenden Abschnitt unter *Installing Natural CICS Interface on z/OS* in der Natural *Installation*-Dokumentation.

Sitzungsbeendigung erfolgt nicht unter Kontrolle der Natural CICS-Schnittstelle

In den folgenden Situationen erhält die Natural CICS-Schnittstelle beim Beenden der Sitzung keine Kontrolle und kann daher die von der Sitzung noch belegt gehaltenen Ressourcen nicht ordnungsgemäß freigeben:

1. Ein Nicht-Natural-Programm, das von Natural aufgerufen wird, gibt ein -Kommando aus, um die CICS Task abnormal zu beenden.
2. Einige CICS-Monitoring-Produkte bieten Tools zum Bereinigen von CICS Tasks an, so dass ein von der Anwendung gesetzter Exit zur abnormalen Beendigung umgangen werden kann.
3. Ein Benutzer trennt ein Terminal von der CICS Region (durch Ausschalten des Geräts oder Verwendung einer geeigneten Session Manager-Funktion), während die entsprechende Natural-Sitzung in CICS zu diesem Zeitpunkt nicht aktiv ist (pseudo-konversationelle Bildschirm-Ein-/Ausgabe).
4. Ein Benutzer war in der Natural-Terminalsitzung für den Zeitraum inaktiv, der mit dem `TIMEOUT`-Parameter im CICS-Segment des RACF-Benutzerprofils angegeben wurde.

Wiederherstellungsmechanismen

Die Natural CICS-Schnittstelle bietet einige Wiederherstellungsmechanismen zur Rückerholung aus solchen Situationen zu erholen.

Beispiel:

Immer wenn eine neue Natural-Sitzung gestartet werden soll, prüft Natural, ob eine andere Sitzung mit der gleichen Terminalkennung aktiv ist. Wenn eine solche Sitzung existiert, wird sie logisch beendet, und alle ihre Ressourcen werden freigegeben, bevor die neue Sitzung gestartet wird.

Wenn jedoch `COMARET=OFF` im Makro `NTCICSP` gesetzt ist, werden die Informationen zur Wiederaufnahme einer Natural-Sitzung in einem CICS-Zwischenspeicherdatensatz gespeichert, in dem die Terminalkennung Teil des Namens der Speicherwarteschlange ist. Dies hat zur Folge, dass ein anderer CICS-Benutzer, der versucht, eine Natural-Sitzung mit dieser Terminalkennung zu starten, die alte Sitzung fortsetzt, anstatt eine neue Sitzung zu starten. Es wird daher dringend empfohlen, `COMARET=ON` zu setzen.

CICS bietet die NEP-Exit-Schnittstelle (NEP = Node Error Program) `DFHZNEP`, die aufgerufen wird, wenn ein Benutzer ein Terminal aus der CICS-Region abmeldet. Die Sitzung geht verloren, wenn der Benutzer die Terminalverbindung trennt, während die entsprechende Natural-Sitzung in CICS nicht aktiv ist. Sie können dann mit `DFHZNEP` das Natural-spezifische Knotenfehlerprogramm `NCIZNEP` ausführen und die verlorene Sitzung beenden. Dieses Programm wird mit der Natural CICS-Schnittstelle ausgeliefert (siehe auch ***Natural CICS-Beispielprogramme***).

Wird eine aktive und laufende Natural-Sitzung abgebrochen

- mit `CEMT SET TASK(...) FORCEPURGE`,
- mit `CEKL SET TASK(...) FORCEPURGE`,
- durch ein CICS-Monitoring-Produkt,
- durch Beendigung der Natural-Sitzung mittels der Funktion **Flush Session** der SYSTP Utility,

dann wird die entsprechende CICS-Task sofort mit CICS Abend AKEH beendet, ohne dass `NCIZNEP` zur Sitzungsbereinigung ausgeführt wird.

NCIZNEP einrichten

CICS stellt das Standard-Knotenfehlerprogramm (NEP) DFHZNEP zur Verfügung, das aufgerufen wird, wenn ein Benutzer die Verbindung zwischen einem Terminal und der CICS-Region trennt. Darüber hinaus bietet CICS das Beispielprogramm DFHZNEPX, das für eine benutzerdefinierte Fehlerbehandlung bestimmt ist, um vom Benutzer bereitgestellte Fehlerverarbeitungsprogramm aufzurufen.

Die Natural CICS-Schnittstelle bietet das Natural-spezifische-Knotenfehlerprogramm NCIZNEP. Sie können das Beispiel-Knotenfehlerprogramm XNCINEP2 verwenden, um DFHZNEP einzurichten und NCIZNEP auszuführen, um eine verlorengegangene Sitzung zu beenden. XNCINEP2 kann anstelle des Beispielprogramms DFHZNEPX verwendet werden.

Das Beispielprogramm XNCINEP2 führt folgende Funktionen aus:

- Es ermöglicht es einem Fehlerverarbeitungsprogramm, die Kontrolle für alle möglichen, an DFHZNEP übergebenen Fehlercodes zu erhalten. Wenn XNCINEP2 die Kontrolle erhält, gibt es eine CICS-Trace-Anfrage aus, die anzeigt, mit welchen Fehlercodes DFHZNEP bei bestimmten Aktionen aufgerufen wurde und welche TCTTE-Fehlerinformationen damit zusammenhängen.
- Ruft über EXEC CICS LINK das NCIZNEP-Modul auf, das eine Natural-Sitzungsbereinigung durchführt.

Das Beispielprogramm XNCINEP2 wird in der Quellcode-Library der Natural CICS-Schnittstelle mitgeliefert. Wenn DFHZNEPX bereits angepasst wurde, können Sie den entsprechenden Quellcode von XNCINEP2 nach DFHZNEPX kopieren und ihn an geeigneter Stelle vor dem abschließenden EXEC CICS RETURN-Aufruf in DFHZNEPX einfügen, wobei andere Produkte oder Komponenten zu berücksichtigen sind, für die eine Fehlerverarbeitung durchgeführt wird.

NCIZNEP-Funktionalität

Das Programm NCIZNEP versucht, eine Sitzung asynchron wiederaufzunehmen und anschließend logisch zu beenden, und zwar auf der Grundlage der Sitzungs-Neustartinformationen (NEXTTRANSID und Neustartdaten in COMMAREA- oder im CICS-Zwischenspeicher) einer Terminalsitzung mit anstehender pseudo-konversationeller Bildschirm-Ein-/Ausgabe.

Nach Beendigung bereinigt NCIZNEP die Parametereingabe und teilt dem Aufrufer (in der Regel DFHZNEP) mit, ob es seine Arbeit erfolgreich abgeschlossen hat, und startet die Bereinigungs-Task für die Natural-Sitzung.

Wenn in einem CICS-System mehr als eine Natural-CICS-Schnittstellenversion aktiv ist, kann die vom aufgerufenen Natural-versionsspezifischen Knotenfehlerprogramm durchgeführte Bereinigungsverarbeitung fehlschlagen, weil die zu beendende Natural-Sitzung von einer anderen Nat-

ural-CICS-Schnittstellenversion gehostet wird. Nach Abschluss der Ausführung von NCIZNEP kann DFHZNEP prüfen, ob die Durchführung der Operation erfolgreich war. Wenn die Operation nicht erfolgreich war, ruft NCIZNEP ein anderes Knotenfehlerprogramm auf, das zu einer anderen Natural CICS-Schnittstellenversion Bezug hat.

Wenn ein Benutzer ein Terminal von der CICS-Region trennt, während die entsprechende Natural-Sitzung im CICS nicht aktiv ist, kann DFHZNEP mehr als einmal die Kontrolle für verschiedene interne Fehlercodes erhalten, da jeder interne Fehlercode mit einer bestimmten CICS-Fehlermeldung verbunden ist.

Wenn Sie möchten, dass NCIZNEP versucht, eine zurzeit aktive Natural-Sitzung, für die NCIZNEP die Kontrolle erhält, sofort zu löschen, geben Sie PURGE=YES für das NCIZNEP-Modul an. Siehe die Installationsprozedur für die Natural CICS-Schnittstelle und den Abschnitt *CICS Startup Parameters* in *Installing Natural CICS Interface on z/OS* in der Natural-*Installation*-Dokumentation.

- Erfordernisse bei MRO/CICSPlex-Umgebungen
- Besondere Aspekte beim Adabas System Coordinator

Erfordernisse bei MRO/CICSPlex-Umgebungen

Allgemein ist bei MRO/CICSPlex-Umgebungen Folgendes erforderlich:

- DFHZNEP und NCIZNEP müssen in der TOR installiert sein.
- Das Modul NCIZNEP muss in CICS mit EXECKEY(CICS) definiert sein.

Bei MRO/CICSPlex-Umgebungen, in denen NCIZNEP eine zurzeit aktive Natural-Session löschen soll, sind die folgenden zusätzlichen Erfordernisse zu beachten:

- Das Modul NCIZNEP muss auch in den mit einer TOR verbundenen CICS AORs definiert sein. Eine Natural-Sitzung kann nur in der AOR bereinigt werden, in der eine Transaktion zur Löschung der Task gestartet wird.
- Für das Modul NCIZNEP muss in der CICS TOR und in den AORs eine Transaktionskennung mit den Attributen TASKDATAKEY(CICS) und PRIORITY(255) definiert werden.
- Die Transaktionskennung muss außerdem mit dem Parameter NEPTRAN in den CICS-Startparametern für das Modul NCIZNEP angegeben werden. Beschreibung siehe Natural-*Installation*-Dokumentation unter *Installing Natural CICS Interface on z/OS*.
- Die Natural-Transaktionen, für die Sie die aktiven Natural-Sitzungen löschen wollen, müssen in CICS mit dem Attribut SPURGE(YES) definiert werden.

Besondere Aspekte beim Adabas System Coordinator

Wenn Sie die Knotenfehlerprogramme NCIZNEP (für die Natural CICS-Schnittstelle) und CORNEP (für den Adabas System Coordinator) von DFHZNEP aus aufrufen wollen, sollten Sie berücksichtigen, dass sowohl NCIZNEP als auch CORNEP möglicherweise versuchen, in Bezug zu Adabas stehende Ressourcen zur gleichen Zeit freizugeben. Um Störungen durch gleichzeitige Ausführung zu vermeiden, müssen Sie NCIZNEP immer vor CORNEP aufrufen und die Startzeit von CORNEP leicht verzögern, um sicherzustellen, dass NCIZNEP beendet ist, bevor CORNEP mit der Ausführung beginnt.

Weitere Informationen siehe *Adabas System Coordinator Installation*-Dokumentation.

Behandlung von Zeitüberschreitungen (Timeouts)

Wird eine Benutzersitzung durch Zeitüberschreitung (Timeout) beendet, werden die von der Natural-Sitzung gehaltenen Ressourcen vom CICS nicht freigegeben. Eine Benutzersitzung wird beispielsweise wegen Zeitüberschreitung beendet, wenn der Benutzer in der Natural-Terminalsitzung für den Zeitraum inaktiv war, der mit dem Parameter TIMEOUT im CICS-Segment des RACF-Benutzerprofils festgelegt wurde.

Im Falle einer Zeitüberschreitung aufgrund der Einstellung des Parameters TIMEOUT im RACF-Benutzerprofil wird die von CICS durchgeführte Aktion durch den Wert des Systeminitialisierungsparameters GNTRAN bestimmt. Natural-Ressourcen werden nicht ordnungsgemäß freigegeben, wenn entweder GNTRAN=NO gesetzt ist oder das ausgeführte “Good Night”-Programm das NCIZNEP-Modul nicht aufruft.

Um sicherzustellen, dass alle von Natural belegten Ressourcen ordnungsgemäß freigegeben werden:

1. Geben Sie für den Parameter GNTRAN eine CICS-Transaktionskennung an, bei der das PROGRAM-Attribut XNCIGNIT angibt.
2. Verwenden Sie das Beispiel-Knotenfehlerprogramm XNCIGNIT, das in der Quellcode-Library der Natural CICS-Schnittstelle enthalten ist (siehe auch ***Natural CICS-Beispielprogramme***), und definieren Sie es in CICS.

   XNCIGNIT muss assembliert und mit den CICS EXEC-Schnittstellenmodulen DFHELII und DFHEAIO verlinkt werden.

   XNCIGNIT ruft über EXEC CICS LINK das Modul NCIZNEP auf, das eine Natural-Sitzungsbereinigung durchführt.

Bei MRO/CICSPlex-Umgebungen müssen Sie die folgenden Schritte durchführen:

- Definieren und installieren Sie in der TOR das Beispielprogramm XNCIGNIT und die CICS-Transaktionskennung, für die das PROGRAM-Attribut XNCIGNIT angibt.

- Fügen Sie im Beispielprogramm `XNCIGNIT` für Knotenfehler den Parameter `SYSIDSYSID` zum Statement `EXEC CICS LINK` hinzu, das `NCIZNEP` ausführt. `SYSID` muss den Systemnamen einer AOR angeben, die sich auf die TOR bezieht. Sie können eine beliebige AOR angeben, da eine CICS-Task für eine Sitzung in einer AOR nicht aktiv ist, wenn die Sitzung eine Zeitüberschreitung auslöst.

API USR4006N zum Abrufen oder Abbrechen von Sitzungen

Die Anwendungsprogrammierschnittstelle (API) USR4006N ermöglicht das Abrufen oder Abbrechen von Natural-Sitzungen auf der Grundlage von Auswahlkriterien, beispielsweise der Dauer der Inaktivität von Terminalsitzungen.

Es kann vorkommen, dass alle Sitzungen sequenziell gelesen werden, um eine oder mehrere Sitzungen zu finden, die einem bestimmten Kriterium entsprechen, da USR4006N nicht in der Lage ist, Sitzungen zufällig nach einem bestimmten Kriterium auszuwählen. Wir empfehlen Ihnen daher, USR4006N nur gelegentlich auszuführen, und auch nur dann, wenn Sie alle Natural-Sitzungen, die ein bestimmtes Kriterium erfüllen, in einem einzigen Durchgang entfernen möchten. Alternativ können Sie das Modul `NCIZNEP` und das Knotenfehler-Beispielprogramm `XNCIGNIT` verwenden, welche nur auf eine einzige Natural-Sitzung wirken und nicht auf alle Sitzungen zugreifen müssen.

9 Unterstützung der CICS 3270 Bridge

- Standardmäßige Unterstützung der CICS 3270 Bridge 60
- Vollständige Unterstützung der CICS 3270 Bridge 60
- NCIXFATU - NCI-Lademodul 60
- Profilparameter DSC=OFF empfohlen 60

Dieses Kapitel beschreibt, wie die CICS 3270 Bridge durch die Natural CICS-Schnittstelle unterstützt wird.

Standardmäßige Unterstützung der CICS 3270 Bridge

Standardmäßig unterstützt die Natural CICS-Schnittstelle die CICS 3270 Bridge insofern, als sie in der Lage ist, mit "gebrückten Geräten" (Bridged Devices) zu arbeiten, d.h. mit Terminals, die über einen CICS 3270 Bridge Exit emuliert werden.

Vollständige Unterstützung der CICS 3270 Bridge

Wenn Sie volle Unterstützung der CICS 3270 Bridge wünschen, müssen Sie das NCI-Lademodul `NCIXFATU` installieren. Siehe entsprechenden Installationsschritt in *Installing Natural CICS Interface on z/OS* in der Natural-*Installation*-Dokumentation.

NCIXFATU - NCI-Lademodul

Das Modul `NCIXFATU` ist ein CICS `XFAINTU` Global User Exit (GLUE). Es dient dazu, Natural-Ressourcen freizugeben, wenn die Belegungszeit der Bridge Facility abgelaufen ist und eine zugehörige Natural-Sitzung noch nicht beendet wurde.

Das Modul `NCIXFATU` ruft das Modul `NCIZNEP` über `EXEC CICS LINK` auf, das eine Bereinigung der Natural-Sitzung durchführt.

Profilparameter DSC=OFF empfohlen

Wenn Sie die CICS 3270 Bridge verwenden, wird empfohlen, eine Natural-Sitzung mit der Einstellung `OFF` des Profilparameters `DSC` (Datenstromkomprimierung für Terminals des Typs 3270 deaktiviert) zu starten, um Natural zu zwingen, immer vollständige Bildschirme und nicht die Differenz (das Delta) zum vorherigen Bildschirm zu senden.

10 Überlegungen zur Thread-Sicherheit

Da Natural vollständig reentrant ist, kann die Natural CICS-Schnittstelle mit dem Attribut `CICS CONCURRENCY(REQUIRED)` als thread-sicher definiert werden. Die Natural CICS-Schnittstelle kann dann in der offenen CICS-Transaktionsumgebung (Open Transaction Environment, OTE) unter einem offenen Thread Control Block (TCB) ausgeführt werden, um die QR CPU-Beschränkung durch Verlagerung von Aufgaben auf andere Prozessoren zu reduzieren.

Die Natural CICS-Schnittstelle sorgt für eine zusätzliche Serialisierung mittels CICS ENQ/DEQ, wenn sie unter einem offenen TCB ausgeführt wird.

Um diesen Serialisierungsaufwand zu minimieren, ist es sehr empfehlenswert,

- Threads mit `TYPE=GETM` ohne Angabe des Parameters `THREADS` (oder `THREADS=0`) zu verwenden,
- den Natural Roll Server anstelle der Roll Facilities in CICS zu verwenden.

Notes:

1. Alle Benutzerprogramme, die als `CSTATIC` definiert sind, müssen thread-sicher sein.
2. Alle dynamischen Benutzerprogramme müssen thread-sicher sein, wenn sie unter Verwendung von Standard-Linkage-Konventionen entweder explizit (d.h. mit dem Terminalkommando `%P=S`, `%P=SC`, `*P=L` oder `%P=LS` oder mit dem Profilparameter `PGP` mit der entsprechenden Eigenschaft) oder implizit (d.h. wenn im Makro `NTCICSP` der Subparameter `SLCALL` auf `ON` gesetzt ist) aufgerufen werden. Wenn diese Programme nicht thread-sicher sind, können Sie das Terminalkommando `%P=SQ` oder den Profilparameter `PGP` (mit der Eigenschaft `STDLQ`) verwenden, um quasi-reentrante Benutzerprogramme aufzurufen. Weitere Informationen zu thread-sicheren Programmen finden Sie in der IBM *Application Programming*-Dokumentation.

11 Überlegungen zur CICS Open Transaction-Umgebung

Wenn Sie Natural in der offenen CICS-Transaktionsumgebung (Open Transaction Environment, OTE) unter einem offenen Thread Control Block (TCB) ausführen möchten, müssen alle externen Subprogramme thread-sicher und vollständig reentrant sein, wenn sie von Natural unter Verwendung von Standard-Linkage-Konventionen (direkte Verzweigung mit einer BASR-Anweisung) anstelle eines `EXEC CICS LINK`-Kommandos aufgerufen werden. Es reicht nicht aus, dass die aufgerufenen Subprogramme nur quasi-reentrant sind. Dies gilt für alle externen Subprogramme, die mit den Profilparametern `CSTATIC`, `RCA` oder `PGP` unter Natural definiert sind oder nach Absetzen des Terminalkommandos `%P=S` aufgerufen wurden.

Wenn ein Subprogramm quasi-reentrant, aber nicht thread-sicher ist, müssen Sie das Terminalkommando `%P=SQ` anstelle von `%P=S` verwenden, oder Sie müssen den Profilparameter `PGP` mit gesetzter Eigenschaft `STDLQ` verwenden. Bei Verwendung des Terminalkommandos `%P=SQ` oder des Profilparameters `PGP` (mit `STDLQ`) wird das externe Subprogramm auf dem CICS QR TCB und nicht unter einem offenen TCB ausgeführt. Natural gibt dann das CICS-Kommando `EXEC CICS LINK PROGRAM('NCILINKQ')` aus, um zum QR TCB zu wechseln und das `NCILINKQ`-Programm auszuführen, das das Subprogramm unter Verwendung von Standard-Linkage-Konventionen aufruft.

Wenn Sie das Terminalkommando `%P=SQ` oder den Profilparameter `PGP` (mit `STDLQ`) verwenden, müssen Sie `NCILINKQ` so definieren, wie in Routing-Modul für quasi-reentrante Standard-Linkage-Aufrufe (`%P=SQ`) im Abschnitt *Installing Natural CICS Interface* in der *Installation for z/OS*-Dokumentation beschrieben.

Wenn ein aufgerufenes externes Subprogramm nicht mit dem Profilparameter `CSTATIC` oder `RCA` unter Natural definiert ist, aber einen PPT-Eintrag im Operating PPT hat und sich in einer Load Module Library befindet, so dass CICS das Subprogramm lokalisieren und laden kann, wird der Aufruf durch ein `EXEC CICS LINK`-Kommando durchgeführt. Wenn das aufgerufene Subprogramm mit `CONCURRENCY(QUASIRENT)` definiert ist, schaltet CICS automatisch auf den QR TCB um, um den Aufruf auszuführen.

Wenn ein aufgerufenes externes Subprogramm mit dem Profilparameter `CSTATIC` oder `RCA` unter Natural definiert ist, gelten das Terminalkommando `%P=SQ` und der Profilparameter `PGP` (mit `STDLQ`) nicht.

Um Probleme zu vermeiden, sollten Sie die Programmattribute `API(OPENAPI)` und `CONCURRENCY(REQUIRED)` konsequent für alle Programmaufrufe verwenden, die eine direkte Verzweigungsanweisung (BASR) verwenden. Dies gilt auch für Frontend-Programme von Natural-Add-on-Produkten, die den umgebungsabhängigen Nukleus von Natural aufrufen. Weitere Informationen finden Sie unter ***Überlegungen zur Thread-Sicherheit***.

12 Unterstützung von CICS-Channels und -Containern

Der IBM CICS Transaction Server für z/OS unterstützt Channels und Container für `EXEC CICS LINK`. In dieser Hinsicht können CICS-Container als benannte COMMAREAs ohne die 32 KB-Grenze betrachtet werden.

Die Natural CICS-Schnittstelle unterstützt CICS-Container auf zwei Arten:

1. Über ein spezielles Statement `SET CONTROL 'P=CC'` werden die Parameterdaten des `CALL`-Statements in einem Container übergeben.
2. Wenn im Makro `NTCICSP` der Subparameter `CNTCALL` auf `ON` gesetzt ist, verwendet ein `CALL` mit `%P=C` automatisch einen CICS Container statt einer CICS COMMAREA, wenn die mit dem `CALL`-Statement übergebenen Parameterdaten 32 KB überschreiten.

In beiden Fällen ist der Standard-Containername `NCI-COMMAREA`, es sei denn, ein Containername wird ausdrücklich über die Anwendungsprogrammierschnittstelle USR4204N vor dem “echten” `CALL`-Statement definiert.

13 Natural CICS-Schnittstelle und IBM Language Environment (LE)

Die Natural CICS-Schnittstelle unterstützt LE-Programme. Dieses Kapitel enthält Informationen zur LE-Fähigkeit von Natural unter CICS.

Der CICS Transaction Server for z/OS ist LE-kompatibel.

Wenn der bei Ihnen installierte CICS Transaction Server for z/OS die Natural CICS-Schnittstelle unterstützt, ist diese von sich aus LE-kompatibel, d.h. eine Natural CICS-Schnittstellen-Task kann LE-Programme, die in Sprachen wie C, COBOL oder PL/I geschrieben sind, direkt per `CALL` aufrufen (Standard-Linkage-Konventionen, nicht CICS LINK), wenn

- `SET CONTROL 'P=LS'` angegeben wurde,
- `SET CONTROL 'P=S'` angegeben wurde,
- der Subparameter `SLCALL=ON` im Makro `NTCICSP` angegeben wurde und das aufzurufende Programm *kein* CICS-Programm ist.

14 Spezielle Natural CICS-Funktionalität

- Aufruf von Nicht-Natural-Programmen 70
- Dummy-Bildschirm-Ein-/Ausgabe bei Natural unter CICS 71

In diesem Kapitel werden spezielle Natural CICS-Funktionen erläutert. Folgende Themen werden behandelt:

Aufruf von Nicht-Natural-Programmen

Eine der ersten Aktionen, die eine Natural-Task beim Start ausführt, ist die Aktivierung eines Exits für die Verarbeitung abnormaler Beendigungen. Dieser Exit wird verwendet, um alle Ressourcen einschließlich des Threads im Falle einer abnormalen Beendigung freizugeben. Daher darf ein Nicht-Natural-Programm nicht `EXEC CICS ABEND CANCEL` oder die entsprechende Makroanforderung ausgeben, weil eine solche Anforderung die aktuelle Sitzung ohne Berücksichtigung eines aktiven Exits abbricht. Dadurch ist Natural nicht in der Lage, seine Ressourcen aufzuräumen, und der Thread und die Roll Facility werden nicht freigegeben.

Ein Thread wird einer Natural-Task zugewiesen, wenn ein Natural-Programm aktiv ist. Dies gilt auch, wenn Nicht-Natural-Programme aufgerufen werden (gemäß den CICS-Linkage-Konventionen).

Daher sollten solche Programme keine übermäßigen Ein-/Ausgaben oder andere Arbeiten ausführen, ohne dass Natural zwischendurch die Kontrolle erhält. Falls ein Nicht-Natural-Programm Ein-/Ausgaben auf dem Bildschirm durchführt, können Sie im Natural-Programm ein `SET CONTROL 'P=V'`-Statement codieren, das das Nicht-Natural-Programm vor dem aufrufenden Statement aufruft: Dies zeigt an, dass die Parameterdaten aus dem Thread kopiert werden und die Sitzung vor dem Aufruf des Nicht-Natural-Programms ausgespeichert wird.

Folgende Themen werden anschließend behandelt:

- Aufruf von Nicht-Natural-Programmen über Standard-Linkage-Konventionen
- Aufruf von Nicht-Natural-Programmen mit Parameterwerten in einer COMMAREA oder in einem Container

Aufruf von Nicht-Natural-Programmen über Standard-Linkage-Konventionen

Ein Nicht-Natural-Programm wird von Natural in der Weise aufgerufen (per `CALL`) wie Programme im zugrunde liegenden Betriebs- und/oder TP-Monitorsystem aufgerufen werden.

In CICS werden Nicht-Natural-Programme durch `EXEC CICS LINK`-Anfragen aufgerufen. Wenn jedoch z.B. dasselbe Subprogramm (das keine CICS- oder Betriebssystemanfragen ausgibt) sowohl für die Batch- als auch für die Online-Verarbeitung verwendet werden soll, kann ein Nicht-Natural-Programm auch über Standard-Linkage-Konventionen aufgerufen werden, d.h. über `BASR R14,R15`.

Weitere Informationen finden Sie unter dem Terminalkommando `%P=S` in der *Terminalkommandos*-Dokumentation. Siehe auch Subparameter `SLCALL` im Makro `NTCICSP`.

Aufruf von Nicht-Natural-Programmen mit Parameterwerten in einer COMMAREA oder in einem Container

Standardmäßig werden Nicht-Natural-Programme mit den Adressen der Anfrageparameterlisten (siehe Beschreibung des `CALL`-Statement in der Natural-*Statements*-Dokumentation) aufgerufen, die im TWA und/oder einer COMMAREA übergeben werden (abhängig von der Einstellung des Subparameters `CALLRPL` des Makros `NTCICSP`).

Eine an CICS orientierte Methode ist die Übergabe der Parameterwerte in einer CICS-COMMAREA oder einem CICS-Container (siehe ***Unterstützung von CICS-Channels und -Containern***), insbesondere wenn sich das aufgerufene Programm in einer anderen CICS-Region befindet (Distributed Program Link, DPL), da Adressen innerhalb der "aufrufenden" Region für die "aufgerufene" Region nicht zugänglich sind.

Einzelheiten und Einschränkungen finden Sie bei den Terminalkommandos `%P=C` und `%P=CC` in der *Terminalkommandos*-Dokumentation.

Voraussetzung:

Diese Funktionalität erfordert, dass der Subparameter `CALLRPL` im Makro `NTCICSP` auf `ON` gesetzt ist.

Wenn die Parameterwerte in einer CICS COMMAREA oder einem CICS Container übergeben werden, werden unabhängig von der `CALLRPL`-Einstellung keine Parameterlistenzeiger in der CICS TWA übergeben.

Dummy-Bildschirm-Ein-/Ausgabe bei Natural unter CICS

Wenn ein `SET CONTROL 'QO'`-Statement vor einem Natural-Statement steht, das eine Bildschirm-Eingabe/-Ausgabe bewirkt, wird die Terminalausgabe von Natural unter CICS nicht ausgeführt. Folglich werden sowohl die Enter-Taste als auch die Benutzereingabe nicht an Natural rückübertragen.

Diese Funktionalität kann in den folgenden Situationen nützlich sein:

1. Wenn Sie pseudo-konversationelle Bildschirm-Ein-/Ausgaben an Nicht-Natural-Programme überlassen, die von Natural aufgerufen werden. Das Nicht-Natural-Programm führt die Operation `EXEC CICS SEND` durch und kehrt zu Natural zurück. Aufgrund des Statement `SET CONTROL 'QO'` beendet die nächste Natural-Bildschirm-Ein-/Ausgabe die Task einer Pseudo-Konversationssitzung. Bei der Bildschirmeingabe erhält Natural die Kontrolle und ruft erneut das Nicht-Natural-Programm auf, das dann die Operation `EXEC CICS RECEIVE` durchführt.
2. Beim Ändern der pseudokonversationellen Natural-Transaktionskennung während der laufenden Vorgangs, ohne dass ein Terminalbediener eingreifen muss:

```
MOVE *INIT-ID TO termid
CALLNAT 'CMTRNSET'  trnid  /* change the restart transaction ID

* starting a task on your terminal forces an interrupt as if
* pressing any attention identifier

CALL 'CMTASK' USING trnid
H'0000' H'00000000' termid
SET CONTROL 'QO'
INPUT 'DUMMY'                 /* dummy I/O, which you will never see
WRITE 'HELLO' *INIT-PROGRAM   /* now the new transaction ID is active ↩
```

3. Beim Wechsel zu einer Anwendung außerhalb von Natural, vielleicht sogar in einer anderen CICS AOR (Application Owning Region), ohne dass ein Terminalbediener eingreifen muss:

```
* starting a task on your terminal forces an interrupt as if
* pressing any attention identifier

CALL 'CMTASK' USING trnid data-length start-data termid
SET CONTROL 'QO'
INPUT 'DUMMY'                  /* dummy I/O, which you will never see
WRITE 'HELLO' *INIT-PROGRAM    /* now the new transaction ID is active
```

In diesem Fall ist die aufgerufene Anwendung dafür verantwortlich, die Natural-Neustartdaten auf den Stack zu legen, wenn sie in einer CICS-COMMAREA übergeben werden, da eine COMMAREA höchstwahrscheinlich auch von der neuen (pseudo-konversationellen) Anwendung verwendet wird.

15 Natural CICS-Beispielprogramme

- Frontend-Programme 74
- Backend-Programme 75
- User Exits 75
- Subprogramm-Aufrufe (Calls) 76
- Sonstige Programme 76

Dieses Kapitel enthält eine Übersicht über die Natural CICS-Beispielprogramme, die in der Natural CICS Source Library enthalten sind.

Frontend-Programme

Name	Sprache	Funktion
XNCIFRNP	Assembler	Initialisierungsprogramm, das die Natural CICS-Umgebung beim CICS-Start initialisiert.
XNCIFRNL	Assembler	Front-end program for invoking Natural via EXEC CICS LINK.
XNCIFRNR	Assembler	Frontend-Programm für den Aufruf von Natural über EXEC CICS RETURN IMMEDIATE.
XNCIFRNS	Assembler	Frontend-Programm für den Aufruf von Natural über EXEC CICS START.
XNCIFRNX	Assembler	Frontend-Programm für den Aufruf von Natural über EXEC CICS XCTL.
XNCIFRNN	Assembler	Frontend-Programm für den Aufruf von Natural über EXEC CICS LINK ohne Frontend-Parameter.
XNCIFRCL	COBOL	Frontend-Programm für den Aufruf von Natural über EXEC CICS LINK.
XNCIFRCN	COBOL	Dies ist ein Dummy-Frontend-Programm für den Aufruf von Natural über EXEC CICS LINK zwecks LE-Konformität.
XNCIFRCR	COBOL	Frontend-Programm für den Aufruf von Natural über EXEC CICS RETURN IMMEDIATE.
XNCIFRCS	COBOL	Frontend-Programm für den Aufruf von Natural über EXEC CICS START.
XNCIFRCX	COBOL	Frontend-Programm für den Aufruf von Natural über EXEC CICS XCTL.
XNCIFRPL	PL/1	Frontend-Programm für den Aufruf von Natural über EXEC CICS LINK.
XNCIFRPN	PL/1	Dies ist ein Dummy-Frontend-Programm für den Aufruf von Natural über EXEC CICS LINK zwecks LE-Konformität.
XNCIFRPR	PL/1	Frontend-Programm für den Aufruf von Natural über EXEC CICS RETURN IMMEDIATE.
XNCIFRPS	PL/1	Frontend-Programm für den Aufruf von Natural über EXEC CICS START.
XNCIFRPX	PL/1	Frontend-Programm für den Aufruf von Natural über EXEC CICS XCTL.
XNCIFRDN	C	Dies ist ein Dummy-Frontend-Programm für den Aufruf von Natural über EXEC CICS LOAD and BASR zwecks LE-Konformität.

Backend-Programme

Name	Sprache	Funktion
XNCIBACK	Assembler	Abbruchdaten-Dump: Dieses Backend-Programm zeigt die Natural-Abbruchmeldung und alle Abbruchdaten im Dump-Format an. Wenn es von einer asynchronen Task aufgerufen wird, wird die Natural-Abbruchmeldung auf der Bedienerkonsole ausgegeben, und mögliche Abbruchdaten werden im Dump-Format angezeigt. NCIBACK kann auch von einer Backend-Transaktion aufgerufen werden (RET=*xxxx* oder RTI=*xxxx* oder STR=*xxxx*, dabei ist *xxxx* der mit XNCIBACK verbundene Transaktionscode).

User Exits

Name	Sprache	Funktion
XNCIDIRX	Assembler	**System Directory Module Name Exit** Dieses Quellcode-Modul enthält ein Beispiel für einen Systemverzeichnismodulnamen-Exit (siehe auch ***NCIDIREX - Systemverzeichnismodulnamen-Exit-Schnittstelle***).
XNCIDTPX	Assembler	**DTP Terminal Exit** Dieses Quellcode-Modul enthält ein Beispiel für einen DTP-Terminal-Exit (siehe auch ***DTP-Terminal-Ein-/Ausgabe-Exit-Schnittstelle***).
XNCIRDC1	Assembler	**Exit für SYSRDC** Dieses Programm enthält einen Beispiel-Exit für das Dienstprogramm SYSRDC, siehe den entsprechenden Abschnitt in der *Debugger und Dienstprogramme*-Dokumentation.
XNCITIDX	Assembler	**Terminal ID Exit** Dieses Programm enthält ein Beispiel für einen User Exit zum Testen der Terminalkennung und/oder zum Setzen einer logischen Terminal- oder Sitzungskennung (siehe auch ***NCITIDEX - Terminalkennung-Exit-Schnittstelle***).
XNCITIOX	Assembler	**DTP Terminal Exit** Dieses Quellcode-Modul enthält einen Terminal-I/O-Exit, der genereller ist als das XNCIDTPX-Beispiel (siehe auch ***NCIDTPEX - DTP-Terminal-Ein-/Ausgabe-Exit-Schnittstelle***).

Name	Sprache	Funktion
XNCIUIDX	Assembler	**User ID Exit** Dieses Programm enthält ein Beispiel für einen User Exit zum Testen/Einstellen der Benutzerkennung (siehe auch *NCIUIDEX - Benutzerkennung-Exit-Schnittstelle*).
XNCIXIDX	Assembler	**Transaction ID Exit** Dieses Programm enthält einen Beispiel-User Exit zum Testen/Einstellen der pseudo-konversationellen Transaktionskennung (siehe auch *NCIXIDEX - Transaktionskennung-Exit-Schnittstelle*).

Subprogramm-Aufrufe (Calls)

Name	Sprache	Funktion
XNCI3GC1	COBOL	Dieses Programm enthält ein Beispiel für einen COBOL-Aufruf eines Natural-Subprogramms unter CICS.
XNCI3GC2	COBOL	Dieses Programm enthält ein Beispiel für einen COBOL-Aufruf eines Natural-Subprogramms unter CICS.
XNCI3GC3	COBOL	Dieses Programm enthält ein Beispiel für einen COBOL-Aufruf eines Natural-Subprogramms unter CICS.
XNCI3GP1	PL/1	Dieses Programm enthält ein Beispiel für einen PL/1-Aufruf eines Natural-Subprogramms unter CICS.
XNCI3GP2	PL/1	Dieses Programm enthält ein Beispiel für einen PL/1-Aufruf eines Natural-Subprogramms unter CICS.
XNCI3GP3	PL/1	Dieses Programm enthält ein Beispiel für einen PL/1-Aufruf eines Natural-Subprogramms unter CICS.

Sonstige Programme

Name	Sprache	Funktion
XNCIUCTR	Assembler	Umschaltung Groß-/Kleinschreibung: Dieses Programm dient dazu, das Terminal auf Groß-/Kleinschreibung umzuschalten.
XNCINEP2	Assembler	Dieses Knotenfehlerprogramm ruft das NCIZNEP-Modul auf.
XNCIGNIT	Assembler	"Good Night"-Programm: Dieses Beispielprogramm ruft das NCIZNEP-Modul zum Bereinigen von Natural-Sitzungen auf.
XNCITRPX	Assembler	Hilfe beim Debugging: Falls für die Fehlerdiagnose erforderlich, wird der Support Sie über die Verwendung dieses Programms informieren.

16 Aufruf von Natural aus Benutzerprogrammen

- Kommandos zum Aktivieren einer Natural-Sitzung 78
- Frontend-Parameter 80
- Frontend-Aufruf über LINK 81
- Frontend-Aufruf über RETURN IMMEDIATE 82
- Frontend-Aufruf über START 82
- Frontend-Aufruf über XCTL 82
- Frontend-Aufruf über Distributed Program Link (DPL) 83
- Aufruf eines Frontend-Programms als Backend 83

Dieses Kapitel beschreibt die verschiedenen Möglichkeiten, wie Natural aus Benutzerprogrammen aufgerufen werden kann.

Kommandos zum Aktivieren einer Natural-Sitzung

Eine Natural-Sitzung kann von Benutzer-Frontend-Programmen mit einem der folgenden Kommandos aktiviert werden:

- `EXEC CICS XCTL`
- `EXEC CICS LINK`
- `EXEC CICS RETURN IMMEDIATE`
- `EXEC CICS START`

oder die entsprechenden Anforderungen aus der CICS-Makroebene.

Die Kommandos zum Aktivieren einer Natural-Sitzung werden in den folgenden Abschnitten beschrieben.

- Verwendung von EXEC CICS XCTL oder EXEC CICS LINK
- Verwendung von EXEC CICS RETURN IMMEDIATE
- Verwendung von EXEC CICS START
- Verwendung von Distributed Program Link (DPL)
- Beispielprogramme
- Verwendung der externen Subroutine CMTASK

Verwendung von EXEC CICS XCTL oder EXEC CICS LINK

Bei Verwendung von `EXEC CICS XCTL/LINK` können die von Natural verwendeten Parameter in einer CICS COMMAREA oder in der TWA übergeben werden.

- Natural bestimmt, an welcher Stelle sich die Startparameter befinden, indem es die Länge der COMMAREA prüft, die ihm während der Sitzungsinitialisierung zur Verfügung übergeben wird.
- Wenn die Länge 22 beträgt, versucht Natural, die Parameter in der COMMAREA zu finden, andernfalls wird angenommen, dass sie in der TWA übergeben wurden.

Um ein Frontend-Programm korrekt zu identifizieren, müssen die ersten 4 Bytes der Frontend-Parameterliste die aktuelle Transaktionskennung enthalten.

Die mit dem Frontend-Programm verknüpfte Transaktionskennung muss eine TWA-Größe haben, die gleich oder größer als die Natural-TWA-Größe ist. Siehe auch `ncitransact` in *Installing Natural CICS Interface on z/OS* in der Natural *Installation*-Dokumentation.

Verwendung von EXEC CICS RETURN IMMEDIATE

Bei Verwendung von `EXEC CICS RETURN IMMEDIATE` können die von Natural verwendeten Frontend-Parameter in einer CICS COMMAREA und die von Natural verwendeten dynamischen Parameter mit `INPUTMSG (...)` und `INPUTMSGLEN (...)` des Kommandos `EXEC CICS RETURN IMMEDIATE` übergeben werden.

Verwendung von EXEC CICS START

Bei Verwendung von `EXEC CICS START` können die von Natural verwendeten Frontend- und dynamischen Parameter mit `FROM (...)` und `LENGTH (...)` des Kommandos `EXEC CICS START` übergeben werden. Die Parameter werden weiter unten beschrieben.

Verwendung von Distributed Program Link (DPL)

Wenn Sie `EXEC CICS LINK` mit dem Parameter `SYSID` verwenden, der auf eine entfernte Region zeigt, müssen die von Natural verwendeten Frontend- und dynamischen Parameter in einer CICS COMMAREA übergeben werden. Außerdem muss ein `TRANSID`-Parameter angegeben werden, der den Transaktionscode einer Spiegeltransaktion mit einer TWA-Größe angibt, die den Erfordernissen von Natural entspricht.

Es ist zu beachten, dass für Natural-Sitzungen, die über DPL aufgerufen werden, dieselben Einschränkungen wie für asynchrone Natural-Sitzungen gelten, d.h. es sind keine Eingaben möglich (daher werden dieselben dynamischen Parametereinstellungen empfohlen) und der Aufrufer erhält nach Beendigung der Natural-Sitzung lediglich die Kontrolle zurück.

Beispielprogramme

Eine Reihe von Beispielprogrammen für die verschiedenen Programmiertechniken ist in der Natural CICS Source Library enthalten. Siehe auch ***Natural CICS-Beispielprogramme***

Verwendung der externen Subroutine CMTASK

Sie können eine Natural-Sitzung aus einem Natural-Programm heraus starten, indem Sie die externe Subroutine `CMTASK` aufrufen. Siehe dazu das Natural-Beispielprogramm `ASYNCICS` in der Library `SYSEXTP`.

Frontend-Parameter

Die folgende Liste mit Parametern muss übergeben werden, um Natural von einem Benutzer-Frontend-Programm aus aufzurufen:

Pos.	Inhalt
1 - 4	**Aufrufende Transaktionskennung** Dieser Wert muss mit der aktuellen Transaktionskennung übereinstimmen. Über die aufrufende Transaktionskennung erkennt Natural, dass es von einem Benutzer-Frontend-Programm aufgerufen wurde. Beim Aufruf mit `XCTL` wird die Transaktion am Ende der Natural-Sitzung über `RETURN` mit `TRANS ID` neu gestartet, es sei denn, es wird ein Rückkehrprogrammname angegeben (siehe 5. Parameter).
5 - 8	**Adresse/Offset des dynamischen Parameterstrings** Wenn dynamische Parameterüberschreibungen ausgewertet werden sollen, sollte dieser Wert auf die Adresse gesetzt werden, die 12 Bytes vor dem dynamischen Parameterzuweisungsstring liegt. Beim Aufruf mit `START` oder DPL muss das Feld auf den Offset (relativ zum Beginn der Frontend-Parameterliste) der Adresse gesetzt werden, die 12 Bytes vor dem dynamischen Parameterzuweisungsstring liegt.
9 - 10	**Länge des dynamischen Parameterstring** Null bedeutet, dass keine Parameter übergeben werden sollen. `32760` ist die maximal zulässige Länge. Wird der Maximalwert überschritten, wird die Sitzung mit einer entsprechenden Fehlermeldung beendet.
11 - 14	**Natural-Transaktionskennung** Der angegebene Wert ist die Transaktionskennung, die zur Steuerung einer pseudokonversationellen Natural-Sitzung verwendet wird, wenn diese mit `START` oder `XCTL` aufgerufen wird. Diese Transaktion wird jedes Mal aufgerufen, wenn die Natural-Sitzung im pseudokonversationellen Modus neu gestartet wird, d.h. bei jeder Terminal-Ein-/Ausgabe. Wird die Natural-Transaktionskennung nicht angegeben, startet Natural erneut mit der Transaktionskennung, mit der die aktuelle CICS-Task gestartet wurde, und das Frontend-Programm erhält nach jeder pseudokonversationellen Ein-/Ausgabe wieder die Kontrolle zurück.
15 - 22	**Name des Backend-Programms** Dieser 8-Byte-Wert ist der Name des Programms, an das die Kontrolle am Ende der Natural-Sitzung mit einem CICS `XCTL`-Kommando übergeben wird, anstatt die aufrufende Transaktionskennung über `RETURN` mit `TRANSID` neu zu starten.

Pos.	Inhalt
	Wenn das erste Byte dieses Feldes numerisch ist, führt Natural einfach `RETURN` aus, ohne ein Backend zu aktivieren. Beachten Sie, dass dieses Feld durch den Natural-Profilparameter `PROGRAM` überschrieben werden kann. Die Konventionen für den Aufruf von Nicht-Natural-Backend-Programmen finden Sie unter *Konventionen für den Aufruf von Backend-Programmen* in der Natural *Operations*-Dokumentation.

Frontend-Aufruf über LINK

Bei der Rückkehr zum Frontend gibt Natural in der TWA an, ob die Sitzung beendet wurde oder nicht: Wenn die Sitzung beendet wurde, enthält die TWA reguläre Backend-Informationen (siehe *Konventionen für den Aufruf von Backend-Programmen* in der *Operations*-Dokumentation), andernfalls setzt Natural die `NEXTTRANSID` in die ersten vier Bytes der TWA.

Wenn Natural im pseudokonversationellen Modus läuft (Profilparameter `PSEUDO` auf `ON` gesetzt) und durch `EXEC CICS LINK` (oder die entsprechende Anforderung auf CICS-Makroebene) aufgerufen wurde, wird die ursprüngliche aufrufende Transaktion jedes Mal aufgerufen, wenn Natural auf ein Terminal schreibt und auf eine Eingabe wartet, was bedeutet, dass Natural eine "logische" `CICS RETURN TRANSID (..)` ausgibt, nachdem es seine Neustartinformationen in den CICS-Zwischenspeicher geschrieben hat.

Die aufrufende Transaktion muss diese Situation erkennen (z.B. durch Überprüfung, ob ein `NEXTTRANSID` gesendet wurde oder durch das Vorhandensein von `NCOMxxxx` TS-Sätzen - wobei `NCOM` die Natural CICS-Parametergenerierungsoption und *`xxxx`* die Terminalkennung ist -) und die Kontrolle an Natural zurückgeben.

Der Vorteil dieser Vorgehensweise ist, dass das Frontend-Programm während der Sitzung beschließen kann, die Kontrolle an eine andere Anwendung (z.B. COBOL) zu übergeben und die Natural-Sitzung später wieder aufzunehmen.

Weitere Einzelheiten finden Sie in der Beschreibung des Parameters `PSEUDO` in der *Parameter-Referenz*-Dokumentation.

Natural behandelt ein LINK-Frontend-Programm beim Beenden der Sitzung wie ein Backend-Programm, d.h. es gelten die *Konventionen für den Aufruf von Backend-Programmen*.

In CICSplex-Umgebungen

Stellen Sie sicher, dass alle beteiligten CICS-AORs auf die `NCOM`*xxxx* TS-Sätze zugreifen können (z.B. über entsprechende CICS-TST-Definitionen).

Alternativ kann das LINK-Frontend-Programm die NCI-Sitzungsneustart-Informationen bei Task-Ende auch im CICS-Zwischenspeicher abholen und selbst in einer CICS-COMMAREA übergeben. Eine solche COMMAREA muss dann vor dem Aufruf von Natural zur Sitzungswiederaufnahme wieder im CICS-Zwischenspeicher abgelegt werden.

Frontend-Aufruf über RETURN IMMEDIATE

Frontend-Aufruftechnik funktioniert nur bei Natural-Terminalsitzungen. Natural sucht nach Startparametern, die mit dem COMMAREA übergeben werden. Beachten Sie, dass bei dieser Technik potenzielle dynamische Parameter nicht verkettet an die Frontend-Parameter übergeben werden können, d.h. die Adressfelder der dynamischen Parameter müssen Null sein. Stattdessen können potenzielle dynamische Parameter über Terminaleingabedaten übergeben werden, die Natural durch ein `EXEC CICS RECEIVE`-Kommando erhält.

Frontend-Aufruf über START

Wenn es sich bei der Natural-Sitzung um eine gestartete Task handelt (d.h. durch ein `EXEC CICS START`- oder `EXEC CICS LINK/XCTL`-Kommando von einem gestarteten Frontend-Benutzerprogramm aufgerufen), sucht Natural zunächst nach mit der COMMAREA gelieferten Startparametern, dann nach Parametern in der TWA und schließlich versucht es, die erforderlichen Parameter durch ein `EXEC CICS RETRIEVE`-Kommando zu erhalten.

Frontend-Aufruf über XCTL

Wenn die Natural-Sitzung von einem Frontend-Programm mit `XCTL` initiiert wird und kein Rückkehrprogramm angegeben ist (d.h. weder ein fünfter Parameter in den Sitzungsstartparametern noch eine `PROGRAM`-Angabe in den dynamischen Natural-Parametern oder dem `NTPRM`-Makro), startet Natural die Benutzer-Frontend-Transaktion bei Sitzungsende über `RETURN` mit `TRANSID` neu, indem es intern eine Angabe `PROGRAM='RET=`*xxx*`'` simuliert, wobei *xxx* der Code der Frontend-Transaktion ist.

Um eine Schleifenbildung zu vermeiden, muss in die Benutzer-Frontend-Routine eine Logik aufgenommen werden, die entscheidet, ob eine neue Sitzung gestartet oder eine alte Sitzung fortgesetzt werden soll.

Frontend-Aufruf über Distributed Program Link (DPL)

Wenn die Natural-Sitzung über DPL aufgerufen wird, stellt Natural zunächst fest, ob sie direkt in der Server-Region oder indirekt über `EXEC CICS LINK/XCTL` durch ein lokales Frontend-Programm aufgerufen wird. Bei einem direkten Aufruf holt Natural die Startparameter aus der CICS COMMAREA. Beim indirekten Aufruf sucht Natural nach Startparametern, die mit der COMMAREA geliefert werden, und dann nach Parametern in der TWA. Bei der Rückkehr übergibt Natural normale Backend-Daten in der TWA, wenn ein lokales LINK-Frontend-Programm verfügbar ist, andernfalls gibt es die Abbruchmeldung und potenzielle Backend-Daten in der COMMAREA des Remote-Client zurück.

Aufruf eines Frontend-Programms als Backend

Wenn die Natural-Sitzung von einem Frontend-Programm initiiert wird und dieses Programm auch als Rückkehrprogramm angegeben ist, sollte das Benutzer-Frontend auch die initiierende Transaktionskennung überprüfen.

Dies gilt insbesondere dann, wenn sich das Frontend-Programm nicht im pseudokonversationellen Modus befindet, Natural aber im konversationellen Modus ist.

In diesem Fall wird Natural erneut aufgerufen, anstatt beendet zu werden, aber dieses Mal ohne zu erkennen, dass es von einem Frontend-Programm aufgerufen wird, da der erste Parameter in den Startparametern die Natural-Transaktionskennung ist.

17 Asynchrone Natural-Verarbeitung unter CICS

- Asynchrone Natural-Verarbeitung 86
- Asynchrone Natural-Sitzungen unter CICS 86
- Testen und Debuggen 87

Dieses Kapitel befasst sich mit speziellen Aspekten, die für den Einsatz von asynchronem Natural unter CICS gelten.

Asynchrone Natural-Verarbeitung

Die asynchrone Natural-Verarbeitung wird generell im Abschnitt *Asynchrone Verarbeitung* in der *Operations*-Dokumentation behandelt. Bei der Ausführung unter CICS sind jedoch einige zusätzliche Überlegungen nötig, die in den folgenden Abschnitten beschrieben werden.

Asynchrone Natural-Sitzungen unter CICS

Achten Sie darauf, dass bei einer asynchronen Natural-Sitzung geeignete `SENDER`- und `OUTDEST`-Ziele (Destinations) angegeben werden, da sonst jede Ausgabe (z.B. unerwartete Fehlermeldungen) zu einer abnormalen Beendigung führt.

Stellen Sie außerdem sicher, dass im Makro `NTCICSP` des Natural-Parametermoduls eine geeignete Transaktionskennung für die Meldungsumschaltung (Message Switching Transaction ID) mit dem Subparameter `MSGTRAN` angegeben und im CICS definiert ist.

Zusätzlich zu den CICS-Terminalkennungen und transienten Datenzielen für `SENDER` und `OUTDEST` werden die folgenden Schlüsselwörter von der Natural CICS-Schnittstelle unterstützt:

`DUMMY`	Jede Ausgabe wird ignoriert.
`CONSOLE`	Jegliche Ausgabe wird an die Bedienerkonsole weitergeleitet. Beim Umgang mit der Konsole sollte der Terminaltyp entsprechend umgeschaltet werden, indem der Profilparameter `TTYPE` oder das Terminalkommando `%T=` auf `ASYL` oder entsprechend andere Werte gesetzt wird.

Standardmäßig wird das 3270-Datenstromprotokoll für die Ausgabe einer asynchronen Natural-Sitzung unter CICS verwendet.

Es ist auch möglich, Natural-Ausgabedaten ohne 3270-Terminal- oder Druckersteuerinformationen zu senden, z.B. an ein CICS-Nachrichtenziel wie CSSL. Dies kann durch Umschalten in den Zeilenmodus mit dem Statement `SET CONTROL 'T='` oder durch Starten mit dem Profilparameter `TTYPE=`*`xxxx`* erreicht werden, wobei *`xxxx`* für `BTCH` oder `ASYL` steht. Die gesamte Natural-Ausgabe wird dann zeilenweise gesendet, mit einem vorangestellten ASA-Steuerzeichen, wenn der Natural-Profilparameter `EJ` auf `ON` gesetzt ist. Bei `EJ=OFF` wird überhaupt kein Steuerzeichen gesendet.

Caution: Bei Verwendung von `SET CONTROL 'T=`*`xxxx`*`'` oder `SET CONTROL '+'` oder bei aktivierter Personalcomputer-Unterstützung (Profilparameter `PC` auf `ON`) wird die Natural-

Systemvariable `*DEVICE` verändert, so dass sie nicht mehr zur Bestimmung einer asynchronen Natural-Sitzung verwendet werden kann.

Beachten Sie, dass einige Parametereinstellungen für asynchrone Natural-Sitzungen erzwungen werden können, indem im Makro `NTCICSP` der Subparameter `RCVASYN` auf `ON` gesetzt wird.

Testen und Debuggen

Neuere CICS-Versionen bieten die Transaktion CEDX, die das Debugging von asynchronen Tasks in CICS ermöglicht. In früheren CICS-Versionen war ein solches Debugging nur mit Terminal-Tasks möglich.

Wenn Sie asynchrone Natural-Sitzungen ohne CEDX testen wollen, müssen Sie die asynchrone Natural-Sitzung von einem Terminal aus starten und entweder `ASYN` als allererste fünf Zeichen im dynamischen Parameterstring oder den Profilparameter `TTYPE=ASYN` oder `TTYPE=ASYL` angeben. Die Natural CICS-Schnittstelle baut dann eine asynchrone Natural-Sitzung auf. Beachten Sie, dass Natural die Sitzung zwar als asynchron behandelt, CICS sie aber weiterhin wie eine Terminalsitzung behandelt.

18 Protokollieren von Natural-Sitzungen unter CICS

- Möglichkeiten der Protokollierung 90
- Definition der Natural-Protokolldatei 90
- Natural-Protokollsätze (Log Records) 91

Dieses Kapitel beschreibt, wie Informationen über Natural-Sitzungen in einer Datei protokolliert werden können, die im Batch-Modus verarbeitet und ausgewertet werden kann.

Möglichkeiten der Protokollierung

Optional können Informationen zu Natural-Sitzungen in einer Datei protokolliert werden, die im Batch-Modus verarbeitet und ausgewertet werden kann. Damit lässt sich die Nutzung des Natural-CICS-Systems über einen längeren Zeitraum verfolgen.

Im Gegensatz dazu steht die Online-Protokollierung mit Hilfe des Dienstprogramms SYSTP, das nur eine Momentaufnahme der aktuellen Systemnutzung liefert.

Besondere Aspekte

- Es ist möglich, dass mehrere Natural CICS-Umgebungen (d. h. mehrere Systemverzeichnisse mit eindeutigen Threads, Roll Facilities und Buffer Pools) dasselbe Natural-Protokollziel (Log Destination) verwenden. Wenn eine SCP-Umgebung initialisiert wird, wird eine Systemkennung (System-ID) in das Systemverzeichnis geschrieben. Diese Systemkennung ist Teil eines Auswertungsprogramms zum “Sortieren” von Protokollsätzen nach Natural CICS-Systemumgebung.
- Es wird empfohlen, die Natural-Protokolldatei (Log File) im CICS der Natural-Anwendung zu definieren, da die Protokollierung in einer entfernten (remote) Protokolldatei die Leistung beeinträchtigen würde.
- Wenn Sie das Programm zur Auswertung der Protokolldatei ausführen (siehe *SYSTP im Batch-Modus für CICS-Sitzungen* in der *Dienstprogramme*-Dokumentation), sollte die Protokolldatei im CICS geschlossen werden, da es sonst zu unvorhersehbaren Ergebnissen kommen kann, weil der letzte Puffer noch im Speicher ist oder der EOF-Satz in der Datei fehlt.
- Für die Natural-Protokolldatei sollte ausreichend Plattenplatz reserviert werden. Vorzugsweise sollte die Protokolldatei über eine sekundäre Zuordnung definiert werden.

Definition der Natural-Protokolldatei

Die Natural-Protokolldatei ist eine sequenzielle Plattendatei, d. h. ein “zusätzliches Partitionsziel” im Sinne von CICS. Standardmäßig lautet der interne (logische) Name der Protokolldatei `NLOG`. Dieser Name kann durch Angabe des Subparameters `LOGDEST` im Makro `NTCICSP` geändert werden.

Die Protokolldatei muss in einem CICS-DCT als `TYPE=EXTRA` mit zugehörigen Dataset-Steuerungsinformationen definiert werden (`TYPE=SDSCI`-Eintrag im DCT). Diese Datei muss auch in der CICS-Startup-JCL (DD-Anweisung) definiert werden.

Natural-Protokollsätze (Log Records)

Die folgenden Datensätze werden in der Natural-Protokolldatei protokolliert:

- Natural CICS-Systemneustart-Datensatz
- Natural-Sitzungsende-Datensatz

Natural CICS-Systemneustart-Datensatz

Länge=`96`

Nach erfolgreicher SCP-Systeminitialisierung wird ein Datensatz in die Protokolldatei geschrieben, der das Initialisierungsdatum und die Uhrzeit sowie andere Systemdaten wie das gemeinsame Systempräfix, die Anzahl der RCBs oder die Anzahl der Thread-Gruppen enthält.

Wenn diese erste Protokollanforderung fehlschlägt, wird die Natural-Protokolldatei im Systemverzeichnis als nicht verfügbar gekennzeichnet und es findet keine weitere Protokollierung statt.

System-Neustart-Datensätze werden immer dann geschrieben, wenn die Höchststandmarken des Systems durch die entsprechende Systemverwaltungsfunktion des Dienstprogramms SYSTP zurückgesetzt werden (siehe *Debugger und Dienstprogramme (Utilities)*-Dokumentation). Zusätzlich zu den Systemstartinformationen enthalten diese Sätze die Terminalkennung und die Benutzerkennung des SYSTP-Benutzers.

Natural-Sitzungsende-Datensatz

Länge=`216`

Bei der (normalen oder abnormalen) Beendigung einer Natural-Sitzung wird ein Sitzungsprotokollsatz in die Protokolldatei geschrieben. Dieser Datensatz ist intern in sechs Teile unterteilt:

1. Der Satzsteuerungsteil, der die eigentlichen Sitzungsstatistiken enthält:
   - das aktuelle Datum und die Uhrzeit (d.h. Datum und Uhrzeit der Beendigung der Sitzung),
   - die Systemkennung, die die Natural CICS-Umgebung angibt, in der die Sitzung aktiv war,
   - die Satzart = Session Record.

   Der Satzsteuerungsteil ist bei allen Natural-Log-Sätzen vorhanden, um die verschiedenen Satztypen zu unterscheiden. Das Makro `NCMLOG` enthält die Datensatzlayouts.
2. Der Benutzersitzungsteil, der die eigentlichen Sitzungsstatistiken enthält:
   - die Terminalkennung,
   - die (letzte) Benutzerkennung,
   - das Datum und die Uhrzeit des Sitzungsbeginns,

- den von der Sitzung maximal belegten Speicherplatz,
- die Anzahl der Wiederaufnahmen/Einspeicherungen (Roll Ins) der Sitzung,
- die maximale Anzahl an Datensätzen, die von der Sitzung per Rolling verschoben wurden (falls vorhanden).

3. Der Thread-Gruppen-Teil, der die aktuellen Daten der mit der Sitzung verbundenen Thread-Gruppe enthält:
   - die Thread-Gruppen-Nummer,
   - die Anzahl der TCB-Slots in der Gruppe (falls vorhanden),
   - die gemeinsame Thread-Größe der Gruppe,
   - den von einer Sitzung in der Gruppe maximal belegten Speicherplatz,
   - die maximale Anzahl der in der Gruppe aktiven Sitzungen,
   - die maximale Größe der Warteschlange der Gruppe (bei `TYPE=SHR`-Thread-Gruppen) und die maximale Anzahl der in der Gruppe nebenläufig aktiven Sessions (bei `TYPE=GETM`-Thread-Gruppen),
   - wie oft diese maximale Warteschlangengröße erreicht wurde.
4. Der Thread-Teil, der die Daten des `TYPE=SHR`-Threads enthält, der von der Sitzung als letzter Thread verwendet wurde (falls überhaupt verwendet):
   - den Thread-Namen,
   - die Anzahl der Thread-Nutzungen,
   - der höchste von einer Sitzung verwendete Thread-Speichergröße,
   - die Anzahl der Wiederaufnahmen/Einspeicherungen (Roll Ins) der Sitzung in diesem Thread,
   - die maximale Warteschlangengröße dieses Threads,
   - wie oft diese maximale Warteschlangengröße erreicht wurde.
5. Der Roll Facilty-Teil, der Informationen über die Roll Facilty enthält, der die Sitzung zugewiesen wurde (falls sie überhaupt zugewiesen wurde):
   - den Namen der Roll Facility,
   - die maximale Anzahl der dieser Roll Facility zugewiesen Sitzungen,
   - die Datensatzgröße der Roll Facility
   - die Slot-Größe der Roll Facility,
   - die Anzahl der Slots in dieser Roll Facility,
   - die maximale Anzahl der Aus-/Einspeicherungen (Roll Outs/Roll Ins) bei dieser Roll Facility.
6. Der Teil des Systemverzeichnisses, der Statistiken über die globale Systemnutzung enthält:
   - die maximale Anzahl der UCB-Blockerweiterungen,
   - die maximale Anzahl der im System aktiven Sitzungen,

- die maximale Anzahl von Sitzungen, die gleichzeitig im SCP aktiv sind,
- die Anzahl der SCP-Systemwiederherstellungen.

Konzeptgemäß werden die Datensätze zur Beendigung von Sitzungen nach Datum und Uhrzeit der Sitzung gespeichert. Das bedeutet, dass die Teile 3 bis 6 eines späteren Sitzungsdatensatzes immer aktuellere Informationen enthalten als die Teile eines früheren Datensatzes. Die Teile 3 bis 6 des Datensatzes werden vom Programm zur Auswertung der Protokolldatei verwendet, um die entsprechenden Informationen zu aktualisieren, d.h. Informationen über die Thread-Gruppe, den Thread, die Roll Facility und die SCB.

Diese Vorgehensweise dient dazu, aktuelle Informationen über die Natural CICS-Systemressourcen zu erhalten, falls CICS unkontrolliert beendet wird.

Die Sitzungsbeendigungsprotokolle geben natürlich nur die Ressourcen wieder, die von den entsprechenden Sitzungen verwendet wurden. Daher spiegeln diese Aufzeichnungen möglicherweise nicht die gesamte SCP-Umgebung wider. Berichte über eine vollständige SCP-Umgebung erhalten Sie, wenn Sie einen Snapshot der gesamten Umgebung erstellen, indem Sie entweder die Funktion **System Administration Facilities** des Dienstprogramms SYSTP verwenden (siehe die *Debugger und Dienstprogramme*-Dokumentation) oder Natural unter CICS in die CICS PLTSD stellen (wie im Abschnitt ***Spezielle Natural CICS-Funktionalität*** beschrieben).

System-Snapshot-Sätze in der Natural-Protokolldatei sind Sitzungsbeendigungssätze ohne sitzungsspezifische Informationen, wie sie in Teil 2 aufgeführt sind.

19 Natural CICS Performance-Überlegungen

- Wahl zwischen Roll Server und Roll Facilities 96
- Auswahl der Roll Facility 97
- Shared Storage Threads im Vergleich zu GETMAINed Threads 100
- CICS-Parameter-Einstellungen 102
- Zeilenkompromierungssysteme (Line Compression) 103
- Pseudokonversationelle im Vergleich zu konversationellen Transaktionen 103
- Natural und Adabas 104
- CICS-Monitoring-Produkte 104

Dieses Kapitel enthält Leitlinien für die Einrichtung von Natural in einer CICS-Umgebung.

Wahl zwischen Roll Server und Roll Facilities

Der Roll Server ist vielseitig und skalierbar, wenn er zum Speichern und Wiederherstellen von Natural-Sitzungsdaten über eine Bildschirm-E/A verwendet wird. Sein Einsatz ist in einer CICSPlex- oder Parallel Sysplex-Umgebung unerlässlich, er kann aber auch mit einzelnen CICS-Regionen verwendet werden. Eine große Anzahl gleichzeitig ausgeführter Natural-Sitzungen kann vom Roll Server problemlos bewältigt werden, kann aber bei Verwendung von Roll Facilities zu Ressourcenkonflikten führen.

Da sich ein Roll Server auf ein bestimmtes Subsystem bezieht, können Natural-Installationen in verschiedenen CICS-Regionen denselben Roll Server nutzen, sofern sie dieselbe Subsystemkennung verwenden. Wenn die Terminalkennungen in CICS-Regionen mit Natural-Installationen, die sich denselben Roll Server teilen, in diesen Regionen nicht eindeutig sind, müssen Sie den NTCICSP-Parameter UNITID=ON verwenden, um sicherzustellen, dass eine eindeutige Sitzungskennung erstellt werden kann, oder separate Roll Server für jede CICS-Region mit unterschiedlichen Subsystemkennungen verwenden.

In einer nicht-parallelen Sysplex-Umgebung kann der Roll Server ohne Roll File arbeiten und nur den speicherinternen Local Roll Buffer (LRB) verwenden, der in einem oberhalb der Speichergrenze zugewiesenen z/OS-Speicherobjekt enthalten ist. In diesem Fall sollte der LRB mindestens so viele Slots haben wie die maximale Anzahl der gleichzeitig ausgeführten Natural-Sitzungen. Die Slotgröße sollte mindestens so groß sein, dass ein durchschnittlich komprimierter Natural Thread darin Platz findet. Wenn “Reserve”-Slots verfügbar sind, belegen Threads, die nicht in einen einzigen LRB-Slot passen, zusätzliche Slots.

Wenn Natural for zIIP installiert ist, plant der Roll Server die Thread-Komprimierung für die Ausführung auf einem zIIP.

Weitere Informationen finden Sie unter *Natural Roll Server-Funktionen* in der *Operations*-Dokumentation.

Um den Aufwand für die Einrichtung und Pflege eines Roll Servers für Natural-Installationen und CICS-Regionen mit einer geringeren Anzahl gleichzeitig laufender Natural-Sitzungen zu vermeiden, kann eine der unterstützten Roll Facilities wie unten beschrieben verwendet werden.

Auswahl der Roll Facility

Dieser Abschnitt enthält Informationen, die Ihnen bei der Auswahl der zu verwendenden Roll Facility helfen. Um eine Roll Facility für eine Thread-Gruppe zu spezifizieren, können Sie den Parameter `PRIMERF` des Makros `NCMTGD` verwenden.

Für Natural-Installationen und CICS-Regionen mit einer großen Anzahl gleichzeitig ausgeführter Natural-Sitzungen wird die Verwendung einer Roll Facility nicht empfohlen; in diesem Fall wird die Verwendung des Roll Servers dringend angeraten.

Dieser Abschnitt behandelt die folgenden Themen:

- Verwendung von CICS-Hauptzwischenspeicher
- Verwendung von CICS-Hilfszwischenspeicher
- Kontrollintervall für VSAM Roll Files und Hilfszwischenspeicher
- VSAM Roll Files im Vergleich zu CICS-Zwischenspeicher
- Verwendung von VSAM RRDS Roll Files

Verwendung von CICS-Hauptzwischenspeicher

Wenn der CICS-Hauptzwischenspeicher als Roll Facility angegeben ist (`NCMTGD`-Parameter `PRIMERF=MAIN`), ist beim Aus-/Einspeicherungsvorgang keine VSAM-E/A-Aktivität oder Kommunikation mit einem Zwischenspeicherserver erforderlich; daher ist die Verwendung von `PRIMERF=MAIN` schneller als `PRIMERF=AUX`. Aufgrund der Menge des verwendeten Hauptspeichers kann eine gewisse Optimierung erforderlich sein.

Der Hauptzwischenspeicher wird in einem 64-Bit-Speicher (oberhalb der Speichergrenze) zugewiesen, der durch den CICS-Systeminitialisierungsparameter `TSMAINLIMIT` und den z/OS-Parameter `MEMLIMIT` gesteuert wird. Die auszulagernden Thread-Daten werden in Abschnitte (Chunks) mit einer maximalen Größe von 32 KB aufgeteilt.

Wenn Sie jedoch zusätzlich zu `PRIMERF=MAIN` mit dem Makro `NTCICSP` den Parameter `MEMOBJR=ON` (Standardeinstellung) angeben, werden die auszuspeichernden Thread-Daten in einem einzigen Vorgang als zusammenhängendes Speicherstück in den gemeinsamen DSA-Speicher (GSDSA) oberhalb der Grenze kopiert.

Verwendung von CICS-Hilfszwischenspeicher

Wenn die Natural CICS-Schnittstelle einen Hilfszwischenspeicher als Roll Facility verwendet (NCMTGD-Parameter PRIMERF=AUX), muss die Größe des Kontrollintervalls für die verwendete Datei mindestens 4 KB betragen. Wenn kein Hilfszwischenspeicher verfügbar ist, wird stattdessen der Hauptzwischenspeicher verwendet.

Wenn Sie VSAM Roll Files verwenden (NCMTGD-Parameter PRIMERF=VSAM) und die definierten VSAM Roll Files während einer CICS-Sitzung voll werden oder nicht mehr nutzbar sind, verwendet die Natural CICS-Schnittstelle einen Hilfszwischenspeicher.

Die Anzahl der VSAM-Pufferspeicher, die durch den CICS-Systeminitialisierungsparameter TS definiert ist, sollte auf einen angemessenen Wert erhöht werden, um die Anzahl der physischen E/A-Operationen zu verringern. Die CICS-Statistiken sollten auf Engpässe in diesem Bereich überprüft werden.

Kontrollintervall für VSAM Roll Files und Hilfszwischenspeicher

Um die Anzahl der E/A-Operationen und den für die Durchführung des Aus-/Einspeicherungsvorgangs erforderlichen CPU-Mehrverbrauch zu minimieren, sollten Sie die höchstmögliche Kontrollintervallgröße von 32 KB angeben, wenn Sie VSAM-Dateien oder den Hilfszwischenspeicher als Auslagerungsmöglichkeit (Roll Facility) verwenden (PRIMERF=VSAM bzw. PRIMERF=AUX).

Gründe für eine Kontrollintervallgröße von weniger als 32 KB können die bessere Ausnutzung von Festplattenspuren oder die Verwendung von virtuellem Speicher für die VSAM-Pufferspeicher sein.

VSAM Roll Files im Vergleich zu CICS-Zwischenspeicher

Bei gleicher CISIZE/Datensatzgröße verursacht der Zwischenspeicher weniger CPU-Overhead als VSAM Roll Files:

Um *n* Datensätze in den Zwischenspeicher zu schreiben, müssen Sie *n* +1 CICS-Anfragen stellen (d. h. 1 für DELETQ und n für PUTQ), während Sie für VSAM Roll Files aufgrund der VSAM-Transaktionslogik 2 *n* Anfragen stellen müssen: *n* mal (READ für UPDATE plus REWRITE).

Bei VSAM-Update-Anforderungen wird immer eine physische Ein-/Ausgabe durchgeführt, während bei Zwischenspeichersätzen (AUX) Pufferspeicherung erfolgt, so dass sich in vielen Fällen die zu lesenden Sätze noch in den Pufferspeichern befinden.

Der CICS-Zwischenspeicher kann jedoch zu einem Engpass werden, wenn er auch von anderen Anwendungen genutzt wird.

Durch VSAM Roll Files für Natural kann dieser Engpass überwunden werden (wenn auch auf Kosten von zusätzlichem VSAM-Pufferspeicherplatz), insbesondere wenn E/A-Konflikte vermieden werden können. VSAM Roll Files mit optimaler/maximaler CISIZE/Datensatzgröße sind besonders

effizient, wenn diese Datensatzgröße für die CICS-Zwischenspeicher aufgrund anderer Erfordernisse nicht angegeben werden kann.

CICS-Zwischenspeicher sollte immer dann verwendet werden, wenn er für Natural genutzt werden kann. Wenn der CICS-Zwischenspeicher auch von anderen Anwendungen genutzt wird, sollten Sie prüfen, ob die Natural-Performance bei Verwendung von VSAM Roll Files besser ist.

Wenn Natural mit CICS-Zwischenspeicher nicht schlechter abschneidet, sollten Sie den CICS-Zwischenspeicher als Roll Facility wählen und den "eingesparten" VSAM Roll File-Pufferspeicherplatz für weitere TS-Puffer oder für einen zusätzlichen Thread verwenden.

Verwendung von VSAM RRDS Roll Files

VSAM Roll Files sollten für die normale CICS VSAM File-Optimierung in Betracht gezogen werden, z.B. `BUFNO`- und `STRNO`-Parameter in der FCT. Die CICS-Shutdown-Statistiken sollten auf Engpässe in diesem Bereich überprüft werden.

Da die Roll Files als eine Art Page Dataset für Natural dienen, sollte alles vermieden werden, was das Natural-Aus-/Einspeichern verlangsamt, da es Journaling und Logging gibt. Dynamic Transaction Backout (DTB) und Forward Recovery für Roll Files sind nutzlos und verursachen nur Overhead.

In MRO-Umgebungen

Aus Performance-Gründen sollten die VSAM Roll Files in demselben CICS-System definiert werden, in dem auch die Natural-Anwendungen laufen. Das MRO Function Shipping sollte nicht aufgerufen werden. CICS Local Shared Resources (LSR) können verwendet werden, wenn genügend Puffer vorhanden sind.

Separater LSR-Pool für Natural

Es wird empfohlen, einen separaten LSR-Pool für Natural Roll Files zu definieren, wobei die Anzahl der Strings (`STRNO`) größer sein sollte als die Anzahl der Threads. Auch die Anzahl der Puffer sollte größer sein als die Anzahl der Threads. Eine größere Anzahl von Puffern erhöht die Look-Aside-Trefferquote.

Shared Storage Threads im Vergleich zu GETMAINed Threads

Dieser Abschnitt behandelt die folgenden Themen:

- Speichernutzung
- Kontrolle über die Speichernutzung
- Aus-/Einspeichern - Rolling
- Überlegungen zu CICS/TS
- Schlussbemerkung

Speichernutzung

Shared-Storage-Threads werden bei der Initialisierung des Natural-CICS-Systems vorab zugewiesen. Das bedeutet, dass der von diesen Threads belegte Speicher dem Natural-CICS-System vorbehalten ist, unabhängig davon, ob es aktive Sitzungen gibt oder nicht. GETMAINed-Threads hingegen existieren nur, solange die CICS-Task aktiv ist.

Kontrolle über die Speichernutzung

Bei Shared-Storage-Threads (`TYPE=SHR`) verwendet Natural unter CICS immer den Speicher, der bei der Initialisierung von Natural vorab zugewiesen wurde. Daher ist die Größe des von Natural-Threads verwendeten Speichers leicht vorhersehbar. Bei GETMAINed-Threads (`TYPE=GETM`) hängt der tatsächlich verwendete Speicherplatz jedoch von der Anzahl der gerade aktiven Natural-Sitzungen ab.

Obwohl Natural selbst keinen Steuerungsmechanismus für die maximale Anzahl von GETMAINed-Threads hat, kann dies durch die Gruppierung der Natural-Transaktionscodes in einer `TRANCLASS` gesteuert werden. Wenn ein Transaktionscode zu einer solchen Klasse gehört, kann die maximale Anzahl der parallelen Tasks durch den Parameter `MAXACTIVE` in der `TRANCLASS`-Definition geregelt werden.

Aus-/Einspeichern - Rolling

Wenn eine Natural-Sitzung ihren gemeinsam genutzten Speicher-Thread freigibt, werden die Sitzungsdaten in dem Thread in unkomprimiertem Format aufbewahrt, es sei denn, eine andere Sitzung muss diesen bestimmten Thread verwenden. In diesem Fall ist die neue Sitzung für die Speicherung der Daten der alten Sitzung verantwortlich.

Ein solcher Vorgang wird als verzögertes Aus-/Einspeichern (Deferred Rolling) bezeichnet. Er ermöglicht es Ihnen, ein Aus-/Einspeichern vollständig zu vermeiden, vorausgesetzt, die Anzahl der verfügbaren Threads ist größer oder gleich der Anzahl der nebenläufig aktiven Natural-Sitzungen.

Umgekehrt speichern Sitzungen, die GETMAINed-Threads verwenden, ihre Daten immer vor der FREEMAIN-Operation bei Beendigung der CICS-Task, was unabhängig von der Anzahl der nebenläufig aktiven Natural-Sitzungen zu einem Roll-Overhead führt.

In Umgebungen mit hohem Natural-Transaktionsaufkommen gibt es praktisch keinen Unterschied zwischen dem Sichern von Sitzungsdaten über die "sofortige" oder die "verzögerte" Aus-/Einspeichern-Methode.

In stark ausgelasteten Natural-Umgebungen mit einem hohen Verhältnis von Natural-Sitzungen zu Programmspeicher-Threads entsteht ein höherer Aus-/Einspeichern-Aufwand, da diese Threads von mehreren Natural-Sitzungen gemeinsam genutzt werden. Ein potenzielles Problem in dieser Situation ist die Thread-Konkurrenz (Thread Contention), die durch Natural-Tasks mit lange laufenden Adabas-Anfragen verursacht wird, d. h. mit vielen Adabas-Aufrufen.

Um zu verhindern, dass solche Tasks einen Thread zu lange blockieren, kann man sie durch entsprechende Einstellung des Natural-Profilparameters DBROLL dazu zwingen, ihren Thread freizugeben.

Bei GETMAINed-Threads kommt es jedoch nie zu Konflikten zwischen zwei oder mehreren Natural-Sitzungen, da ein TYPE=GETM-Thread ausschließlich zu der Natural-Sitzung gehört, der er zugeordnet wurde.

TYPE=GETM-Threads können daher als "Einweg-Ressourcen" betrachtet werden, die nie gemeinsam genutzt werden, während TYPE=SHR-Threads als "Mehrweg-Ressourcen" betrachtet werden können, die gemeinsam genutzt werden können.

Überlegungen zu CICS/TS

Das wichtigste Funktionsmerkmal von CICS/TS ist die Transaktionsisolierung, d. h. der Speicher einer Task kann vor anderen Tasks geschützt werden.

Um diese Möglichkeit bei Natural zu nutzen, sollten TYPE=GETM-Threads verwendet werden, da diese Threads der Transaktionsisolierung unterliegen, während dies bei TYPE=SHR-Threads ("shared") nicht der Fall ist. Außerdem entsteht bei TYPE=SHR-Threads mit CICS/TS zusätzlicher CICS-Overhead.

Während der Thread-Auswahlalgorithmus für TYPE=GETM-Threads sehr einfach ist (wenn eine Natural-Task gestartet wird, wird über CICS GETMAIN ein Thread zugewiesen), ist er für TYPE=SHR-Threads komplizierter: Die Umgebung der Natural-Threads wird vom umgebungsabhängigen Nukleus verwaltet (Warteschlangenbildung und Lastausgleich), während CICS nichts über Natural-Threads weiss. Im Gegensatz zu TYPE=GETM-Threads, bei denen CICS den Thread spätestens am Ende der Task freigeben würde, muss bei TYPE=SHR-Threads Natural die Threads ihren Sitzungen zuordnen/freigeben. Zu diesem Zweck führt Natural eine Liste von Thread Control Blocks (TCBs).

Obwohl Natural immer einen Exit aktiv hält, um im Falle einer abnormalen Task-Beendigung Sitzungsressourcen freigeben zu können, die CICS unbekannt sind (z. B. TYPE=SHR-Threads),

können Situationen auftreten, in denen eine Natural-Task beendet wird, ohne dass ihr Thread in der zugehörigen TCB als frei markiert wird, beispielsweise wenn eine `EXEC CICS ABEND CANCEL`-Anforderung in einem von Natural aufgerufenen Nicht-Natural-Programm ausgegeben wurde oder wenn Natural-Sitzungen durch `KILL`-Transaktionen eines Performance-Monitors geleert wurden (Flushing).

Um Probleme mit versehentlich belegt gebliebenen Threads zu vermeiden, prüft Natural unter CICS in seinem Thread-Auswahlalgorithmus immer, ob die mit einem belegten Thread verknüpfte CICS-Task noch vorhanden ist. Wenn dies nicht der Fall ist, wird der Thread freigegeben.

Bei CICS-Versionen vor CICS/TS erfolgte diese Prüfung auf aktive CICS-Tasks durch Kontrollblocksprünge, d. h. Natural prüfte auf eine aktive Task, indem es die Konsistenz der Kontrollblöcke EISTG, TCA und TQE der Task prüfte. Bei CICS/TS ist der Speicher einer anderen Task aufgrund der Transaktionsisolierung möglicherweise überhaupt nicht zugänglich.

Um diese Funktion in CICS/TS zu erfüllen, gibt der umgebungsabhängige Nukleus eine `EXEC CICS INQUIRE STORAGE TASK()`-Anforderung für jeden Thread aus, der in der Thread-Auswahlroutine als belegt identifiziert wurde. Das bedeutet, dass es einige CICS-Anforderungen gegeben haben kann, bevor die Task schließlich für Thread-Ressourcen in die Warteschlange gestellt wird. Dasselbe CICS-Kommando wird auch für die Serialisierung von Natural-Sitzungen verwendet (z. B. verzögertes Aus-/Einspeichern (Deferred Rolling) von `TYPE=SHR`-Threads).

Schlussbemerkung

Sowohl `TYPE=SHR`- als auch `TYPE=GETM`-Threads haben ihre Vor- und Nachteile. Bei CICS/TS sind `TYPE=GETM`-Threads jedoch zu bevorzugen, und zwar aufgrund folgender Aspekte:

- Unterstützung der Transaktionsisolierung,
- mehr CICS-ähnliche Optimierungsmöglichkeiten,
- schlechtere Performance der `TYPE=SHR`-Threads.

CICS-Parameter-Einstellungen

Die CICS-SIT-Parameter `AMXT` und `CMXT` sollten verwendet werden, um die Anzahl der nebenläufigen Natural-Tasks zu kontrollieren.

Die angegebene Anzahl sollte größer sein als die Anzahl der Threads. Es sollte auch in Betracht gezogen werden, für asynchrone Natural-Tasks und für Natural Advanced Facilities Spool-Tasks eine eigene Transaktionsklasse mit einem geeigneten `CMXT`-Parameter anzugeben, um zu verhindern, dass “normale” Natural-Terminal-Tasks durch zu viele solcher “Hintergrund”-Tasks, die Threads belegen, abgemeldet werden. Für diese Transaktionen können spezielle Thread-Gruppen definiert werden.

CICS-Dumps für Natural-Transaktionen sollten unterdrückt werden, es sei denn, sie werden vom Support zu Debugging-Zwecken angefordert. Natural selbst erzeugt Dumps (über `EXEC CICS DUMP`) bei Nicht-Programmprüfungsabbrüchen und auch bei Programmprüfungen, wenn der Natural-Session-Parameter `DU` auf `ON` gesetzt ist. Wenn kein Natural-Dump erzeugt wird, ist ein CICS-Dump überflüssig und verursacht nur zusätzlichen Overhead (insbesondere bei der Erstellung eines System/Region-Dumps, da das gesamte CICS-System angehalten wird, bis der Snap-Dump abgeschlossen ist).

Der CICS-Trace ist für die Problemanalyse unerlässlich, obwohl er Mehraufwand für das System bedeutet. Auch CICS-Performance-Monitoring-Tools und Accounting Packages verursachen einen System-Overhead von bis zu 30 Prozent und mehr. Einige Packages schalten intern den CICS-Trace ein und fangen ihn dann ab. Sie sollten sich dieses potenziellen System-Mehraufwands bewusst sein. Denken Sie auch daran, dass die Natural CICS-Schnittstelle die Anwendungsprogrammierschnittstelle der CICS-Kommandoebene verwendet: Anfragen auf CICS-Kommandoebene erzeugen viel mehr Trace-Einträge (abgesehen von sonstigem Overhead) als Anfragen auf CICS-Makroebene.

Zeilenkompromierungssysteme (Line Compression)

Natural selbst optimiert seine Datenströme mit Hilfe von RA (Repeat to Address) und anderen Techniken wie Screen Imaging usw. Wenn andere Line Zeilenkompromierungssysteme installiert sind, sollten die Natural-Transaktionen von der Verarbeitung durch diese Systeme ausgeschlossen werden, da dies zu einem Mehraufwand führen würde, der keinen Nutzen bringt.

Pseudokonversationelle im Vergleich zu konversationellen Transaktionen

Bei der Wiederaufnahme einer Sitzung sind konversationelle Natural-Tasks an ihren ursprünglichen Thread gebunden, was bedeutet, dass eine konversationelle Task auf diesen Thread warten muss, wenn er gerade nicht verfügbar ist. Pseudokonversationelle Natural-Tasks hingegen sind flexibel und können in jeden verfügbaren Thread verschoben werden.

Mit anderen Worten: Der “klassische” Vorteil von konversationellen Tasks - weniger Ein-/Ausgaben für das Speichern/Wiederherstellen von Daten gegenüber Bildschirm-E/A-Operationen - gilt für Natural aufgrund seiner Thread-Technik nicht.

Natural und Adabas

Da eine Natural-Task in CICS auf den Abschluss eines Adabas-Aufrufs wartet, sollte die versorgende Adabas-Region/Partition immer eine höhere Priorität haben als die CICS-Region/Partition, um die Wartezeit zu minimieren.

CICS-Monitoring-Produkte

CICS-Monitoring-Produkte können Funktionen zum Löschen von CICS-Tasks bieten, wobei alle von der Anwendung gesetzten Exits für abnormale Beendigung umgangen werden.

Caution: Solche Funktionen sollten nicht zum Abbrechen von Natural-Tasks verwendet werden, da Natural möglicherweise nicht in der Lage ist, die von ihm belegten Ressourcen zu bereinigen, und, schlimmer noch, das Natural-CICS-System in einem inkonsistenten Zustand verbleiben kann, je nachdem, womit diese Task beschäftigt war.

Um Natural-Sitzungen abzubrechen, sollten Sie stattdessen die Funktionen **Cancel Session** oder **Flush Session** des Dienstprogramms SYSTP verwenden. Siehe *Debugger und Dienstprogramme (Utilities)*-Dokumentation.

20 Natural-Druck- und Arbeitsdateien unter CICS

- Angepasste Nutzung von Druck- und Arbeitsdateien 106
- CICS-Zwischenspeicher für Druck- und Arbeitsdateien 106
- Transiente CICS-Daten bei Druck- und Arbeitsdateien 107

Dieses Kapitel behandelt die Verwendung von Natural-Druck- und Arbeitsdateien (Print- und Work-Files) unter CICS.

Angepasste Nutzung von Druck- und Arbeitsdateien

Die Natural CICS-Schnittstelle unterstützt Natural-Druck- und Arbeitsdateien in CICS entweder als Warteschlangen für transiente CICS-Daten oder als CICS-Zwischenspeicherwarteschlangen, sowohl für Hilfs- als auch für Hauptspeicher.

Um die Nutzung anzupassen, müssen Sie die folgenden Subparameter in den Profilparametern `PRINT` und `WORK` setzen:

```
AM=CICS, TYPE=TD/AUX/MAIN, DEST=queuename
```

Weitere Informationen finden Sie unter den nachstehenden Links:

- Beschreibung des Profilparameters `WORK` und wie Sie die oben genannten Subparameterwerte setzen, siehe `NTWORK`-Parametermakro.
- Beschreibung des Profilparameters `PRINT` und wie Sie die obigen Subparameterwerte setzen, siehe `NTPRINT`-Parametermakro.

Die Unterstützung der Druckdateien durch die Natural CICS-Schnittstelle wurde zu Zwecken der Nachverfolgung und Protokollierung bereitgestellt. Sie ist nicht für die Bearbeitung von Reports gedacht. Insbesondere werden die Schlüsselwort-Subparameter `PRTY`, `CLASS`, `COPIES` usw. beim Statement `DEFINE PRINTER` nicht verarbeitet.

CICS-Zwischenspeicher für Druck- und Arbeitsdateien

CICS-Zwischenspeicher-Warteschlangen, sowohl Hilfs- als auch Hauptspeicher, für CICS-Druck- und -Arbeitsdateien sind von konzeptbedingt `RECFM=V`-Dateien, die für Ein- und Ausgaben zur Verfügung stehen.

Obwohl es in Natural unter CICS keine ausschließliche Kontrolle einer bestimmten TS-Warteschlange durch eine Natural-Sitzung gibt, können Sie automatisch sitzungs- oder terminalabhängige Druck- oder Arbeitsdateien anlegen, indem Sie die im Makro `NTCICSP` mit dem Subparameter `TERMVAR` definierte Zeichenkette (`&TID` ist der Standardwert) in den Subparameter `DEST` des Profilparameters `PRINT` oder in den Subparameter `DEST` des Profilparameters `WORK` eingeben. Wenn eine solche Zeichenfolge im acht Zeichen langen `DEST`-Subparameter gefunden wird, wird sie durch die tatsächliche Terminalkennung ersetzt.

In einer CICSplex-Umgebung

Wenn Sie in einer CICSplex-Umgebung arbeiten, müssen Natural-Druck- und Arbeitsdateien im CICS-Zwischenspeicher in einer CICS-TST als `TYPE=SHARED` oder `TYPE=REMOTE` definiert werden.

NCI-System-Warteschlangen

In Natural unter CICS kann auf NCI-System-Warteschlangen nicht zugegriffen werden. (NCI-System-Warteschlangen sind TS-Warteschlangen mit einem im Subparameter `TSKEY` des Makros `NCMDIR` definierten Präfix).

Transiente CICS-Daten bei Druck- und Arbeitsdateien

In der CICS-DCT muss eine CICS Transient Data Queue für eine Natural-CICS-Druck- und Arbeitsdatei definiert werden. Bei indirekten Zielen (Destinations) werden die Attribute der *Basis*-Destinationen weitergegeben. Insbesondere bestimmen die Attribute einer *Extra-Partition*-Destination, wie `RECFM` oder `TYPEFLE`, die Natural-Arbeitsdateiattribute.

Bei *Intra-Partition*-Destinationen ist `RECFM=V` voreingestellt und sie sind sowohl für die Eingabe als auch für die Ausgabe verfügbar.

Transiente CICS-Daten bei Druck- und Arbeitsdateien sind gemeinsam genutzte Dateien (“Shared Files”) in dem Sinne, dass mehr als eine Sitzung Ein-/Ausgabe-Operationen gegen eine solche Datei durchführen kann.

III Natural unter Com-plete/SMARTS

21 Natural unter Com-plete/SMARTS

- Treiberparameter für die Natural Com-plete/SMARTS-Schnittstelle 112
- Verwendung der Abend-Exits 112
- Speichernutzung 113
- Unterstützung für Backend-Programme 114
- Unterstützung von Com-plete in Natural-Batch-Läufen 114
- Asynchrone Natural-Verarbeitung unter Com-plete/SMARTS 114
- Aufrufen von Natural aus Benutzerprogrammen 115
- Behandlung von Speicherschutzschlüsseln 115
- Unterstützung für die Behandlung von User Exits während der Sitzungsinitialisierung 116
- Verwendung der SMARTS-Server-Umgebung 116
- Unterstützung für das Recoverable Session Handling von Com-plete 119
- Unterstützung von Natural for zIIP unter Com-plete 119

Dieses Kapitel beschreibt die Funktionalität der Natural Com-plete/SMARTS-Schnittstelle (Produktcode NCF) sowie den Betrieb und die einzelnen Komponenten von Natural in einer Com-plete-Umgebung.

Note: SMARTS ist ein Akronym für "Software AG Multi-Architecture Runtime System". Es stellt eine Laufzeitschicht dar, die es ermöglicht, POSIX-ähnliche Anwendungen auf Großrechner-Betriebssystemen auszuführen. Die Produkte der Software AG kommunizieren mit dem Betriebssystem über die SMARTS-Schicht.

Verwandte Dokumente:

- Einzelheiten zum Produkt Com-plete siehe *Com-plete*-Dokumentation
- *Online-Verarbeitung* in der Natural-*System Architecture*-Dokumentation
- *Natural Com-plete/SMARTS-Schnittstelle auf z/OS installieren* in der *Installation für z/OS*-Dokumentation
- *SYSTP Utility* in der Natural-*Debugger und Dienstprogramme*-Dokumentation
- *Natural unter Com-plete/SMARTS-Benutzerabbruchcodes* in der *Meldungen und Codes*-Dokumentation

Treiberparameter für die Natural Com-plete/SMARTS-Schnittstelle

Informationen zu den Treiberparametern, die für die Natural Com-plete/SMARTS-Schnittstelle zur Verfügung stehen, finden Sie in der Beschreibung des Profilparameters `COMP` oder des Parametermakros `NTCOMP` in der *Parameter-Referenz*-Dokumentation. `COMP` (bzw. `NTCOMP`) besteht aus mehreren Schlüsselwort-Subparametern, deren Verwendung in den folgenden Abschnitten erläutert wird.

Verwendung der Abend-Exits

Die `ABEXIT`-Exits können generell deaktiviert werden, indem der Schlüsselwort-Subparameter `SPIEA` auf `OFF` gesetzt wird.

Der `ABEXIT`-Exit wird während der EOJ-Behandlung von Com-plete bei einer abnormalen Programmabbruchverarbeitung aufgerufen.

Standardmäßig wird ein `OCX`-Abbruch von der Exit-Routine `ABEXIT` interpretiert.

- Bei der Ausführung mit `DU=ON` wird ein Speicher-Dump generiert und die Natural-Sitzung mit der Fehlermeldung NAT9974 ordnungsgemäß beendet.
- Bei der Ausführung mit `DU=SNAP` wird wie bei `DU=ON` ein Speicher-Dump generiert, aber die Sitzung wird nicht beendet.

Note: Nach dem Dump wird die Natural-Sitzung fortgesetzt, als ob sie mit `DU=OFF` laufen würde.

- Bei der Ausführung mit `DU=ABEND` wird ein Speicher-Dump der Natural-Sitzung generiert, ohne die Sitzung zu beenden.
- Bei der Ausführung mit `DU=FORCE` wird die Exit-Routine `ABEXIT` deaktiviert, und es wird ein sofortiger Dump während der Com-plete-Ausführung erstellt.
- Wenn `LE370=ON` im `NTCOMP`-Makro angegeben ist oder dynamisch angegeben wird und der Abbruch auftritt, während ein LE-Programm die Kontrolle hat, werden vom Benutzer geschriebene oder sprachspezifische Condition-Handler ignoriert. Der Abbruch wird von der Exit-Routine `ABEXIT` behandelt, und es tritt die Natural-Fehlermeldung NAT0950 oder NAT9967 auf.

Ist `DU=OFF`, antwortet Natural mit der Fehlermeldung NAT0950, NAT0954, NAT0955 oder NAT0956, und das gesamte Abbruch-PSW und die Register 0 bis 15 sind im IOCB mit dem Offset `x'290'` enthalten.

Speichernutzung

Bei der Sitzungsinitialisierung wird der mit dem Schlüsselwort-Subparameter `NTHSIZE` definierte Speicherplatz je nach Einstellung des Schlüsselwort-Subparameters `THABOVE` als Thread `GETMAIN` oberhalb oder unterhalb der 16-MB-Grenze für die Verwendung durch Natural zugeordnet.

Der Natural-Profilparameter `WPSIZE` bestimmt die Größe der Workpools unterhalb und oberhalb der 16 MB-Grenze. Standardmäßig ist die Größe des unterhalb gelegenen Subpools auf 32 KB eingestellt.

Daher müssen Sie den Natural-Com-plete-Frontteil mit det Com-plete Utility ULIB katalogisieren, RG-Größe = 36 KB oder größer.

Die verbleibenden Bereiche innerhalb der Com-plete-Thread-Teile unterhalb und/oder oberhalb (Com-plete ULIB `RG=`*`specification`* and/or `THABOVESIZE=`*`specification`*) werden von Com-plete für Folgendes verwendet:

- Benutzer-Subroutinen,
- Vergrößerung von variablen Puffern innerhalb des Natural-Threads,
- Unterprodukte, die “physische” `GETMAIN`-Anforderungen ausführen. Dadurch wird die Zuordnung des Natural-Workpool erzwungen.

Weitere Einzelheiten zu den Parametern `ULIB RG` und `THABOVESIZE` finden Sie in der *Com-plete Utilities*-Dokumentation.

Unterstützung für Backend-Programme

Natural übergibt den folgenden String an ein Backend-Programm:

- den Natural-Rückgabecode (Vollwort),
- die Natural-Beendigungsmeldung (A72),
- die Länge des Beendigungsbereichs (Vollwort),
- die Beendigungsdaten.

Dieser String wird durch das Makro `NAMBCKP` gemapped.

Das Quellcode-Modul `XNCFBACK` ist ein Beispiel für ein Natural-Backend-Programm in einer Com-plete-Umgebung. Es ist als reentranter Code geschrieben und kann als `RESIDENTPAGE`-Programm oder einmal pro Benutzer geladen werden.

Unterstützung von Com-plete in Natural-Batch-Läufen

Wenn Sie die Com-plete-Dienste in einem Natural-Batch-Lauf verwenden, bleibt die Batch-Benutzerkennung am Ende des Batch-Laufs angemeldet.

Um diese Situation zu vermeiden, müssen Sie das Modul `COMPBTCH` aus der Com-plete Distribution Library in den Natural-Batch-Nukleus aufnehmen. Dadurch wird der Einstiegspunkt für das Modul `EOJ` aufgelöst, das am Ende des Natural-Batch-Jobs zur Beendigungsbereinigung aufgerufen wird.

Das Modul `NCFAM` wird für den Zugriff auf Com-plete-Druck-/Arbeitsdateien verwendet. Er muss beim Verlinken des Natural-Nukleus zusammen mit dem Modul `COMPBTCH` aus der Com-plete Distribution Library eingebunden werden.

Asynchrone Natural-Verarbeitung unter Com-plete/SMARTS

Die asynchrone Natural-Verarbeitung wird im Kapitel *Asynchrone Verarbeitung* in der Natural-*Operations*-Dokumentation behandelt. Allerdings gelten einige zusätzliche Überlegungen, wenn Natural unter Com-plete ausgeführt wird.

Stellen Sie sicher, dass für eine asynchrone Natural-Sitzung geeignete `SENDER`- und `OUTDEST`-Ziele (Destinations) angegeben sind. Andernfalls führt jede Ausgabe zu einer abnormalen Beendigung.

Zusätzlich zu den Com-plete-Terminalkennungen für `SENDER` und `OUTDEST` werden die folgenden Schlüsselwörter von der Natural Com-plete/SMARTS-Schnittstelle unterstützt:

Profilparameter	Mögliche Werte	Erläuterung
SENDER	DUMMY	Ausgabe wird ignoriert.
	TID	Ausgabeziel.
	LUNAME	
OUTDEST	DUMMY	Ausgabe wird ignoriert.
	CONSOLE	Ausgabeziel.
	LUNAME	

Standardmäßig wird für die Ausgabe einer asynchronen Natural-Sitzung, die unter Com-plete läuft, das 3270-Datenstrom-Protokoll verwendet.

Ein Beispiel zum Starten einer asynchronen Natural-Transaktion unter Com-plete finden Sie in der Library SYSEXTP, Programm ASYNCOMP.

Aufrufen von Natural aus Benutzerprogrammen

Mit der Com-plete-Funktion FETCH können Sie Natural aus einem Benutzer-Frontend-Programm unter Com-plete aufrufen. Einzelheiten finden Sie im *Com-plete Application Programmer's Manual*.

Behandlung von Speicherschutzschlüsseln

Wenn Sie den Schutzmodus zwischen Com-plete und Ihrem Anwendungsprogramm verwenden möchten, müssen Sie den Natural-Profilparameter SKEY im Natural-Parametermodul auf OFF setzen. Das Anwendungsprogramm läuft im entsprechenden Thread Key

Sie können die Performance des Anwendungsprogramms unter Com-plete verbessern, indem Sie die Funktion **Storage-Protection Override** auf Ihrer Maschine aktivieren.

Setzen Sie den Thread Key im Com-plete-Startparameter THREAD-GROUP für Ihre Natural-Untergruppe auf 9.

Der Frontend-Treiber versetzt die Natural-Anwendung automatisch in den privilegierten Modus, wenn der Thread Key 9 ist, und verwendet die SPKA-Anweisung für die Schlüsselumschaltung anstelle der Com-plete-Funktion MODIFY mit den Funktionscodes THRD/TCS.

Die Speicherschlüsselumschaltung wird bei keinem Natural- oder Editor-Buffer-Pool-Aufruf durchgeführt.

Unterstützung für die Behandlung von User Exits während der Sitzungsinitialisierung

Während der Sitzungsinitialisierung ist es möglich, benutzerspezifische Sitzungsinformationen über die Aktivierung eines User Exit an Natural zu übergeben. Der Exit wird aufgerufen, bevor Natural initialisiert wurde, nachdem die Treiber/IOCB-Initialisierung abgeschlossen ist.

Der Treiber übergibt als Parameter die Adresse des IOCB in Register 1, wohingegen der Exit durch die Com-plete-Funktionen `COLOAD/CODEL` aktiviert/deaktiviert wird. Einzelheiten finden Sie im *Com-plete Application Programmer's Manual*.

Das Quellcode-Modul `NCFUEXIT` ist ein Beispiel für einen User Exit. Das User Exit-Modul kann mit dem Schlüsselwort-Subparameter `EXIT` definiert werden.

Verwendung der SMARTS-Server-Umgebung

Mit der SMARTS-Server-Umgebung ist es möglich, das portable Dateisystem von SMARTS als Container für Ein- und Ausgabedateien sowie für Datensätze auf dem nativen Dateisystem zu verwenden. Es hängt von der Einstellung der SMARTS-Parameter `CDI_DRIVER und MOUNT_FS` ab, ob sich die Umgebungsvariable auf das portable Dateisystem oder auf ein natives Dateisystem bezieht. Weitere Informationen finden Sie im *SMARTS Installation and Operations Manual*.

Wenn keine Umgebungsvariablen definiert sind, wird auf die normalen Datasets zugegriffen, wie im Abschnitt *Von Natural im z/OS-Batch-Modus verwendete Datasets* in der Natural *Operations*-Dokumentation.

Die folgenden Themen werden behandelt:

- Eingabe/Ausgabe
- Druckdatei/Arbeitsdatei (Print File/Work File)

Eingabe/Ausgabe

Die Ein-/Ausgabe in der SMARTS-Server-Umgebung wird von der DLL `NCF92IO` durchgeführt.

`NCF92IO` muss in den residenten Bereich geladen werden. Wenn `NCF92IO` in den Thread des Anwendungsprogramms geladen wird, wird die Natural-Sitzung mit der Fehlermeldung NAT9980 beendet.

Unterstützte Umgebungsvariablen:

- `CMPRINT` **- Primäre Reportausgabe**

- CMSYNIN - Primäre Kommandoeingabe
- CMOBJIN - Eingabe für Natural-INPUT-Statements

Diese Umgebungsvariablen werden im Folgenden beschrieben.

CMPRINT - Primäre Reportausgabe

Syntax:

```
CMPRINT=/pathname/filename[/],[mode]
```

Bedeutung:

pathname	**Pfadname** Gibt den Speicherort der Ausgabedatei an. Bezieht sich *pathname* auf ein portables Dateisystem, wird der Pfad erstellt. Bezieht er sich auf einen nativen Dataset, muss dieser verfügbar sein.
filename	**Dateiname** Gibt den Namen der Ausgabedatei an. Ein Stern (*) als Dateiname bedeutet, dass der Dateiname aus der aktuellen Benutzerkennung generiert wird. Wenn *pathname* auf das native Dateisystem verweist und *filename* mit dem Schrägstrich (/) abgeschlossen ist, wird auf den sequenziellen Dataset *pathname/filename* zugegriffen. Wenn er nicht mit "/" abgeschlossen ist, wird auf das Member *filename* im Dataset *pathname* zugegriffen.
mode	**Modus** Gibt den Dateimodus an, wie er in der C-Bibliothek für die Funktion `fopen()` dokumentiert ist. Die Standardeinstellung ist `w` (write).

Beispiel: Angenommen, `/fs/` ist auf das native Dateisystem und `/pfs/` auf ein portables Dateisystem gemapped.

`CMPRINT=/fs/natural/test/print`	Es wird auf das Member `print` im Dataset `natural.test` zugegriffen.
`CMPRINT=/fs/natural/test/print/`	Es wird auf den sequenziellen Dataset `natural.test.print` zugegriffen.
`CMPRINT=/pfs/natural/test/print`	Es wird auf das Member `print` in `/natural/test` des portablen Dateisystems zugegriffen.

CMSYNIN - Primäre Kommandoeingabe

Syntax:

```
CMSYNIN=/pathname/filename[/]
```

Gibt den Pfadnamen *pathname* und den Dateinamen *filename* der entsprechenden Kommandoeingabedatei an.

Wenn *pathname* auf das native Dateisystem verweist und *filename* mit dem Zeichen "/" abgeschlossen ist, wird auf den sequenziellen Dataset *pathname/filename* zugegriffen. Wenn er nicht mit einem Schrägstrich (/) abgeschlossen ist, wird auf das Member *filename* im Dataset *pathname* zugegriffen.

CMOBJIN - Eingabe für Natural-INPUT-Statements

Syntax:

```
CMOBJIN=/pathname/filename[/]
```

Gibt den Pfadnamen *pathname* und den Dateinamen *filename* der entsprechenden Dateneingabedatei an.

Wenn *pathname* auf das native Dateisystem verweist und *filename* mit dem Zeichen "/" abgeschlossen ist, wird auf den sequenziellen Dataset *pathname/filename* zugegriffen. Wenn er nicht mit einem Schrägstrich (/) abgeschlossen ist, wird auf das Member *filename* im Dataset *pathname* zugegriffen.

Druckdatei/Arbeitsdatei (Print File/Work File)

Der Zugriff auf Druck- und Arbeitsdateien in der SMARTS-Serverumgebung wird von der DLL `NCF92APS` durchgeführt.

`NCF92APS` muss in den residenten Bereich geladen werden. Wenn `NCF92APS` in den Thread des Anwendungsprogramms geladen wird, wird die Natural-Sitzung mit der Fehlermeldung NAT9980 beendet.

Unterstützte Umgebungsvariablen:

- `NAT_PRINT_ROOT` - Pfad zu den Druckerdateien in einem PFS oder nativen Dateisystem.
- `NAT_WORK_ROOT` - Pfad zu den Arbeitsdateien in einem PFS oder nativen Dateisystem.

Syntax-Beispiel:

```
NAT_WORK_ROOT=/qualifier/path1/path2
```

Bedeutung:

qualifier	**Kennzeichner** Bestimmt, ob auf ein portables SMARTS-Dateisystem oder ein natives, vom Betriebssystem verwaltetes Dateisystem zugegriffen wird.
path1/path2	**Pfad** Ist der Pfad zum Speicherort der Datei im entsprechenden Dateisystem.

Unterstützung für das Recoverable Session Handling von Com-plete

Um das unter z/OS verfügbare Recoverable Session Handling von Com-plete zu nutzen, müssen Sie das Modul `NCFROLLS` mit Ihrem Frontend-Modul verlinken. `NCFROLLS` dient als Schnittstelle zum Natural Roll Server, der gestartet werden muss, um wiederherstellbare Sitzungen zu unterstützen.

Außerdem muss das Modul `ATRRCSS` nicht mit Ihrem Frontend-Modul verlinkt werden, da das RRS-Schnittstellenmodul Teil der Com-plete-Routine `TLOPUSER` ist. Wenn eine konversationelle Terminal-E/A durchgeführt werden soll, wird der Natural Thread in komprimierter Form in das Natural Roll File geschrieben, um die Wiederaufnahme der Natural-Sitzung nach einem Com-plete-Neustart zu ermöglichen. Bei nicht-konversationellen Terminal-Ein-/Ausgaben und bei Anwendungen mit gesperrtem Thread wird der Natural Thread nicht in das Natural Roll File geschrieben. Folglich können solche Sitzungen nicht wiederhergestellt werden.

Unterstützung von Natural for zIIP unter Com-plete

Das Natural Com-plete/SMARTS Interface unterstützt Natural for zIIP unter Com-plete, wenn die in der *Natural for zIIP*-Dokumentation beschriebenen Voraussetzungen erfüllt sind.

Unterstützung von zIIP-fähigem Natural Roll Server

Das Natural Com-plete/SMARTS Interface unterstützt den Natural Roll Server. Um diese Funktion zu nutzen, müssen Natural for zIIP unter Com-plete und der Natural Roll Server NATRSM*vr* verwendet werden. In diesem Fall wird die Komprimierung und Dekomprimierung des Natural Threads auf den IBM System z Integrated Information Processors (zIIPs) ausgeführt, wodurch die CPU-Belastung der General Central Processors (GCPs) reduziert wird.

Um die Verwendung des Natural Roll Servers zu aktivieren, brauchen Sie nur den Natural Roll Server `NATRSM`*vr* zu starten.

Note: Die Notation *vr* steht für die jeweilige Produktversion (siehe auch *Version* im *Glossar*).

IV Natural unter IMS TM

Dieses Kapitel beschreibt die Funktionalität der Natural IMS TM-Schnittstelle (Produktcode NII) sowie den Betrieb und die einzelnen Komponenten von Natural in einer IMS TM-Umgebung.

- **Umgebungen**
- **Komponenten**
- **Konfiguration**
- **Serviceprogramme**
- **Servicemodule**
- **User Exits**
- **Spezielle Funktionen**
- **Wiederherstellungsbehandlung**

Verwandte Themen:

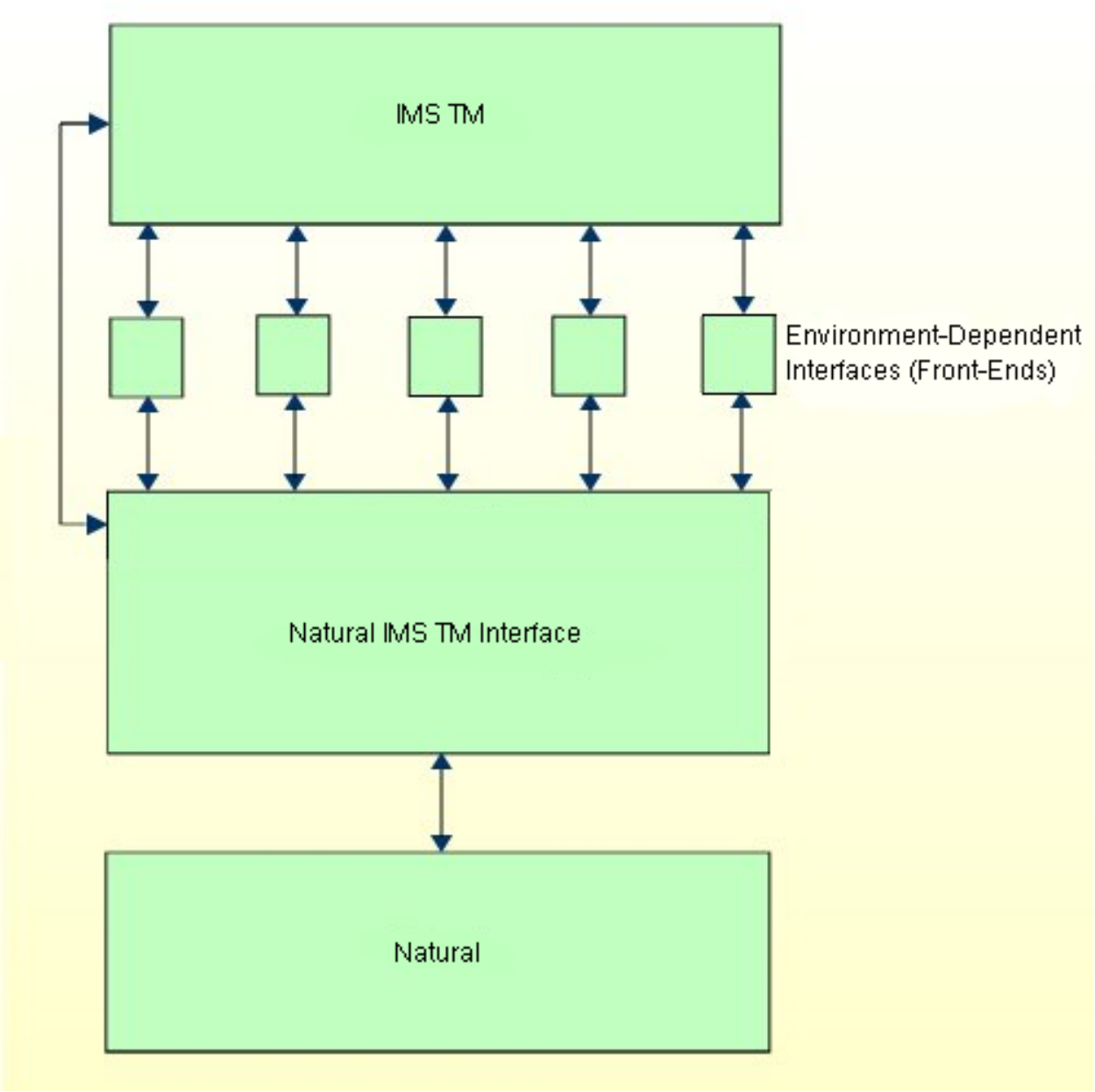
- *Natural IMS TM-Schnittstelle auf z/OS installieren* in der Natural *Installation für z/OS*-Dokumentation.
- *Natural unter IMS TM-Fehlermeldungen* in der *Meldungen und Codes*-Dokumentation

22 Natural unter IMS TM - Umgebungen

- Übersicht über die IMS TM-Schnittstelle 124
- IMS TM-Umgebungen 125
- Dialogorientierte Umgebungen 126
- Nachrichtenorientierte Umgebung 128
- Batch Message Processing-Umgebung 131
- Unterstützung des Natural-Statements WRITE (n) 132
- SET CONTROL 'N' (Terminalkommando %N) 135
- Unterstützung von TS=ON bei Natural unter IMS TM-Nachrichten 135
- SENDER-Ziel 135
- Unterstützung des Natural-Profilparameters PROGRAM 136
- Natural Development Server- bzw. Natural Web I/O Server-Umgebung 136

Dieses Kapitel beschreibt, wie Natural in verschiedenen IMS TM-Umgebungen läuft.

Übersicht über die IMS TM-Schnittstelle



IMS TM-Umgebungen

IMS TM bietet zwei verschiedene Arten von Umgebungen:

- Natural in einer Message Processing Region (MPP-Umgebung)
- Natural in einer Batch-Message-Processing-Region (BMP-Umgebung)

Um Natural in jeder dieser Umgebungen nutzen zu können, werden verschiedene umgebungsspezifische Schnittstellen für die Natural IMS TM-Schnittstelle bereitgestellt. Die Aufgabe einer solchen Schnittstelle besteht darin, Eingaben (in der Regel eine Terminal-Eingabenachricht) aus der Umgebung zu empfangen, die Eingaben zur Verarbeitung an Natural weiterzuleiten und die daraus resultierende Ausgabe an das richtige Ziel (in der Regel eine Terminal-Ausgabenachricht) zurückzuleiten. Auf diese Weise ist es möglich, die Funktionalität von Natural in allen verfügbaren IMS TM-Umgebungen zu nutzen.

Zusätzlich zu den verfügbaren Umgebungen gibt es innerhalb jeder Umgebung auch verschiedene Betriebsarten.

Natural in einer Message Processing Region (MPP-Umgebung)

In einer Message Processing Region können Natural Online-Transaktionen eine der folgenden sein:

- Dialogorientiertes Natural
- Nachrichtenorientiertes Natural

Dialogorientiertes Natural

Eine dialogorientierte Natural-Sitzung stellt eine fortwährende Interaktion mit einem IMS TM-Bildschirm her. Ein- und Ausgabenachrichten an und von Natural stehen in einem logischen Zusammenhang. Über Dialogschritte hinweg speichert Natural Informationen, um die nächste Eingabenachricht korrekt verarbeiten zu können. Natural kann dialogorientiert entweder als konversationelle oder als nicht-konversationelle Transaktion ausgeführt werden.

In einer dialogorientierten Umgebung kann Natural in Multi-Message-Processing-Regionen, als Wait-for-Input (WFI)-Transaktion und mit der Parallel-Scheduling-Option ausgeführt werden.

Um Natural in dialogorientierten Umgebungen auszuführen, müssen Sie entweder den Roll Server oder Roll Files verwenden (siehe ***Das Roll File und der Roll Server***).

Wenn die Natural IMS TM-Schnittstelle eine Fehlersituation feststellt, wird ein Datensatz mit Informationen über diese Fehlersituation in die IMS TM-Protokolldatei geschrieben (siehe ***Wiederherstellungsbehandlung***). Daher müssen alle Terminals, auf denen Natural ausgeführt werden soll, und alle Natural-Transaktionscodes berechtigt sein, den Kommando `/LOG` über das Automated Operator Interface (AOI) auszuführen.

Nachrichtenorientiertes Natural

Eine Nachrichten-orientierte (message-oriented) Natural-Sitzung verarbeitet nicht-3270-formatierte Nachrichten aus der IMS TM Message Queue. Die eingegebenen Nachrichten werden als unabhängig voneinander betrachtet und sind nicht Teil eines Dialogs. Bei einer nachrichtenorientierten Verarbeitung muss Natural als nicht-konversationelle Transaktion ausgeführt werden.

Natural in einer Batch-Message-Processing-Region (BMP-Umgebung)

In einer Batch Message Processing Region kann Natural über einen Input Transaction Code auf die IMS TM Message Queue zugreifen. Bei Batch-orientierten BMP-Regionen unterstützt Natural symbolische Checkpoints und erweiterten Neustart. Bei den Eingabenachrichten handelt es sich um nicht-3270-formatierte Nachrichten.

Dialogorientierte Umgebungen

In diesem Abschnitt werden spezielle Punkte behandelt, die nur für die dialogorientierte Gesprächsumgebung gelten.

- Besondere Aspekte einer dialogorientierten konversationellen Umgebung
- Besondere Aspekte einer nicht-konversationellen Umgebung
- Besondere Aspekte einer MSC-Umgebung

Besondere Aspekte einer dialogorientierten konversationellen Umgebung

Die dialogorientierte konversationelle Umgebung wird durch das Conversational MPP Interface realisiert, das mit dem Natural-Parametermodul zum Conversational MPP Front-End verlinkt ist. Dieses Frontend ist das IMS TM-Anwendungsprogramm. Es wird von IMS TM eingeplant, wenn eine Eingabenachricht für den zugeordneten Transaktionscode in der IMS TM-Nachrichtenwarteschlange verfügbar ist.

Die dialogorientierte konversationelle Umgebung erfordert eine Scratch Pad Area (SPA) von mindestens 157 Byte zuzüglich des im Makro `NTIMSPT` des Natural-Parametermoduls angegebenen `NRAST`-Wertes.

Besondere Aspekte einer nicht-konversationellen Umgebung

Die dialogorientierte, nicht-konversationelle Umgebung wird durch das Non-Conversational MPP Interface realisiert, das mit dem Natural-Parametermodul zum Non-Conversational MPP Frontend verlinkt ist. Dieses Frontend ist das IMS TM-Anwendungsprogramm. Es wird von IMS TM eingeplant, wenn eine Eingabenachricht für den zugeordneten Transaktionscode in der IMS TM-Nachrichtenwarteschlange verfügbar ist.

Wenn eine dialogorientierte, nicht-konversationelle Umgebung verwendet wird, sind der *Natural Authorized Services Manager* mit aktivierter SIP-Funktion und die Physical Input Edit Routine Voraussetzung.

- Der *Natural Authorized Services Manager* wird verwendet, um die IMS TM Scratch Pad Area (SPA) zu simulieren.
- Die Physical Input Edit Routine wird verwendet, um den Transaktionscode vor der Eingabenachricht einzufügen.

Sie müssen die gleiche Natural-Subsystemkennung an folgenden Stellen angeben:

- Im Schlüsselwort-Subparameter `SPATID` des Makros `NTIMSPE` (Natural-Parametermodul),
- im Parameter `SPATID` des Makros `NIMPIXT`,
- in den Startparametern des Authorized Services Manager.

Besondere Aspekte einer MSC-Umgebung

In der folgenden Umgebung bereitet die Natural IMS TM-Schnittstelle die Nachricht `X'000500006D'` für `NAT-B` vor, was bedeutet, dass der Terminal-Benutzer CLEAR gedrückt hat.

```
          IMS-A                        IMS-B
          -----                        -----
MPP-A1  MPP-A2  MPP-A3     MPP-B1  MPP-B2  MPP-B3
------  ------  ------     ------  ------  ------
                  DIRECT SWITCH
   NAT-A ----------------------------> NAT-B
```

Es müssen zwei Einträge in der Transaktionscode-Tabelle erstellt werden: der erste Eintrag ist für `NAT-A`, der zweite für `NAT-B`.

Diese beiden Einträge müssen unterschiedliche Offsets für die Natural Reserved Area (NRA) angeben und sicherstellen, dass sich diese Bereiche nicht überschneiden.

`NAT-B` erkennt, dass eine Natural-Sitzung in `IMS-B` auf die übliche Weise gestartet werden soll, und übergibt daher die Kontrolle an seine Session-Start-Exit-Routine. Die Session-Start-Exit-Routine prüft die Eingabenachricht auf die Zeichenkette `X'000500006D'` und setzt die Länge der Eingabenachricht aus der Sicht von Natural auf `0`.

Wenn weder im Exit `NIIXSTAR` noch im Exit `NIIXSSTA` eine zusätzliche Logik vorgesehen ist, startet Natural eine neue Benutzersitzung in `IMS-B`.

Es wird davon ausgegangen, dass `IMS-A` und `IMS-B` über unterschiedliche, für Natural zugewiesene Roll Files verfügen.

Beide (oder mehrere) Natural-Sitzungen können miteinander kommunizieren, indem sie beim direkten Wechsel von einem Programm zum anderen Daten in der Scratch Pad Area (SPA) übertragen.

Wenn zwei oder mehr Natural-Sitzungen in einer solchen Umgebung existieren, wird derzeit nur die "aktive" Sitzung korrekt beendet.

Nachrichtenorientierte Umgebung

Dieser Abschnitt beschreibt die nachrichtenorientierte (message-oriented) Schnittstelle mit Natural for IMS TM.

- Einführung in die nachrichtenorientierte Umgebung
- Funktionsweise der nachrichtenorientierten Umgebung

Einführung in die nachrichtenorientierte Umgebung

Diese Schnittstelle ist für die Verarbeitung von Nettodaten-Eingabenachrichten konzipiert, was bedeutet, dass die Nachrichten keinen 3270-Datenstrom darstellen. Die nachrichtenorientierte Schnittstelle wird durch ein vom Anwender geschriebenes Natural-Programm gesteuert, das die Schnittstelle anweist, auf die IMS TM-Warteschlange zuzugreifen, um Eingabenachrichten abzurufen.

Die nachrichtenorientierte Schnittstelle wurde geschaffen, um nicht-konversationelle, nicht-terminalgesteuerte Transaktionen zu unterstützen, die als nicht-konversationelle MPP-Transaktionen ausgeführt werden müssen.

Funktionsweise der nachrichtenorientierten Umgebung

Die nachrichtenorientierte Schnittstelle umfasst Funktionen sowohl der MPP- als auch der BMP-Schnittstelle. Die BMP-Schnittstelle wird als Basis benutzt, da ein Großteil der erforderlichen Verarbeitung BMP-artige Transaktionen emuliert.

Da die nachrichtenorientierte Schnittstelle nicht terminalorientiert ist, werden keine Nachrichten oder Bildschirmmasken automatisch erzeugt, um an ein Terminal gesendet zu werden. Der Natural-Nukleus wird darüber informiert, dass er in einer Batch-Umgebung läuft. Deshalb wird die Ausgabe als Druckerausgabe interpretiert und die Eingabe aus einer `CMSYNIN`-Datei erwartet. Alle Ausgaben, die normalerweise in `CMPRINT` geschrieben werden, werden an das mit dem Natural-

Profilparameter SENDER angegebene IMS TM-Ziel gesendet. Einzelheiten finden Sie unter ***SENDER-Ziel*** weiter unten.

Wenn Natural versucht, Eingabedaten abzurufen, und von der Anwendung über das STACK-Kommando keine Eingabedaten geliefert wurden, zeigt EOF an, dass keine Eingabe vorhanden ist, und Natural wird beendet.

Sie können SENDER zur Laufzeit auf einen neuen Wert setzen, indem Sie das Service-Modul CMSNFPRT benutzen.

Mit Ausnahme der Checkpoint-Verarbeitung erfolgt die Verarbeitung bei Natural for Db2 so, als ob es sich um den BMP-Modus handelt. Dies ist notwendig, da eine physische Einplanung (Scheduling) mehrere nicht zusammenhängende Eingabemeldungen verarbeiten kann (und normalerweise auch wird). Bei der konversationellen MPP-Schnittstelle werden alle Transaktionen, die während einer Natural-Sitzung verarbeitet werden, innerhalb dieser Natural-Sitzung als zusammengehörig betrachtet, so dass die Datenbankpositionierung und die PCB-Verwendung beibehalten werden müssen.

Da Transaktionen, die während einer Einplanung (und einer Natural-Sitzung) verarbeitet werden, nicht miteinander zusammenhängen, ist die Speicherung von Natural-Sitzungsinformationen im Roll File nicht erforderlich. Daher muss für diese Schnittstelle kein Roll Dataset zugeordnet werden. Ein Roll Slot-Bereich wird weiterhin über GETMAIN zugeordnet und für die Speicherung aller Natural-Kontrollblöcke und Arbeitsbereiche verwendet.

Da die Verarbeitung auf der Basis von einzelnen Nachrichten nacheinander erfolgt, ist keine Verlagerungslogik erforderlich.

Bei der nachrichtenorientierten Schnittstelle wird der Abruf aller Nachrichten aus der Nachrichtenwarteschlange von einem Natural-Frontend-Programm initiiert. Dieses Programm muss vom Benutzer geschrieben werden, so dass es Ihre spezifischen Verarbeitungsanforderungen erfüllt. Es benötigt jedoch eine bestimmte Struktur, wie im Folgenden gezeigt wird:

```
PROGRAM INITIALIZATION
REPEAT
CALL 'CMGETMSG' MESSAGE-AREA MESSAGE-LENGTH
IF MSG-LL = 0 /* QC on GU to message queue
TERMINATE
FETCH RETURN PGMA MESSAGE-AREA
REPEAT
CALL 'CMGETSEG' MESSAGE-AREA MESSAGE-LENGTH
IF MSG-LL = 0 /* QD on GN to message queue
ESCAPE
FETCH RETURN PGMB MESSAGE-AREA
END-REPEAT
END-REPEAT
END
```

Das Servicemodul CMGETMSG liest das erste Nachrichtensegment. Das Servicemodul CMGETSEG liest alle weiteren Nachrichtensegmente.

Da Natural keine Eingaben von CMSYNIN lesen kann, ist es erforderlich, den Natural-Stack für Eingaben zu verwenden. Dies geschieht unter Verwendung des Natural-Profilparameters STACK.

Sie müssen jedoch dafür sorgen, dass Ihre Anwendung vor der Beendigung von Natural auf die IMS TM Message Queue zugreift. Ist dies nicht der Fall, wird die Natural-Transaktion mit dem IMS TM-Abendcode 462 beendet, was darauf hinweist, dass ein GU an die Nachrichtenwarteschlange nicht durchgeführt wurde.

Um diese Natural-Nachrichten auch im Falle eines abnormalen Abbruchs zu erhalten, wird empfohlen, den ersten alternativen PCB als EXPRESS PCB zu definieren.

Die nachrichtenorientierte Umgebung wird durch die NTRD-Schnittstelle implementiert, die mit dem Natural-Parametermodul zum NTRD-Frontend verlinkt ist. Dieses Frontend kann entweder direkt von IMS TM oder über ein Bootstrap-Modul aufgerufen werden, das mit dem Makro NIMBOOT erzeugt wurde.

Wenn es direkt von IMS TM aufgerufen wird, ist dieses Frontend das IMS TM-Anwendungsprogramm, das von IMS TM eingeplant wird, wenn eine Eingabenachricht für den zugewiesenen Transaktionscode in der IMS TM-Nachrichtenwarteschlange vorhanden ist. Es wird empfohlen, ein Natural-Profil zu verwenden, das den erforderlichen STACK-Parameter enthält. Geben Sie in Ihrem Natural-Parametermodul für den Profilparameter PROFILE den Wert PROGRAM an und erstellen Sie ein Profil mit einem Namen, der dem Transaktionscode entspricht, mit dem die Schnittstelle aufgerufen wird. Auf diese Weise haben Sie die Flexibilität, für jeden verwendeten Transaktionscode ein anderes Profil mit einer anderen STACK-Einstellung zu verwenden.

Wenn es über ein Bootstrap-Modul aufgerufen wird, ist dieses Bootstrap-Modul das IMS TM-Anwendungsprogramm, das von IMS TM eingeplant wird, wenn eine Eingabenachricht für den zugewiesenen Transaktionscode in der IMS TM-Nachrichtenwarteschlange verfügbar ist. Dieses Bootstrap-Modul stellt eine Reihe von dynamischen Profilparametern bereit, von denen einer der Profilparameter STACK ist, und ruft das NTRD-Frontend auf, dessen Name bei der Generierung des Bootstrap-Moduls angegeben wird. Wenn Sie Natural mit unterschiedlichen Einstellungen der dynamischen Profilparameter aufrufen wollen, müssen Sie unterschiedliche Bootstrap-Module generieren, die jeweils eine eigene Zeichenfolge dynamischer Profilparameter verwenden. Jedes dieser Bootstrap-Module muss unter einem eindeutigen Namen verlinkt werden. Außerdem muss jedem der entstehenden Lademodule ein eindeutiger IMS-TM-Transaktionscode zugewiesen werden.

Batch Message Processing-Umgebung

Die Umgebung zur Batch-Nachrichtenverarbeitung (Batch Message Processing, BMP) wird durch die BMP-Schnittstelle implementiert, die mit dem Natural-Parametermodul und der Arbeitsdatei-/Druckdatei-Zugriffsroutine `NATWKFO` mit dem BMP-Frontend verbunden ist. Dieses Frontend ist das IMS TM-Anwendungsprogramm, das in der BMP-JCL angegeben ist.

Ein Standard-Batch Natural wird in einer Batch Message Processing Region ausgeführt. Im Vergleich zum Standard-Batch-Natural-Lauf kann der optionale Input-Datensatz `CONTROL` verwendet werden.

Die optionale BMP `CONTROL`-Datei enthält maximal zwei Eingabekarten.

- Die erste Eingabekarte enthält die folgenden Schlüsselwortparameter:

Schlüsselwort	Bedeutung
`ENV-TAB=`	Der Name der zu verwendenden Umgebungstabelle.
`TRNCODE=`	Der Name des zu verwendenden Transaktionscodes, siehe die Beschreibung des Parameters `TRNCODE`.

Beispiel:

```
ENV-TAB=ENVBMPO TRNCODE=NATIMS
```

- Die zweite Eingabekarte der `CONTROL`-Datei enthält dynamische Natural-Parameter.

Verwendung des CMPRMIN-Dataset und der CONTROL-Datei zur Übergabe dynamischer Natural-Parameter

Wird der Dataset `CMPRMIN` auch zur Übergabe dynamischer Natural-Parameter verwendet, so wird die Eingabe von `CONTROL` an die Eingabe von `CMPRMIN` angehängt. Das bedeutet, die in `CONTROL` angegebenen Parameter überschreiben die in `CMPRMIN` angegebenen Parameter.

Betrieb ohne CONTROL-Datei

Wird die `CONTROL`-Datei nicht verwendet, so wird der Name der Umgebungstabelle durch den Eintrag in der Transaktionscode-Tabelle bestimmt, der dem verwendeten Transaktionscode (transaktionsorientierte BMP) bzw. dem verwendeten PSB-Namen (batchorientierte BMP) entspricht.

Unterstützung des Natural-Statements WRITE (n)

Mit dem Statement `WRITE (n)` können innerhalb eines Natural-Programms bis zu 31 verschiedene Reports auf verschiedenen Druckern erzeugt werden. Die Reports werden an die IMS TM-Terminals gesendet, die entweder im Natural-Parametermodul oder mit dem Natural-Statement `DEFINE PRINTER (n)` angegeben wurden. Im `NTPRINT`-Makro, das den Report steuert, müssen Sie `AM=IMS` angeben.

Um dieses Statement verwenden zu können, müssen Sie so viele zusätzliche alternative TP-PCBs in Ihrer PSB definieren, wie Sie parallele Reports innerhalb desselben Natural-Programms erstellen wollen, und die Anzahl der zusätzlichen alternativen TP-PCBs in Ihrer Transaktionscode-Tabelle mit dem Schlüsselwort-Subparameters `WRKPCBS` des Makros `NTIMSPT` (Natural-Parametermodul) angeben.

Caution: Beachten Sie, dass die erste alternative TP-PCB von der Natural IMS TM-Schnittstelle verwendet wird.

Bei Verwendung des Statements `WRITE (n)` in einer dialogorientierten Umgebung gilt die folgende Einschränkung:

Die Generierung eines Reports kann sich nicht über eine oder mehrere Bildschirm-Ein-/Ausgaben erstrecken. Wenn Sie denselben Drucker nach einer Bildschirm-Eingabe/Ausgabe verwenden wollen, müssen Sie ihn vor der Bildschirm-Eingabe/Ausgabe mit dem Statement `CLOSE PRINTER (n)` explizit schließen.

Für die Erstellung von Reports sind die folgenden Schlüsselwort-Subparameter des Makros `NTPRINT` relevant:

Schlüsselwort-Subparameter	Bedeutung
`AM`	Muss auf `IMS` gesetzt werden.
`DEST`	Gibt das IMS TM-Ziel an.
`BLKSIZE`	Gibt die Größe des Puffers an, der an das Ziel gesendet wird. Reportzeilen werden gepuffert.
`DRIVER`	Gibt den Treiber an, der für die Erstellung des Reports verwendet werden soll. Eine Liste der möglichen Werte finden Sie im Schlüsselwort-Subparameter `PRTDRIV` des Makros `NTIMSPE` im Natural-Parametermodul. Der Treiber bestimmt, wo der Seitenvorschub erfolgen soll (am Anfang des Reports, am Ende, sowohl am Anfang als auch am Ende oder kein Seitenvorschub), wo die Seite beginnen soll (in Zeile 1 oder in Zeile 2, um die Kompatibilität mit dem Natural IMS TM Interface Version 2.2 zu gewährleisten) und wo der Report gedruckt werden soll (SCS- oder Nicht-SCS-Drucker). Darüber hinaus können Sie angeben, dass Sie die JES API verwenden möchten.
`NAME` `FORMS`	Diese Parameter werden nur ausgewertet, wenn Sie die JES API verwenden.

Schlüsselwort-Subparameter	Bedeutung
DISP COPIES CLASS PRTY	

Hinweise zu NTPRINT und CLOSE PRINTER

NTPRINT-Einstellungen

Es wird dringend empfohlen, für IMS TM-Drucker (d.h. für Drucker, die mit AM=STD definiert sind) immer die Standardwerte für die Subparameter OPEN und CLOSE im NTPRINT-Makro oder in der PRINT-Profilparameterdefinition zu verwenden. Das bedeutet, dass Sie entweder keine Werte für OPEN und CLOSE angeben oder die Standardwerte OPEN=ACC und CLOSE=CMD verwenden.

Dies ist vor allem dann wichtig, wenn Sie einen Drucker im Natural-Parametermodul statisch für eine andere Zugriffsmethode mit anderen Optionen für OPEN und CLOSE definiert haben und wenn Sie die Zugriffsmethode dynamisch mit AM=IMS überschreiben. Geben Sie in diesem Fall immer AM=IMS,OPEN=ACC,CLOSE=CMD zusammen an.

Note: Die NTPRINT-Optionen werden mit den dynamisch angegebenen PRINT-Optionen zusammengeführt, obwohl die Zugriffsmethode überschrieben wurde.

Probleme, die bei nicht standardmäßigen Werten auftreten können:

1. Bei OPEN=OBJ können Sie an ein falsches Ziel drucken oder einen NAT8211-Fehler erhalten, wenn die Option OUTPUT in einem DEFINE PRINTER-Statement angegeben wurde.

   Bei OPEN=OBJ wird der Drucker geöffnet, bevor die OUTPUT-Überschreibung ausgewertet wurde, und als Druckziel wird nicht das in der OUTPUT-Option, sondern das mit dem PRINT-Parameter angegebene Ziel verwendet.
2. Bei CLOSE=FIN wird der Drucker nicht zum Zeitpunkt CLOSE PRINTER, sondern zum Zeitpunkt FIN geschlossen.

   Das bedeutet, dass das CLOSE erfolgen kann, nachdem ein GU an die Nachrichtenwarteschlange ausgegeben wurde und das Ziel im TP PCB zurückgesetzt wurde.

   Dies führt zu dem NII-Fehler NII3641 beim IMS TM-Statuscode QF (MPP) oder A3 (BMP und OBMP/NTRD). Mit CLOSE=CMD wird der Drucker tatsächlich mit dem Statement CLOSE PRINTER geschlossen.

Verwendung von CLOSE PRINTER oder DEFINE PRINTER

Ein an einen IMS TM-Drucker gerichteter Report wird von IMS TM mit dem nächsten GU-Aufruf implizit geschlossen (d.h. entweder über Terminal-Ein-/Ausgabe oder über `CMGETMSG`). Das bedeutet, dass IMS TM den Report unabhängig von einem `CLOSE PRINTER`- oder `DEFINE PRINTER`-Statement im Programm drucken wird.

Bei Natural ist der Drucker noch geöffnet, und das nächste `WRITE`-Statement mit der gleichen Reportnummer wird den bereits gedruckten Report fortsetzen, was zu einem NAT1518-Fehler führt.

Szenario:

```
DEFINE PRINTER (1)
  WRITE (1) 'line 1'
  INPUT 'Press ENTER' or CALL 'CMGETMSG' (both issue a GU)
  WRITE (2) 'line 2'
```

Mit `INPUT`/`CMGETMSG` wird der Drucker “physisch” geschlossen und IMS TM druckt einen Report, der die Zeile `'line 1'` enthält.

Da der Drucker noch “logisch” für Natural geöffnet ist, wird mit der Zeile `'line 2'` kein neuer Report gestartet, und es wird der Fehler NAT1518 verursacht, da das Ziel durch den GU-Aufruf gelöscht wird.

Es wird daher dringend empfohlen, die folgende Regel zu beachten:

Caution: Wenn nach dem GU der Report mit der gleichen Nummer fortgesetzt wird, ist ein `CLOSE PRINTER`-Statement erforderlich.

Bitte beachten Sie, dass das `DEFINE PRINTER`-Statement ein implizites Schließen durchführt, so dass z.B. das `CLOSE PRINTER`-Statement hinfällig ist. Beispiel:

Korrekt	Korrekt	Falsch (NAT1518)

`REPEAT`
`  DEFINE PRINTER (1)`
`  WRITE (1)`
 	
`  INPUT`
`LOOP`
	`DEFINE PRINTER (1)`
	`REPEAT`
	`  WRITE(1)`
	`  CLOSE PRINTER (1)`
	`  INPUT`
	`LOOP`
		`DEFINE PRINTER (1)`
		`REPEAT`
		`  WRITE(1)`
		
		`  INPUT`
		`LOOP`

SET CONTROL 'N' (Terminalkommando %N)

Das Statement `SET CONTROL 'N'` (Terminalkommando `%N`) gilt nicht unter IMS TM. Wenn es unter IMS TM verwendet wird, bewirkt es, dass der nächste logische Ausgabebildschirm unterdrückt wird.

Unterstützung von TS=ON bei Natural unter IMS TM-Nachrichten

Alle Nachrichten von Natural unter IMS TM werden in Großbuchstaben umgesetzt, wenn `TS=ON` in der Natural-Sitzung angegeben ist.

SENDER-Ziel

In der nachrichtenorientierten Umgebung (`NTRD`) und in der Serverumgebung (`SRVD`) werden alle Ausgaben, die normalerweise in `CMPRINT` geschrieben werden, an das mit dem Natural-Profilparameter `SENDER` angegebene Ziel gesendet. Mit `SENDER` geben Sie entweder ein gültiges IMS TM-Ziel an (in der Regel ein `LTERM`), an das die Ausgabe über die IMS TM-Nachrichtenwarteschlange gesendet wird, oder Sie geben einen der folgenden reservierten Werte an:

Wert	Bedeutung
`*WTO`	Die Natural-Ausgabe wird über eine WTO mit Routing Code 11 (Programmiererinformation) an das Job Log gesendet.
`*MTO`	Die Natural-Ausgabe wird über ein `/BROADCAST MASTER`-Kommando an die IMS TM-Hauptkonsole gesendet.
`*PRINT`*`nn`	Die Natural-Ausgabe wird auf den Natural-Drucker *`nn`* geschrieben, d.h. die Ausgabe wird mit einem internen `WRITE(`*`nn`*`)` geschrieben. Die Druckdatei wird nach jeder Ausgabezeile geschlossen, und das Wagensteuerzeichen jeder Ausgabezeile ist das Leerzeichen.

Notes:

1. Wenn das Kommando `/BROADCAST MASTER` fehlschlägt (z.B. aufgrund von Berechtigungsproblemen), wird eine Fehlermeldung über eine WTO ausgegeben und alle Natural-Nachrichten, einschließlich der aktuellen Nachricht, werden an das Job Log gesendet. Das heißt, das `SENDER`-Ziel (Destination) wird intern auf `*WTO` gesetzt.
2. Im Falle von `*PRINT` *`nn`* wird dringend empfohlen, nur Drucker zu verwenden, die mit `AM=STD` definiert sind.

3. Es wird dringend empfohlen, das `SENDER`-Ziel (Destination) im Natural-Profilparametermodul zu definieren. Dadurch wird sichergestellt, dass das Ziel auch dann gefunden wird, wenn die Natural-Initialisierung fehlschlägt (z.B. aufgrund eines Adabas-Fehlers NAT3048 oder NAT3148).
4. Wenn Sie das `SENDER`-Ziel unbedingt dynamisch angeben wollen, müssen Sie die reservierten Werte in einfache Anführungszeichen setzen (z. B. `SENDER='*WTO'`).

Unterstützung des Natural-Profilparameters PROGRAM

Der Natural-Profilparameter `PROGRAM` wird in den dialogorientierten Umgebungen und in der BMP-Umgebung unterstützt.

In der BMP-Umgebung verhält sich der Parameter `PROGRAM` wie in einer Standard-z/OS-Batch-Umgebung. Ein Beispiel mit dem Namen `XNIIBACK` wird im Dataset `NII`*`vrs`*`.SRCE` ausgeliefert. Es wird erwartet, dass das aufgerufene Backend-Programm zur Natural IMS TM-Schnittstelle zurückkehrt.

In einer dialogorientierten Umgebung verhält sich der Natural-Profilparameter `PROGRAM` etwas anders als in anderen Natural-Umgebungen, einschließlich der BMP-Umgebung.

1. Der mit dem Profilparameter `PROGRAM` oder dem Natural-Subprogramm `CMPGMSET` angegebene Name ist der Name eines IMS TM-Transaktionscodes.
2. Der angegebene Transaktionscode wird nur aufgerufen, wenn die Natural-Sitzung ohne Fehler beendet wird.
3. Die mit dem `TERMINATE`-Statement gelieferten Daten werden als Eingabenachricht an die aufgerufene IMS TM-Transaktion weitergegeben.

Natural Development Server- bzw. Natural Web I/O Server-Umgebung

Bei dieser Umgebung können Sie eine Natural Development Server (NDV)- oder Natural Web I/O Interface (NWO)-Server-Sitzung unter der Kontrolle von IMS TM ausführen und aus einer solchen Sitzung heraus auf IMS TM-Ressourcen wie IMS TM DB oder Db2-Datenbanken zugreifen. Die Server-Transaktion ist eine nicht-konversationelle Transaktion.

Weitere Informationen finden Sie unter:

Introducing the Natural Web I/O Interface Server IMS Adapter , *Installing the Natural Web I/O Interface Server IMS Adapter under z/OS* und *Configuring the Natural Web I/O Interface Server IMS Adapter* in der *Natural Web I/O Interface*-Dokumentation.

Einschränkung für diese Umgebung:

Der Zugriff auf die IMS TM-Ressourcen erfolgt über die Benutzerkennung `EZAIMSLN`. Das heißt, es wird keine Impersonation verwendet.

23 Natural unter IMS TM - Komponenten

- Frontend-Modul 140
- Natural IMS TM-Schnittstellenmodul NIIINTFM 141
- Editierroutine für physische Eingaben (Physical Input Edit Routine) 142
- Benutzernachrichtentabelle (User Message Table) DFSCMTU0 142
- Roll File und Roll Server 143
- Authorized Services Manager 144
- Gemeinsam genutzter Natural-Nukleus 145
- Natural Buffer Pool 145
- Adabas-Schnittstelle 145
- Preload-Liste 146

Dieses Kapitel beschreibt die Komponenten der Natural IMS TM-Schnittstelle.

Frontend-Modul

Das Frontend-Modul wird vom IMS TM Program Controller `DFSPPC20` gesteuert, außer in der Server-Umgebung, wo es über die Aufrufschnittstelle `NIIBOOTS` aufgerufen wird.

Das Front-End-Modul muss während des Installationsvorgangs der Natural IMS TM-Schnittstelle erstellt werden. Siehe *Natural IMS TM-Schnittstelle auf z/OS installieren* in der *Installation für z/OS*-Dokumentation.

Das Frontend-Modul besteht aus den folgenden Komponenten:

- Umgebungsabhängige Schnittstellen (Treiber)
- Natural-Parametermodul
- Zugriffsmethodenmodul NATWKFO für Natural-Arbeitsdateien und Druckdateien (AM=STD)

Umgebungsabhängige Schnittstellen (Treiber)

Die umgebungsabhängigen Schnittstellen werden in Form der folgenden Load-Module ausgeliefert:

- `NIIBMP` für die **Umgebung der Batch-Nachrichtenverarbeitung**,
- `NIICONV` für die **konversationelle dialogorientierte Umgebung**,
- `NIINONC` für die **nicht-konversationelle dialogorientierte Umgebung**,
- `NIINTRD` für die **nachrichtenorientierte Umgebung**,
- `NIISFE` für die **Natural Development Server-/Natural Web I/O Interface Server-Umgebung** und
- `NIISRVD` für die **Server-Umgebung**.

Die Load-Module werden auf dem Installationsmedium mitgeliefert. Die umgebungsabhängigen Schnittstellen können Sie mit dem Makro `NTIMSP` des Natural-Parametermoduls konfigurieren. Siehe auch ***Natural unter IMS TM - Konfiguration***.

Natural-Parametermodul

Das Natural-Parametermodul wird während der Installation erstellt. Siehe auch *Generierung eines Natural-Parametermoduls* in der *Operations*-Dokumentation.

Zusätzlich zu den Parametereinstellungen, die Sie für Ihre Natural-Umgebung anpassen müssen, müssen Sie mindestens ein `NTIMSPT`-Parametermakro für die Natural IMS TM-Schnittstelle angeben. Siehe auch ***Natural unter IMS TM - Konfiguration***.

Die einzelnen Parameter und Makros, die mit dem Natural-Parametermodul angegeben werden können, sind in der *Parameter-Referenz*-Dokumentation beschrieben.

Zugriffsmethodenmodul NATWKFO für Natural-Arbeitsdateien und Druckdateien (AM=STD)

Das Modul `NATWKFO` wird als Teil des Natural-Basisprodukts ausgeliefert. Es dient zur Handhabung von Natural-Arbeitsdateien und Druckdateien, die mit Zugriffsmethode (Access Method) `AM=STD` definiert sind. Es ist anwendbar auf die **BMP-Umgebung**, die **nachrichtenorientierte Umgebung** und die **Server-Umgebung**. Es gilt nicht für die dialogorientierten Umgebungen.

Einige Natural-Produkte, z.B. Natural for Db2, erfordern, dass ihre Module mit dem Natural IMS TM-Frontend-Modul verlinkt sind. Weitere Informationen hierzu finden Sie in der entsprechenden Produktdokumentation.

Natural IMS TM-Schnittstellenmodul NIIINTFM

Das Natural IMS TM-Schnittstellenmodul `NIIINTFM` muss während des Installationsvorgangs erstellt werden und ist für alle Umgebungen gleich.

Das Schnittstellenmodul besteht aus den folgenden Komponenten:

- Natural IMS TM-Nukleus NIINUC
- Nachrichtentextmodul NIIMSGT

Das Schnittstellenmodul ist vollständig reentrant und kann oberhalb der 16-MB-Grenze ausgeführt werden. Es ist daher für die ECSA geeignet, damit nur eine Kopie des Schnittstellenmoduls für alle IMS TM-Umgebungen vorhanden ist.

Natural IMS TM-Nukleus NIINUC

Der Natural IMS TM-Nucleus NIINUC wird als Load-Modul ausgeliefert und enthält alle von der Natural IMS TM-Schnittstelle benötigten Laufzeitroutinen.

Nachrichtentextmodul NIIMSGT

Das Nachrichtentextmodul NIIMSGT ist Teil des Natural IMS TM Interface-Moduls und wird sowohl als Load- als auch als Source-Modul ausgeliefert. Er enthält zu jedem möglichen Natural IMS TM-Laufzeitfehler den entsprechenden Meldungstext. Jeder Eintrag wird durch das Makro NIMMSGT generiert.

Eine detaillierte Beschreibung des Makros NIMMSGT finden Sie unter ***NIMMSGT-Makroparameter***.

Editierroutine für physische Eingaben (Physical Input Edit Routine)

Die Editierroutine für physische Eingaben wird nur in einer dialogorientierten, nicht-konversationellen Umgebung benötigt. Sie wird verwendet, um den Transaktionscode vor der an das Terminal gesendeten Nachricht einzufügen. Dies ist erforderlich, da Natural im MFS-Bypass-Modus läuft und die an das Terminal gesendete Nachricht keinen Transaktionscode enthält.

Die Editierroutine für die physische Eingabe wird mit Hilfe des Makros NIMPIXT generiert. Weitere Informationen über das Makro finden Sie unter ***NIMPIXT-Makro-Parameter***.

Sobald die Editier-Routine für die physische Eingabe generiert ist, muss ihr Name im Makro TYPE oder LINEGRP Ihrer IMS TM-Systemdefinition angegeben werden. Für alle Terminals, auf denen die nicht-konversationelle Umgebung laufen soll, müssen Sie die physische Bearbeitung mit dem Parameter EDIT im Makro TERMINAL aktivieren.

Benutzernachrichtentabelle (User Message Table) DFSCMTU0

Die mitgelieferte User Benutzernachrichtentabelle DFSCMTU0 wird nur in einer dialogorientierten, nicht-konversationellen Umgebung benötigt. Sie enthält die Fehlermeldungen für Fehler, die von der Editierroutine für physische Eingaben erkannt werden.

Die Benutzernachrichtentabelle DFSCMTU0 muss in die bestehende User-Message-Tabelle Ihrer IMS TM-Installation integriert werden. Bei Konflikten mit bereits vorhandenen Benutzernachrichtennummern Ihrer IMS TM-Installation können Sie die Nachrichtennummern der ausgelieferten DFSCMTU0 ändern, indem Sie die EQUATES PIXTE und SIPSE modifizieren, um neue Nachrichtennummernkreise zu erzeugen. Der neue Startwert des Nachrichtennummernkreises muss im Makro NIMPIXT angegeben werden.

Roll File und Roll Server

Die Komponenten Roll File und Roll Server werden nur in dialogorientierten Umgebungen verwendet.

Natural-Sitzungsbezogene Informationen werden im Natural Thread abgelegt. Bei jeder Terminalausgabe wird der Inhalt des Natural Thread entweder in einem Roll File oder unter Verwendung des Roll Servers gespeichert. Das Medium wird durch den Schlüsselwort-Subparameter `ROLLSRV` im Makro `NTIMSPE` (Natural-Parametermodul) definiert. Siehe Natural-*Parameter-Referenz*-Dokumentation.

Verwendung von Roll Files

Um Roll Files verwenden zu können, muss der Schlüsselwort-Subparameter `ROLLSRV` auf `OFF` gesetzt werden.

Für jeden Natural-Benutzer wird bei der Initialisierung der Natural-Sitzung ein Roll Slot im Roll File reserviert. Die Kennung des Slots ist der IMS TM `LTERM`, an dem die Natural-Sitzung gestartet wird. Sie müssen daher sicherstellen, dass alle Terminals, die denselben Satz an Roll Files verwenden, unterschiedliche `LTERM`-Namen haben. Dies ist immer der Fall, wenn die Roll Files von einem einzigen IMS TM verwendet werden. Der Slot wird freigegeben, wenn die Natural-Sitzung normal beendet wird. Im Falle einer abnormalen Beendigung bleibt der Roll Slot zugeordnet, wird jedoch wieder verwendet, wenn derselbe Benutzer (identifiziert durch dessen `LTERM`) eine neue Natural-Sitzung startet.

Der Zugriff auf Roll Files erfolgt über die DD-Anweisungen `ROLLF1` - `ROLLF5`. Die Anzahl der verwendeten Roll Files wird mit dem Schlüsselwort-Subparameter `ROLLFN` im Makro `NTIMSPE` festgelegt.

Wenn Ihr Natural-Transaktionscode in mehr als einer MPP-Region eingeplant ist oder wenn Sie zwischen Transaktionscodes wechseln, die in verschiedenen MPP-Regionen laufen, müssen Sie in allen MPP-Regionen die gleichen Roll Files verwenden.

Wenn Sie Roll Files neu formatieren, müssen Sie sicherstellen, dass keine Natural-Transaktionen aktiv sind. Wenn eine Transaktion nach der Neuinitialisierung des Roll File eingeplant wird, kann sie ihren Roll Slot im Roll File nicht finden und bricht vorzeitig ab. Um dieses Problem zu vermeiden, empfiehlt es sich, IMS TM nach der Neuformatierung des Roll File kalt zu starten.

Die von Natural unter IMS TM verwendeten Roll Files haben das gleiche Layout wie die vom Roll Server verwendeten Roll Files und werden mit dem gleichen Dienstprogramm (Utility) formatiert.

Notes:

1. Diese Roll Files können nur BDAM-Dateien sein, VSAM Files werden von IMS/TM nicht unterstützt.
2. Die von Natural unter IMS TM verwendeten Roll Files dürfen nicht gemeinsam mit dem Roll Server genutzt werden. Wenn Sie Roll Files für Natural unter IMS TM und den Roll Server gleichzeitig verwenden, müssen Sie dem Roll Server einen eigenen Satz von Roll Files zuweisen.

Verwendung des Roll Server

Um den Roll Server verwenden zu können, muss der Schlüsselwort-Subparameter `ROLLSRV` auf `ON` gesetzt werden.

Anstelle von Roll Files, die jeder MPP-Region zugeordnet werden müssen, können Sie den Natural Roll Server verwenden. Der Roll Server bietet die folgenden Vorteile:

- Keine DD-Statements in jeder MPP-Region.
- Für den Zugriff auf die Roll Files ist ein zentraler Adressraum zuständig.
- Unterstützung von Hauptspeicherpuffern zur Verringerung der Festplatten-E/As auf die Roll Files.

Für jeden Natural-Benutzer wird bei der Initialisierung der Natural-Sitzung ein Slot im Roll Server reserviert. Die Kennung des Slots (Roll-Server-Benutzerkennung) ist das IMS TM `LTERM`, bei dem die Sitzung gestartet wird, verkettet mit der z/OS-Host-Kennung und der IMS TM-Subsystemkennung der IMS TM-abhängigen Region, in der die entsprechende Natural-Transaktion geplant ist. Der Slot wird freigegeben, wenn die Natural-Sitzung normal beendet wird. Im Falle einer abnormalen Beendigung bleibt der Slot zugeordnet, wird aber wieder verwendet, wenn derselbe Benutzer (identifiziert durch dessen `LTERM`) eine neue Natural-Sitzung startet.

In einer z/OS Parallel Sysplex-Umgebung müssen Sie den Roll Server verwenden.

Weitere Informationen zu Roll Files und zum Roll Server finden Sie unter *Roll Server* in der Natural-*Operations*-Dokumentation.

Authorized Services Manager

Der *Natural Authorized Services Manager* ist in den folgenden Fällen erforderlich:

- In einer dialogorientierten, nicht-konversationellen Umgebung siehe ***Besondere Aspekte einer nicht-konversationellen Umgebung***).
- Bei Verwendung von Monitoring oder Broadcasting siehe ***Monitoring (Überwachung)*** oder ***Broadcasting (Rundsenden)***.
- Bei Verwendung von Accounting, und wenn die Accounting-Informationen in das SMF geschrieben werden, siehe ***Accounting (Abrechnung)***.

- Bei Verwendung von Buffer Pool Propagation siehe Profilparameter `BPPROP`.

In den ersten beiden Fällen muss die optionale SIP-Funktion während des Starts des Authorized Services Manager verfügbar gemacht werden.

In einer z/OS Parallel Sysplex-Umgebung muß sich der SIP in einer Coupling Facility befinden.

Gemeinsam genutzter Natural-Nukleus

In einer IMS TM-Umgebung ist der Natural-Nukleus stets von der umgebungsabhängigen Schnittstelle (Treiber) getrennt. Das bedeutet, dass Sie den gemeinsam genutzten Natural-Nukleus installieren müssen. Derselbe Natural-Nukleus kann von allen Natural IMS TM-Umgebungen gemeinsam genutzt werden.

Natural Buffer Pool

Da Natural unter IMS TM in mehr als einer MPP-Region ausgeführt werden kann, wird empfohlen, dass als Natural Buffer Pool ein globaler Buffer Pool verwendet wird.

Zwar können Sie einen lokalen Buffer Pool verwenden, dies wird aber in terminalgesteuerten Umgebungen aus Performance-Gründen nicht empfohlen.

Weitere Informationen siehe *Natural Global Buffer Pool* in der Natural-*Operations*-Dokumentation.

Adabas-Schnittstelle

Für den Zugriff auf die Natural-Systemdatei und die Adabas-Benutzerdateien wird die Adabas-Schnittstelle benötigt.

Standardmäßig wird die entsprechende Adabas-Schnittstelle zur Laufzeit dynamisch geladen.

- In terminalgesteuerten, dialogorientierten Umgebungen wird das Adabas IMS TM Interface-Modul `ADALNI` verwendet.
- In allen anderen Umgebungen wird das Adabas-Batch-Schnittstellenmodul `ADALNK` verwendet.

Sie können den Namen der zu verwendenden Adabas-Schnittstelle durch Angabe des Natural-Profilparameters `ADANAME` überschreiben.

Caution: Sie dürfen keines der beiden Schnittstellenmodule in der reentranten Version verwenden.

Preload-Liste

Die Verwendung einer Preload-Liste ist für die Natural IMS TM-Schnittstelle nicht mehr erforderlich. Aus Performance-Gründen empfiehlt es sich jedoch, die Namen der folgenden Module in die Preload-Liste für die Natural-Regionen aufzunehmen:

- die Natural IMS TM Frontends,
- das Natural IMS TM-Schnittstellen-Modul,
- den gemeinsam genutzten Natural-Nukleus,
- die Adabas-Schnittstelle.

24 Natural unter IMS TM - Konfiguration

- NIMMSGT-Makro-Parameter 148
- NIMPIXT-Makro-Parameter 149
- NIMBOOT-Makro-Parameter 150

Die wichtigsten Parameter, die zur Konfiguration der Natural-IMS-TM-Schnittstelle erforderlich sind, werden in den folgenden Profilparametermakros des Natural-Parametermoduls angegeben:

- `NTIMSP` für allgemeine Parametereinstellungen,
- `NTIMSPE` für die Angabe von umgebungsspezifischen Parametersätzen,
- `NTIMSPT` für Transaktionscode-Definitionen.

Important: Sie müssen mindestens das Makro `NTIMSPT` angeben und individuelle Parameter für Ihre Natural IMS TM-Transaktionen definieren (es gibt keine Standardwerte).

Zusätzlich zu den Natural-Parameter-Makros können Sie die im folgenden Abschnitt beschriebenen optionalen Makros verwenden. Diese werden von der Natural IMS TM-Schnittstelle zur Verfügung gestellt.

NIMMSGT-Makro-Parameter

Das Makro `NIMMSGT` generiert jeden Eintrag im Nachrichtentextmodul `NIIMSGT`, das Teil des Natural IMS TM-Schnittstellen-Moduls ist. Jeder generierte Eintrag liefert einen Meldungstext für jede mögliche Natural IMS TM-Fehlernummer.

Das Makro `NIMMSGT` wird auf eine der beiden folgenden Arten angegeben:

```
Nerror-number [*] NIMMSGT  message-text
```

In diesem Fall zeigt Natural unter IMS TM den Meldungstext so, wie er definiert wurde, an. Der Meldungstext kann bis zu 72 Zeichen lang sein.

```
Xerror-number [*] NIMMSGT message-text
```

In diesem Fall hängt Natural unter IMS TM einen fehlerspezifischen Reason Code an den aktuellen Meldungstext an. Der Meldungstext kann bis zu 64 Zeichen lang sein.

Wenn der Fehlernummer ein Stern (*) folgt, wird beim Auftreten eines Fehlers ein Snap-Dump erstellt. Sie können den Meldungstext an Ihre eigenen Erfordernisse anpassen. Sie können auch die Dump-Option bei einer bestimmten Fehlernummer hinzufügen oder löschen. Die Fehlernummer und die der Fehlernummer vorangestellten Zeichen `N` oder `R` dürfen Sie nicht verändern.

NIMPIXT-Makro-Parameter

Das Makro NIMPIXT generiert die Editierroutine für physische Eingaben (**Physical Input Edit Routine**).

Die Parameter, die mit dem Makro NIMPIXT angegeben werden können, sind im Folgenden in alphabetischer Reihenfolge aufgeführt:

PIXTE | SIPSE | SPATID | WTO | USER

Parameter	Möglicher Wert	Beschreibung	Standardwert	Kommentar
PIXTE	1 - 999	Legt den Startwert für Fehlernummern fest, wenn von der Editierroutine für physische Eingaben Fehler erkannt werden.	400	Dieser Wert wird zum Rückgabecode addiert, der von der Editierroutine für physische Eingaben gesetzt wird. Das Ergebnis ist die Nummer der IMS-TM-Fehlermeldung in der User Message-Tabelle DFSCMTU0.
SIPSE	1 - 999	Gibt den Startwert für Fehlernummern an, wenn Fehler durch den *Authorized Services Manager* erkannt werden.	500	Dieser Wert wird zu dem vom *Authorized Services Manager* gesetzten Rückgabecode addiert. Das Ergebnis ist die Nummer der IMS TM-Fehlermeldung in der User Message-Tabelle DFSCMTU0.
SPATID	*xxxx*	Gibt die Natural-Subsystemkennung für den *Authorized Services Manager* an, die zum Speichern der SPA für den nicht-konversationellen Treiber verwendet wird. Es ist eine beliebige Zeichenkette mit bis zu 4 Zeichen möglich.	Keiner	Der Wert dieses Parameters muss mit dem für den Schlüsselwort-Subparameter SPATID im Makro NTIMSPE (Natural-Parametermodul) angegebenen Wert übereinstimmen.
WTO	YES NO	Gibt an, ob eine WTO-Meldung ausgegeben wird, wenn der *Authorized Services Manager* fehlschlägt.	NO	Keiner.
USER	*xxxxxxxx* NO	Gibt an, ob eine benutzerspezifische physische Eingabe-Editierroutine aufgerufen werden soll, wenn das NIMPIXT-Makro die SPA nicht findet. Wenn eine benutzerspezifische Eingabe-Editierroutine aufgerufen	NO	Keiner.

Parameter	Möglicher Wert	Beschreibung	Standardwert	Kommentar
		werden soll, muss der Name der Routine angegeben werden.		

NIMBOOT-Makro-Parameter

Das Makro NIMBOOT generiert das von der nachrichtenorientierten Umgebung verwendete Bootstrap-Modul bzw. die von der Server-Umgebung verwendete Server-Call-Schnittstelle.

Das Makro NIMBOOT enthält die folgenden Parameter:

TYPE | DRIVERN | ENVTNAM | TRNCODE | DYNPARM | SERVERN

Parameter	Möglicher Wert	Standardwert	Kommentar
TYPE	SERVER	*Leer*	TYPE gibt den Typ des zu generierenden Schnittstellenmoduls an. Bei Angabe von TYPE=SERVER, wird die Server-Call-Schnittstelle NIIBOOTS generiert.
	Leer		Wenn nichts angegeben wird, wird das von der nachrichtenorientierten Umgebung verwendete Bootstrap-Modul generiert.
DRIVERN	Beliebiger, gültiger z/OS-Modulname.	Keiner	Dieser Parameter gibt den Namen des Frontend-Moduls an. Bei Angabe von TYPE=SERVER muss das Frontend-Modul für die Serverumgebung generiert worden sein. Wenn kein TYPE angegeben ist, muss das Frontend-Modul für die nachrichtenorientierte Umgebung generiert worden sein.
ENVTNAM	Beliebiger, gültiger z/OS-Modulname.	Keiner	Dieser Parameter wird nur von dem Bootstrap-Modul für die nachrichtenorientierte Umgebung verwendet (TYPE ist leer). Dieser Parameter gibt den Namen der Umgebungstabelle an. Dieser Parameter ist optional. Wenn er nicht angegeben wird, wird die Umgebungstabelle durch den Eintrag in der Transaktionscode-Tabelle bestimmt, der dem verwendeten Transaktionscode entspricht.
TRNCODE			Dieser Parameter wird nur von dem Bootstrap-Modul für die nachrichtenorientierte Umgebung verwendet (TYPE ist leer). Dieser Parameter gibt den Namen des Transaktionscodes an, der intern von der Natural IMS TM-Schnittstelle verwendet wird.

Parameter	Möglicher Wert	Standardwert	Kommentar
			Dieser Parameter ist optional und wird nur berücksichtigt, wenn im Makro NTIMSP (Natural-Parametermodul) TRNCODE angegeben ist. Wird er nicht angegeben oder ist im NTIMSP-Makro TRNCODE=ON angegeben, wird der vom IMS TM INQY-Aufruf zurückgegebene Transaktionscode verwendet. Der Transaktionscode wird verwendet, um den Eintrag in der Transaktionscode-Tabelle zu bestimmen.
DYNPARM	Beliebige Zeichenkette mit bis zu 80 Zeichen.	Keiner	Dieser Parameter wird nur von dem Bootstrap-Modul für die nachrichtenorientierte Umgebung verwendet (TYPE ist leer). Mit diesem Parameter wird eine gültige Zeichenkette von bis zu 80 Zeichen für dynamische Natural-Parameter definiert.
SERVERN	Beliebiger, gültiger z/OS-Modulname.	NIIBOOTS	Dieser Parameter wird nur von der Server-Call-Schnittstelle (TYPE) verwendet. Dieser Parameter gibt den Namen der Serverumgebung an. Er ist nur relevant, wenn Sie mehrere Natural-Server in derselben Region verwenden wollen. In diesem Fall müssen Sie mehrere Server-Call-Schnittstellen generieren und für jede einzelne einen eindeutigen Namen mit SERVERN angeben. Siehe *Spezielle Funktionen*, ***Server-Umgebung***.

25 Natural unter IMS TM - Serviceprogramme

- Einführung in die Serviceprogramme der Natural IMS TM-Schnittstelle 154
- Beschreibung der Serviceprogramme der Natural IMS TM-Schnittstelle 155
- NIIBRCST - Übergebene Nachricht an Terminal senden 155
- NIICMD - IMS TM-Kommando an IMS TM übergeben 155
- NIIDEFT - Aufgeschobenen Wechsel zu Natural-Transaktionscode vorbereiten 156
- NIIDEFTX - Aufgeschobenen Wechsel zu nicht-Natural-Transaktionscode vorbereiten 156
- NIIDIRT - Direkten Wechsel zu Natural-Transaktionscode vorbereiten 157
- NIIDIRTX - Direkten Wechsel zum Transaktionscode vorbereiten 158
- NIIEMOD - Einstellung des Modul-Ausgabe-Deskriptors ändern 159
- NIIGCMD - Nächstes Antwortsegment des vorherigen IMS TM-Kommandos abrufen 159
- NIIGMSG - Erstes Segment der nächsten Nachricht abrufen 160
- NIIGSEG - Nächstes Segment der Eingabenachricht abrufen 160
- NIIGSPA - Daten ab SPA-Beginn abrufen 161
- NIIIMSIN - IMS TM-Umgebungsinformationen abrufen 161
- NIIISRTF - Mehrere Segmente enthaltende Nachrichten erstellen 162
- NIIISRTM - Nachrichtensegment in Nachrichtenwarteschlange einfügen 163
- NIIPCBAD - PSB-Name und PCB-Adresse zurückgeben 163
- NIIPCOM - Daten in Antwortbereich verschieben 164
- NIIPMSG - Nachricht senden 164
- NIIPSBAD - PSB-Adresse zurückgeben 165
- NIIPSPA - Daten in der SPA ersetzen 165
- NIIPURG - PURG-Aufruf ausgeben 166
- NIIRETRM - Daten in den Nachrichtenbereich verschieben 166
- NIISASD - SENDER- und OUTDEST-Einstellungen ändern 167
- NIIU3962 - Sitzung beenden 167

Dieses Kapitel beschreibt die Serviceprogramme der Natural IMS TM-Schnittstelle.

Einführung in die Serviceprogramme der Natural IMS TM-Schnittstelle

Zweck der Serviceprogramme der Natural-IMS-TM-Schnittstelle

Serviceprogramme sind Natural-Subprogramme, die Natural unter IMS TM mit zusätzlicher Funktionalität ausstatten. Sie können sie aus einem Natural-Programm heraus mit einem Standard-`CALLNAT`-Statement aufrufen.

Speicherort der Serviceprogramme

Die Serviceprogramme werden in der Library `SYSEXTP` zur Verfügung gestellt. Sie müssen sie in die Library `SYSTEM` oder in die Steplib kopieren. Natural-Beispielprogramme zum Aufrufen der Serviceprogramme finden Sie ebenfalls in der Library `SYSEXTP`.

Allgemeine Rückgabecodes

Der letzte Parameter in jedem Serviceprogramm ist der Rückgabecode (Return Code). Format/Länge: (I4).

Die folgenden Rückgabecode-Werte sind für alle Serviceprogramme gleich:

0	OK
-1	Nicht unterstützte Funktion. Dies ist ein interner Fehler. Wenden Sie sich an den Support.

Spezifische Rückgabecode-Werte finden Sie in den nachfolgenden Beschreibungen der einzelnen Serviceprogramme.

Fehlerbehandlung

Wenn ein Fehler auftritt, wird entweder eine Natural-Fehlermeldung ausgegeben oder die Sitzung wird mit einer Natural-IMS-TM-Fehlermeldung beendet. Siehe *Natural under IMS TM Error Messages* in der Natural *Messages and Codes*-Dokumentation.

Beschreibung der Serviceprogramme der Natural IMS TM-Schnittstelle

Die folgenden Serviceprogramme werden beschrieben:

NIIBRCST | NIICMD | NIIDEFT | NIIDEFTX | NIIDIRT | NIIDIRTX | NIIEMOD | NIIGCMD | NIIGMSG | NIIGSEG | NIIGSPA | NIIIMSIN | NIIISRTF | NIIISRTM | NIIPCBAD | NIIPCOM | NIIPMSG | NIIPSBAD | NIIPSPA | NIIPURG | NIIRETRM | NIISASD | NIIU3962

NIIBRCST - Übergebene Nachricht an Terminal senden

Sendet die übergebene Nachricht an das angegebene Terminal unter Verwendung des im Parameter `MOD_name` angegebenen Nachrichtenausgabe-Deskriptors.

Die folgenden Parameter werden bereitgestellt:

Parameter	Format/Länge
`Terminal_name`	(A8)
`Message`	(A1/1:V)
`Message_length`	(I4)
`MOD_name`	(A8)
`Return_code`	(I4)

Spezifische Rückgabecode-Werte: Keine.

Beispielprogramm: `NIPGMSG`

NIICMD - IMS TM-Kommando an IMS TM übergeben

Übergibt das angegebene IMS TM-Kommando an IMS TM. Liegt eine Antwort vor, wird sie in den vorgesehenen Antwortbereich verschoben. Passt die Antwort nicht in den Antwortbereich, wird sie abgeschnitten und der Rückgabecode auf 4 gesetzt.

Die folgenden Parameter werden bereitgestellt:

Parameter	Format/Länge	Typ
Command	(A1/1:V)	Eingabe
Command_length	(I4)	Eingabe
Reply_area	(A1/1:V)	Eingabe/Ausgabe
Reply_area_length	(I4)	Eingabe
Reply_length	(I4)	Ausgabe
Status_code	(A2)	Ausgabe
Return_code	(I4)	Ausgabe

Spezifische Rückgabecode-Werte: 4 (Antwort abgeschnitten)

Beispielprogramm: NIPCMD

NIIDEFT - Aufgeschobenen Wechsel zu Natural-Transaktionscode vorbereiten

Bereitet einen aufgeschobenen Wechsel zu dem angegebenen Natural-Transaktionscode vor. Bei der nächsten Terminal-E/A wird die Ausgabe an das Terminal gesendet und die nächste Eingabe von diesem Terminal wird von dem im Parameter Transaction_code angegebenen Transaktionscode verarbeitet.

Die folgenden Parameter werden bereitgestellt:

Parameter	Format/Länge	Typ
Transaction_code	(A8)	Eingabe
Return_code	(I4)	Ausgabe

Spezifische Rückgabecode-Werte: Keine.

Beispielprogramm: NIPDEFT

NIIDEFTX - Aufgeschobenen Wechsel zu nicht-Natural-Transaktionscode vorbereiten

Bereitet einen aufgeschobenen Wechsel zu einem Nicht-Natural-Transaktionscode vor. Bei der nächsten Terminal-E/A wird die Ausgabe an das Terminal mit dem in MOD_name angegebenen Namen gesendet und die nächste Eingabe von diesem Terminal wird von dem im Parameter Transaction_code angegebenen

Ist `Suspend_flag` auf `Y` gesetzt, wird die Natural-Sitzung unterbrochen und kann später wieder aufgenommen werden. Wenn die Natural-Sitzung wieder aufgenommen wird, wird zunächst der letzte Natural-Bildschirm ausgegeben.

Wenn das `Suspend_flag` auf `Y` gesetzt ist, dürfen Sie nicht von einer konversationellen Natural-Sitzung zu einem nicht-konversationellen Transaktionscode wechseln. In diesem Fall wird eine Natural-Fehlermeldung ausgegeben.

Die folgenden Parameter werden bereitgestellt:

Parameter	Format/Länge	Typ
`Transaction_code`	(A8)	Eingabe
`Transaction_type`	(A4)	Eingabe Mögliche Werte: `CONV` für konversationell `NONC` für nicht-konversationell
`Suspend_flag`	(A1)	Eingabe Mögliche Werte: `Y` - die Natural-Sitzung wird unterbrochen, sonst wird die Natural-Sitzung beendet.
`MOD_name`	(A8)	Eingabe
`Message`	(A1/1:V)	Eingabe
`Message_length`	(I4)	Eingabe
`Return_code`	(I4)	Ausgabe

Spezifische Rückgabecode-Werte: Keine.

Beispielprogramm: `NIPDEFTX`

NIIDIRT - Direkten Wechsel zu Natural-Transaktionscode vorbereiten

Bereitet einen direkten Wechsel zu einem angegebenen Natural-Transaktionscode vor. Beim nächsten Terminal Write wird das Kommando `CHNG` an den angegebenen Transaktionscode ausgegeben und der Natural-Bildschirm wird mit dem alternativen TP PCB eingefügt.

Wenn Sie von einer konversationellen Natural-Sitzung zu einer nicht-konversationellen wechseln, wird die Konversation beendet und eine Dummy-Nachricht mit dem `MOD_Name` `NIIMODNC` eingefügt. Diese Nachricht hebt den Schreibschutz des Bildschirms vorübergehend auf und wird daher durch den ersten Bildschirm der nicht-konversationellen Natural-Sitzung überschrieben.

Parameter	Format/Länge	Typ
Transaction_code	(A8)	Eingabe
Return_code	(I4)	Ausgabe

Spezifische Rückgabecode-Werte: Keine.

Beispielprogramm: NIPDIRT

NIIDIRTX - Direkten Wechsel zum Transaktionscode vorbereiten

Bereitet einen direkten Wechsel zu dem angegebenen Transaktionscode vor. Beim nächsten Terminal Write wird der CHNG-Aufruf für den neuen Transaktionscode ausgegeben, und die Nachricht und/oder die SPA werden mit dem alternativen TP PCB eingefügt. Der Transaktionstyp definiert die Art des neuen Transaktionscodes.

- Wenn Sie von einem konversationellen Transaktionscode zu einem nicht-konversationellen Transaktionscode wechseln, wird die Konversation beendet, indem eine Dummy-Nachricht mit dem MOD_name NIIMODN ausgegeben wird, die den Schreibschutz des Bildschirms vorübergehend aufhebt, so dass er durch den Bildschirm überschrieben wird, der von dem nicht-konversationellen Transaktionscode ausgegeben wird.
- Wenn Suspend_flag auf Y gesetzt ist, wird die Natural-Sitzung unterbrochen und kann zu einem späteren Zeitpunkt wieder aufgenommen werden. Wenn die Natural-Sitzung wieder aufgenommen wird, wird der letzte Natural-Bildschirm ausgegeben.

  Wenn Suspend_flag auf Y gesetzt ist, dürfen Sie nicht von einem konversationellen Natural zu einem nicht konversationellen Transaktionscode wechseln. In diesem Fall wird eine Natural-Fehlermeldung ausgegeben.
- Wenn die Nachrichtenlänge (Message_length) auf Null gesetzt ist, wird überhaupt keine Nachricht eingefügt. Dies ist jedoch nur möglich, wenn Sie zu einem konversationellen Transaktionscode wechseln.

Die folgenden Parameter werden bereitgestellt:

Parameter	Format/Länge	Typ
Transaction_code	(A8)	Eingabe
Transaction_type	(A4)	Eingabe Mögliche Werte: CONV für konversationellen Transaktionscode NONC für nicht-konversationellen Transaktionscode

Parameter	Format/Länge	Typ
Suspend_flag	(A1)	Eingabe Mögliche Werte: Y - die Natural-Sitzung wird unterbrochen, sonst wird die Natural-Sitzung beendet.
Message	(A1/1:V)	Eingabe
Message_length	(I4)	Eingabe
Return_code	(I4)	Ausgabe

Spezifische Rückgabecode-Werte: Keine.

Beispielprogramm: NIPDIRTX

NIIEMOD - Einstellung des Modul-Ausgabe-Deskriptors ändern

Ändert die aktuelle Einstellung des Modul-Ausgabe-Deskriptors, der beim Einfügen der letzten Nachricht in einer Natural-Sitzung verwendet wird, und setzt ihn auf den im Parameter MOD_name angegebenen Wert.

Die folgenden Parameter werden bereitgestellt:

Parameter	Format/Länge	Typ
MOD_name	(A8)	Eingabe
Return_code	(I4)	Ausgabe

Beispielprogramm: NIPEMOD

NIIGCMD - Nächstes Antwortsegment des vorherigen IMS TM-Kommandos abrufen

Ruft das nächste Antwortsegment eines zuvor erteilten IMS TM-Kommandos ab. Die Länge der Antwort wird über den Parameter Reply_length zurückgegeben. Wenn die Antwort nicht in den Antwortbereich passt, wird die Antwort abgeschnitten und der Rückgabecode-Wert 4 ausgegeben.

Die folgenden Parameter werden bereitgestellt:

Parameter	Format/Länge	Typ
Reply_area	(A1/1:V)	Eingabe/Ausgabe
Reply_area_length	(I4)	Eingabe
Reply_length	(I4)	Ausgabe
Status_code	(A2)	Ausgabe
Return_code	(I4)	Ausgabe

Spezifische Rückgabecode-Werte: 4 (Antwort abgeschnitten)

Beispielprogramm: NIPCMD

NIIGMSG - Erstes Segment der nächsten Nachricht abrufen

Ruft das erste Segment der nächsten Nachricht aus der Nachrichtenwarteschlange ab, indem ein GU ausgegeben wird. Der Nachrichtenbereich enthält die abgerufene Nachricht einschließlich der führenden LLZZ-Bytes. Befinden sich keine Nachrichten in der Nachrichtenwarteschlange, wird LLZZ auf Null gesetzt.

Die folgenden Parameter werden bereitgestellt:

Parameter	Format/Länge	Typ
Message_area	(A1/1:V)	Ausgabe
Message_area_length	(I4)	Eingabe
Return_code	(I4)	Ausgabe

Spezifische Rückgabecode-Werte: Keine.

Beispielprogramme: NIPGMSG, NIPGSEG

NIIGSEG - Nächstes Segment der Eingabenachricht abrufen

Ruft das nächste Segment der Eingabenachricht ab, indem ein GN-Aufruf ausgeführt wird. Der Nachrichtenbereich enthält die abgerufene Nachricht einschließlich der führenden LLZZ-Bytes. Gibt es keine weiteren Nachrichtensegmente in der aktuellen Nachricht, wird LLZZ auf Null gesetzt.

Die folgenden Parameter werden bereitgestellt:

Parameter	Format/Länge	Typ
Message_area	(A1/1:V)	Ausgabe
Message_area_length	(I4)	Eingabe
Return_code	(I4)	Ausgabe

Spezifische Rückgabecode-Werte: Keine.

Beispielprogramm: NIPGSEG

NIIGSPA - Daten ab SPA-Beginn abrufen

Ruft aus der SPA Daten ab, die am angegebenen Offset und in der angegebenen Länge beginnen.

Die folgenden Parameter werden bereitgestellt:

Parameter	Format/Länge	Typ
Offset	(I4)	Eingabe
Length	(I4)	Eingabe
Area	(A1/1:V)	Eingabe/Ausgabe
Return_code	(I4)	Ausgabe

Spezifische Rückgabecode-Werte: 4

Die abgerufenen Daten befinden sich ganz oder teilweise in dem für Natural reservierten Teil der SPA.

Beispielprogramm: NIPGSPA

NIIIMSIN - IMS TM-Umgebungsinformationen abrufen

Ruft die IMS TM-Umgebungsinformationen über den Aufruf INQY ENVIRON ab. Wenn Sie eine Reply_area_length kleiner als 102 angeben, wird die Antwort abgeschnitten und Sie erhalten den Rückgabecode X'0100' mit dem Reasoncode X'000C'.

Die folgenden Parameter werden bereitgestellt:

Parameter	Format/Länge	Typ
Reply_area	(A1/1:V)	Ausgabe
Reply_area_length	(I4)	Eingabe
Return_code	(I4)	Ausgabe

Spezifische Rückgabecode-Werte: *nnxx*

Bedeutung:

nn: Die ersten 2 Bytes enthalten den Rückgabecode von AIB.

xx: Die zweiten 2 Bytes enthalten den AIB Reason Code. AIB steht für "Application Interface Block" und wird beim Aufruf von IMS TM über die AIBTDLI-Schnittstelle verwendet.

Beispielprogramm: NIPIMSIN

NIIISRTF - Mehrere Segmente enthaltende Nachrichten erstellen

Erzeugt mehrere Segmente enthaltende Nachrichten. NIIISRTF führt den CHNG-Aufruf für das angegebene Ziel aus und fügt das erste Nachrichtensegment ein, ohne einen PURG-Aufruf durchzuführen. Weitere Nachrichtensegmente können mit NIIISRTM eingefügt werden. Die Nachricht muss mit NIIPURG abgeschlossen werden. Die LLZZ-Bytes werden vom Service-Modul erzeugt.

Die folgenden Parameter werden bereitgestellt:

Parameter	Format/Länge	Typ
Destination	(A8)	Eingabe
Message	(A1/1:V)	Eingabe
Message_length	(I4)	Eingabe
Return_code	(I4)	Ausgabe

Spezifische Rückgabecode-Werte: Keine.

Beispielprogramm: NIPISRTM

NIIISRTM - Nachrichtensegment in Nachrichtenwarteschlange einfügen

Fügt das nächste Nachrichtensegment in die Nachrichtenwarteschlange ein, ohne einen `CHNG`- oder `PURG`-Aufruf durchzuführen. Die `LLZZ`-Bytes werden durch das Servicemodul erstellt.

Die folgenden Parameter werden bereitgestellt:

Parameter	Format/Länge	Typ
`Message`	(A1/1:V)	Eingabe
`Message_length`	(I4)	Eingabe
`Return_code`	(I4)	Ausgabe

Spezifische Rückgabecode-Werte: Keine.

Beispielprogramm: `NIPISRTM`

NIIPCBAD - PSB-Name und PCB-Adresse zurückgeben

Gibt den Namen des aktuell geplanten PSB und die Adresse des durch den logischen Namen identifizierten PCB zurück. Wenn der logische PCB-Name nicht in der Transaktionscode-Tabelle definiert ist, wird eine Natural-Fehlermeldung ausgegeben.

Die folgenden Parameter werden bereitgestellt:

Parameter	Format/Länge	Typ
`PSB_name`	(A8)	Ausgabe
`Logical_PCB_name`	(A8)	Eingabe
`PCB_address`	(B4)	Ausgabe
`Return_code`	(I4)	Ausgabe

Spezifische Rückgabecode-Werte: Keine.

Beispielprogramm: `NIPPCBAD`

NIIPCOM - Daten in Antwortbereich verschieben

Verschiebt die im Datenbereich bereitgestellten Daten in den im NIIBOOTS-Aufruf angegebenen Antwortbereich am angegebenen Offset und in der angegebenen Länge. NIIPCOM kann nur aus der Server-Umgebung aufgerufen werden.

Die folgenden Parameter werden bereitgestellt:

Parameter	Format/Länge	Typ
Offset	(I4)	Eingabe
Data_area	(A1/1:V)	Eingabe
Length	(I4)	Eingabe
Return_code	(I4)	Ausgabe

Spezifische Returncode-Werte: 4 (aufrufende Umgebung, Nicht-Server-Umgebung)

Beispielprogramm: NIPPCOM

NIIPMSG - Nachricht senden

Sendet eine Nachricht mit einem gegebenen MOD_name an das Ziel, das durch den E/A PCB repräsentiert wird. Die Nachricht wird aus dem Nachrichtenbereich und in der angegebenen Nachrichtenbereichslänge entnommen. Der Nachrichtenbereich darf die führenden LLZZ-Bytes nicht enthalten. Auf diese Weise können Sie MFS-formatierte Ausgabenachrichten an den Absender der Eingabenachricht zurücksenden.

Die folgenden Parameter werden bereitgestellt:

Parameter	Format/Länge	Typ
Message	(A1/1:V)	Eingabe
Message_length	(I4)	Eingabe
MOD_name	(A8)	Eingabe
Return_code	(I4)	Ausgabe

Spezifische Rückgabecode-Werte: Keine.

Beispielprogramm: NIPPMSG

NIIPSBAD - PSB-Adresse zurückgeben

Gibt die Adresse der PSB zurück, die die Adresse der PCB-Adressliste ist.

Die folgenden Parameter werden bereitgestellt:

Parameter	Format/Länge	Typ
PSB_address	(B4)	Ausgabe
Return_code	(I4)	Ausgabe

Spezifische Rückgabecode-Werte: Keine.

Beispielprogramm: NIPBOOTS

NIIPSPA - Daten in der SPA ersetzen

Ersetzt die Daten in der SPA am angegebenen Offset in der angegebenen Länge durch die im Datenbereich angegebenen Daten.

Die folgenden Parameter werden bereitgestellt:

Parameter	Format/Länge	Typ
Offset	(I4)	Eingabe
Length	(I4)	Eingabe
Data_area	(A1/1:V)	Eingabe
Return_code	(I4)	Ausgabe

Spezifische Rückgabecode-Werte: Keine.

Der Versuch, den Header der SPA (die ersten 14 Bytes) und/oder Daten, die sich im für Natural reservierten Bereich befinden, zu überschreiben, wird abgelehnt und es wird eine Natural-Fehlermeldung ausgegeben.

Beispielprogramm: NIPPSPA

NIIPURG - PURG-Aufruf ausgeben

Gibt einen PURG-Aufruf aus.

Der folgende Parameter wird bereitgestellt:

Parameter	Format/Länge	Typ
Return_code	(I4)	Ausgabe

Spezifische Rückgabecode-Werte: Keine.

Beispielprogramm: NIPISRTM

NIIRETRM - Daten in den Nachrichtenbereich verschieben

Verschiebt Daten aus der Eingabenachricht ab dem angegebenen Offset und in der angegebenen Länge in den angegebenen Nachrichtenbereich.

Der Offset wird ab den LLZZ-Bytes berechnet.

Die folgenden Parameter werden bereitgestellt:

Parameter	Format/Länge	Typ
Offset	(I4)	Eingabe
Length	(I4)	Eingabe
Message_area	(A1/1:V)	Eingabe/Ausgabe
Return_code	(I4)	Ausgabe

Spezifische Rückgabecode-Werte: Keine.

Beispielprogramm: NIPRETRM

NIISASD - SENDER- und OUTDEST-Einstellungen ändern

Ändert die aktuelle Einstellung der Natural-Profilparameter `SENDER` und `OUTDEST`.

Die folgenden Parameter werden bereitgestellt:

Parameter	Format/Länge	Typ
`Sender`	(A8)	Eingabe
`Outdest`	(A8)	Eingabe
`Return_code`	(I4)	Ausgabe

Spezifische Rückgabecode-Werte: Keine.

Beispielprogramm: `NIPNTRD`

NIIU3962 - Sitzung beenden

Beendet die Sitzung mit dem Benutzer-Abendcode U3962 und erzeugt einen Dump.

Der folgende Parameter wird bereitgestellt:

Parameter	Format/Länge	Typ
`Return_code`	(I4)	Ausgabe

Spezifische Rückgabecode-Werte: Keine.

Beispielprogramm: `NIPU3962`

26 Natural unter IMS TM - Servicemodule

- Zweck der Servicemodule 170
- Beschreibungen der Servicemodule 170
- CMCMMND - IMS TM-Operator-Kommandos ausgeben 170
- CMDEFSW - Aufgeschobenen Transaktionswechsel zu einem Natural-Transaktionscode ausführen 171
- CMDEFSWX - Aufgeschobenen Transaktionswechsel zu einem Nicht-Natural-Transaktionscode durchführen 171
- CMDIRNMX - Wechsel zu einer anderen konversationellen Transaktion ohne Nachricht durchführen 172
- CMDIRNMZ - Wechsel zu einer anderen konversationellen Transaktion ohne Nachricht durchführen 172
- CMDIRSWX - Wechsel zu einer anderen konversationellen Transaktion mit Nachricht durchführen 173
- CMDIRSWZ - Wechsel zu einer anderen konversationellen Transaktion mit Nachricht durchführen 174
- CMDISPCB - PCB-Inhalt holen 175
- CMEMOD - MOD-Namen dynamisch ändern 175
- CMGETMSG - Nächste Nachricht lesen 176
- CMGETSEG - Nächstes Segment lesen 176
- CMGETSPA - Daten aus SPA übertragen 177
- CMGSEGO - Nächstes Segment lesen 177
- CMIMSID - MVS-Subsystemkennung abfragen 178
- CMIMSINF - Informationen zur Systemumgebung abfragen 178
- CMPCBADR - PCB-Adresse zurückgeben 179
- CMPRNTR - Standard-Hardcopy-Ziel ändern 179
- CMPUTMSG - Ausgabemeldung in den IO-PCB einfügen 180
- CMPUTSPA - Daten in die SPA verschieben 180
- CMQTRAN - Inhalt des aktuellen Transaktionscode-Tabelleneintrags zurückgeben 181
- CMQUEUE - Nachricht in alternativen PCB einfügen 181
- CMQUEUEX - Vollständige Kontrolle über den Nachrichteninhalt erlangen 182
- CMSNFPRT - Logischen Gerätenamen setzen 183
- CMSVC13D - Natural-Sitzung abbrechen 183
- CMTRNSET - SPA über den alternativen PCB einfügen 183
- NIIDDEFS - Aufgeschobenen Wechsel zu einer Fremdtransaktion durchführen 184
- NIIDPURG - Multisegment-Nachricht versenden 184
- NIIDQUMS - Multisegment-Nachrichten erstellen 185
- NIIDSETT - Fremdtransaktionscode abfragen 185

Dieses Kapitel beschreibt die Servicemodule der Natural IMS TM-Schnittstelle.

Zweck der Servicemodule

Servicemodule führen IMS TM-spezifische Funktionen aus. Sie können aus einem Natural-Programm heraus über die standardmäßige Natural-Call-Schnittstelle aufgerufen werden. Beispielprogramme werden durch ein Natural INPL in die Library `SYSEXTP` geladen.

Beschreibungen der Servicemodule

Dieser Abschnitt enthält detaillierte Beschreibungen der Servicemodule in alphabetischer Reihenfolge. Dazu gehört auch eine Liste der zur Verfügung stehenden Parameter und der Name des modulrelevanten Beispielprogramms.

CMCMMND - IMS TM-Operator-Kommandos ausgeben

Das Modul `CMCMMND` gibt IMS TM-Operator-Kommandos aus und sendet die Antwortsegmente an das Natural-Anwenderprogramm zurück.

Die folgenden Parameter werden bereitgestellt:

Name	Format/Länge	Typ	Kommentar
`Command`		Eingabe	
`Command length`	(B4)	Eingabe	
`Reply`		Ausgabe	
`Length of reply area`	(B4)	Eingabe	

Das im Kommandobereich enthaltene Operator-Kommando wird mit der angegebenen Länge an IMS TM ausgegeben.

Wenn der Benutzer eine Antwortlänge ungleich Null eingestellt hat, werden alle Antwortsegmente von IMS TM über die maximal verfügbare Länge in den Antwortbereich verschoben. Wenn der Antwortbereich mindestens zwei Bytes lang ist, enthalten die ersten beiden Bytes den IMS TM-Statuscode, nachdem der Kommandoaufruf erfolgt ist. Die beiden äußersten rechten Bytes des `REPLGTH`-Feldes enthalten die effektive Länge der gesamten Antwort, die in das `REPLY`-Feld übertragen wurde.

Wenn die Antwort von IMS TM gekürzt werden muss, wird dies durch Setzen von `X'80'` im ganz linken Byte des `REPLGTH`-Feldes angezeigt.

Beispielprogramm: NIPSCMND

CMDEFSW - Aufgeschobenen Transaktionswechsel zu einem Natural-Transaktionscode ausführen

Das Modul CMDEFSW führt einen verzögerten Transaktionswechsel zu einem Natural-Transaktionscode durch.

Der folgende Parameter wird bereitgestellt:

Name	Format/Länge	Typ	Kommentar
Trancode		Eingabe	

Bei der nächsten Terminal-E/A wird die Ausgabe an das Terminal gesendet und die nächste Eingabe von diesem Terminal wird durch den als Parameternachricht übergebenen Transaktionscode verarbeitet.

CMDEFSWX - Aufgeschobenen Transaktionswechsel zu einem Nicht-Natural-Transaktionscode durchführen

Das Modul CMDEFSWX führt einen aufgeschobenen Wechsel zu einem Nicht-Natural-Transaktionscode durch.

Die folgenden Parameter werden bereitgestellt:

Name	Format/Länge	Typ	Kommentar
Trancode		Eingabe	Transaktionscode
Message		Eingabe	Nachricht
Message length		Eingabe	Länge der Nachricht
MOD name		Eingabe	MOD-Name

Bei der nächsten Terminal-E/A wird die gegebene Nachricht mit dem gegebenen MOD_name eingefügt und die Natural-Sitzung wird beendet.

Wenn der neue Transaktionscode ein Natural-Transaktionscode ist, werden die Nachricht und der MOD_name, die als Parameter übergeben wurden, ignoriert und CMDEFSWX funktioniert wie CMDEFSW.

Beispielprogramm: NIPSDEFX.

CMDIRNMX - Wechsel zu einer anderen konversationellen Transaktion ohne Nachricht durchführen

Das Modul CMDIRNMX hat die gleiche Funktionalität wie CMDIRSWX, mit dem Unterschied, dass keine Nachricht auf dem alternativen PCB eingefügt wird. Der einzige Parameter, den Sie angeben müssen, ist daher Trancode.

Der folgende Parameter wird bereitgestellt:

Name	Format/Länge	Typ	Kommentar
Trancode		Eingabe	Transaktionscode

CMDIRNMX kann auch verwendet werden, um direkt zu einem anderen Natural-Transaktionscode zu wechseln, da in diesem Fall standardmäßig die Taste CLEAR als Eingabenachricht an Natural übergeben wird.

Wenn Sie zu einem Nicht-Natural-Transaktionscode wechseln wollen, wird dringend empfohlen, stattdessen das Statement TERMINATE in Verbindung mit dem Servicemodul CMTRNSET zu verwenden.

```
CALL 'CMTRNSET' TRANCODE /* set transaction code */
TERMINATE                /* terminate Natural and call TRANCODE */
```

CMDIRNMZ - Wechsel zu einer anderen konversationellen Transaktion ohne Nachricht durchführen

Das Modul CMDIRNMZ hat die gleiche Funktionalität wie CMDIRSWZ, mit dem Unterschied, dass keine Nachricht in den alternativen PCB eingefügt wird. Der einzige Parameter, den Sie angeben müssen, ist daher Trancode.

Der folgende Parameter wird bereitgestellt:

Name	Format/Länge	Typ	Kommentar
Trancode		Eingabe	Transaktionscode

CMDIRSWX - Wechsel zu einer anderen konversationellen Transaktion mit Nachricht durchführen

Das Modul CMDIRSWX führt einen direkten Wechsel zu einer anderen konversationellen Transaktion durch und gibt eine Nachricht an, die an diese neue Transaktion weitergegeben werden soll.

Die folgenden Parameter werden bereitgestellt:

Name	Format/Länge	Typ	Kommentar
Trancode		Eingabe	Transaktionscode
Message		Eingabe	Nachricht
Message length	(B4)	Eingabe	Länge der Nachricht

Bei der nächsten Terminal-E/A wird ein Änderungsaufruf gegen den alternativen PCB ausgeführt, um dessen Ziel auf den Wert des Feldes Trancode zu setzen. Die SPA und die Nachricht werden dann in den alternativen PCB eingefügt.

Der neue Transaktionscode wird daraufhin überprüft, ob es sich um einen Natural- oder einen Nicht-Natural-Transaktionscode handelt.

Im Falle eines Nicht-Natural-Transaktionscodes wird die Natural-Sitzung abgebrochen.

Bei einem Natural-Transaktionscode wird die Taste CLEAR als Eingabenachricht an Natural übergeben, was bedeutet, dass Natural so reagiert, als ob der Terminalbenutzer die Taste CLEAR gedrückt hätte. Der Typ des neuen Transaktionscodes wird automatisch übernommen.

Wenn Sie zu einem Nicht-Natural-Transaktionscode wechseln wollen, wird dringend empfohlen, stattdessen das TERMINATE-Statement in Verbindung mit dem Servicemodul CMTRNSET zu verwenden.

```
CALL 'CMTRNSET' TRANCODE /* set transaction code */
TERMINATE 0 MESSAGE      /* terminate Natural and call TRANCODE with MESSAGE */
```

Die Nachricht MESSAGE wird in der Länge der Natural-Variablen Message an den Transaktionscode Trancode übergeben. Der Rückgabecode des TERMINATE-Statements muss Null sein. Andernfalls wird die Natural-Sitzung mit der Abbruchfehlermeldung NAT9987 abgebrochen und der Transaktionscode-Wechsel findet nicht statt.

Beispielprogramm: NIPSDIRX

CMDIRSWZ - Wechsel zu einer anderen konversationellen Transaktion mit Nachricht durchführen

Das Modul CMDIRSWZ hat die gleiche Funktionalität wie CMDIRSWX.

Die folgenden Parameter werden bereitgestellt:

Name	Format/Länge	Typ	Kommentar
Trancode		Eingabe	Transaktionscode
Message		Eingabe	Nachricht
Message length	(B4)	Eingabe	Länge der Nachricht

Der Unterschied zu CMDIRSWX besteht darin, dass im Falle eines Wechsels zu einem Nicht-Natural-Transaktionscode die aktuelle Natural-Sitzung nicht abgebrochen wird. Dies geschieht in der folgenden Absicht:

- Eine bestimmte Natural-Sitzung gibt die Kontrolle an einen Nicht-Natural-Transaktionscode ab. Die Sitzung wird nicht abgebrochen.
- Die Nicht-Natural-Transaktion führt eine Terminal-E/A durch und schaltet dann zurück zur ursprünglichen Natural-Transaktion und gibt Daten in die SPA weiter.
- Die Natural-Transaktion beginnt keine neue Sitzung, sondern setzt die alte Sitzung dort fort, wo sie sie verlassen hat. Der Roll Slot wird vom Roll Server geholt und die Kontrolle an Natural übergeben, um so mit einer bestehenden Sitzung fortzufahren.

Der Nicht-Natural-Transaktionscode muss die Nachricht LLZZD übermitteln, wobei LL=H'0005', ZZ=X'0000' und D=X'6D' gegenüber Natural simulieren, dass die Taste CLEAR gedrückt wurde. Indem das Natural-Programm für die CLEAR-Taste sensibilisiert wird, kann es erkennen, dass die aufgerufene Nicht-Natural-Transaktion wieder da ist, und es kann die von der Nicht-Natural-Transaktion vorbereiteten Daten zur Verwendung in der nachfolgenden Verarbeitung abrufen.

CMDIRSWZ kann nicht verwendet werden, wenn der Transaktionscode, zu dem gewechselt werden soll, ein Natural-Transaktionscode ist.

Beispielprogramm: NIPSDIFS

CMDISPCB - PCB-Inhalt holen

Das Modul `CMDISPCB` wird verwendet, um den Inhalt eines Program Control Block (PCB) zu ermitteln.

Die folgenden Parameter werden bereitgestellt:

Name	Format/Länge	Typ	Kommentar
`PCB number`	(B4)	Eingabe	Nummer des PCB
`Receiving area`		Ausgabe	Empfangender Bereich
`Area length`	(B4)	Eingabe	Länge des Bereichs

Nach Ausführung des Aufrufs enthält der empfangende Bereich den Inhalt des PCB mit der angeforderten Nummer und in der angeforderten Länge. Es wird geprüft, ob der angeforderte PCB in Ihrer aktuellen PCB-Liste enthalten ist. Der erste PCB ist PCB Nummer 1, der zweite PCB ist PCB Nummer 2, usw. Wenn Sie eine ungültige Nummer angeben, wird das Feld `PCB number` auf `X'FFFFFFFF'` gesetzt und es werden keine weiteren Informationen an Ihr Anwendungsprogramm übergeben.

Beispielprogramm: `NIPSPCBD`

CMEMOD - MOD-Namen dynamisch ändern

Mit dem Modul `CMEMOD` kann bei normaler Beendigung einer Natural-Sitzung der MOD-Name für ein bestimmtes `LTERM` dynamisch geändert werden.

Der folgende Parameter wird bereitgestellt:

Name	Format/Länge	Typ	Kommentar
`MOD name`	(A8)	Eingabe	MOD-Name

Bei einem normalen Sitzungsende fügt die umgebungsabhängige Schnittstelle die Nachricht `X'00060000403F'` in den IOPCB ein und verwendet dabei den MOD-Namen, dessen Wert im Parameter `MOD name` enthalten ist. Damit soll dem Terminalbenutzer ein aussagefähiger Bildschirm (z.B. ein allgemeines Menü) präsentiert werden, damit er seine Arbeit am Terminal fortsetzen kann.

CMGETMSG - Nächste Nachricht lesen

Das Modul CMGETMSG liest die nächste Nachricht aus der Nachrichtenwarteschlange.

Die folgenden Parameter werden bereitgestellt:

Name	Format/Länge	Typ	Kommentar
Message area		Ausgabe	Nachrichtenbereich
Message area length	(B4)	Eingabe	Länge des Nachrichtenbereichs

Die Länge wird überprüft, um festzustellen, ob die empfangene Nachricht in den Nachrichtenbereich passt. Die Nachricht wird einschließlich der LLZZ-Bytes in den Nachrichtenbereich verschoben. Gibt es keine weiteren Nachrichten, wird LL=0 in den Nachrichtenbereich verschoben.

Passt die Nachricht nicht in den Nachrichtenbereich, wird eine entsprechende Fehlermeldung zurückgegeben.

Beispielprogramme: NIPSGETM und NIPSOBMP.

CMGETSEG - Nächstes Segment lesen

Das Modul CMGETSEG liest das nächste Segment der aktuellen Nachricht aus der Nachrichtenwarteschlange.

Die folgenden Parameter werden bereitgestellt:

Name	Format/Länge	Typ	Kommentar
Message area		Ausgabe	Nachrichtenbereich
Message area length	(B4)	Eingabe	Länge des Nachrichtenbereichs

Die Länge wird überprüft, um festzustellen, ob die empfangene Nachricht in den Nachrichtenbereich passt. Das Nachrichtensegment wird einschließlich der LLZZ-Bytes in den Nachrichtenbereich verschoben. Sind keine weiteren Nachrichtensegmente vorhanden, wird LL=0 in den Nachrichtenbereich verschoben.

Passt die Nachricht nicht in den Nachrichtenbereich, wird eine entsprechende Fehlermeldung zurückgegeben.

Alle gelesenen Nachrichtensegmente werden zu einem String verkettet im internen Eingabe-Nachrichtenpuffer aufbewahrt, dessen Größe durch den Schlüsselwort-Subparameter MISIZE des Makros NTIMPSPE festgelegt wird (siehe *Parameter-Referenz*-Dokumentation).

Wenn Sie einen Überlauf des internen Eingabenachrichtenpuffers vermeiden wollen, können Sie anstelle von CMGETSEG das Modul CMGSEGO verwenden.

Beispielprogramm: NIPSOBMP

CMGETSPA - Daten aus SPA übertragen

Das Modul CMGETSPA überträgt die Daten aus der SPA ab dem angegebenen Offset in der gewünschten Länge in den Empfangsbereich.

Die folgenden Parameter werden bereitgestellt:

Name	Format/Länge	Typ	Kommentar
Offset	(B4)	Eingabe	Versatz
Length	(B4)	Eingabe	Länge
Area	(B4)	Ausgabe	Bereich

Programmbeispiele: NIPSGSPA und NIPSPSPA

CMGSEGO - Nächstes Segment lesen

Das Modul CMGSEGO liest das nächste Segment der aktuellen Nachricht aus der Nachrichtenwarteschlange.

Die folgenden Parameter werden bereitgestellt:

Name	Format/Länge	Typ	Kommentar
Message area		Ausgabe	Nachrichtenbereich
Message area length	(B4)	Eingabe	Länge des Nachrichtenbereichs

Die Länge wird überprüft, um festzustellen, ob die empfangene Nachricht in den Nachrichtenbereich passt. Das Nachrichtensegment wird einschließlich der LLZZ-Bytes in den Nachrichtenbereich verschoben. Sind keine weiteren Nachrichtensegmente vorhanden, wird LL=0 in den Nachrichtenbereich verschoben.

Passt die Nachricht nicht in den Nachrichtenbereich, wird eine entsprechende Fehlermeldung zurückgegeben.

Nur das erste und das aktuelle Nachrichtensegment werden im internen Eingabepuffer aufbewahrt, dessen Größe durch den Subparameter MISIZE des Makros NTIMPSPE festgelegt wird (siehe *Para-*

meter-Referenz-Dokumentation). Wenn Sie alle Nachrichtensegmente behalten wollen, müssen Sie anstelle von CMGSEGO das Modul CMGETSEG verwenden.

Beispielprogramm: NIPSOBMP

CMIMSID - MVS-Subsystemkennung abfragen

Mit dem Modul CMIMSID können Natural-Programme die MVS-Subsystemkennung des IMS TM-Systems abfragen, in dem sie zurzeit eingeplant sind.

Der folgende Parameter wird bereitgestellt:

Name	Format/Länge	Typ	Kommentar
IMSID	(A4)	Ausgabe	Subsystemkennung

Nach der Ausführung des Aufrufs enthält das Feld IMSID die MVS-Subsystemkennung des IMS TM-Systems, für das Sie zurzeit eingeplant sind.

Das Modul CMIMSID hängt von einem internen IMS TM-Kontrollblock ab. Es handelt sich daher um eine IMS TM-release-abhängige Funktion, die aktualisiert wird, wann immer dies möglich ist.

CMIMSINF - Informationen zur Systemumgebung abfragen

Das Modul CMIMSINF liefert Informationen zur Systemumgebung.

Die folgenden Parameter werden bereitgestellt:

Name	Format/Länge	Typ	Kommentar
IMSID	(A4)	Ausgabe	Die IMS TM-Kennung.
SUFFIX	(A2)	Ausgabe	Das Preload-Suffix.
APPLGNAM	(A8)	Ausgabe	Der Gruppenname der Anwendung.
APPLNAM	(A8)	Ausgabe	Der Name der Anwendung.
NRENT	(B4)	Ausgabe	Die Anzahl der vorab geladenen reentranten Module.
NNONR	(B4)	Ausgabe	Die Anzahl der vorab geladenen nicht-reentranten Module.

CMIMSINF ist ebenfalls ein IMS TM Release-abhängiges Modul.

Beispielprogramm: NIPSINF

CMPCBADR - PCB-Adresse zurückgeben

Das Modul `CMPCBADR` gibt die Adresse eines PCB zurück, der durch einen logischen Namen identifiziert wird. Der PSB-Name wird ebenfalls an das Natural-Programm zurückgegeben.

Die folgenden Parameter werden bereitgestellt:

Name	Format/Länge	Typ	Kommentar
`PSB name`	(A8)	Ausgabe	Der PSB-Name.
`PCB name`	(A8)	Eingabe	Der PSB-Name.
`PCB address`	(B4)	Ausgabe	Die PSB-Adresse.

Nach der Ausführung des Aufrufs enthält das Feld `PCBADR` die Adresse des PCB, der im Tabellenmodul durch den logischen Namen `PCBNAME` in dem Tabelleneintrag identifiziert wird, der dem aktuell eingeplanten Transaktionscode entspricht. Existiert der logische Name für diesen Transaktionscode nicht, wird im Feld `PCBADR` der Wert `X'FFFFFFFF'` zurückgegeben. In jedem Fall enthält das Feld `PSBNAME` den Namen des aktuell eingeplanten PSB.

Beispielprogramm: `NIPSPCBA`

CMPRNTR - Standard-Hardcopy-Ziel ändern

Das Modul `CMPRNTR` ändert das vom Modul `NIIIMSHC` gesetzte Standard-Hardcopy-Ziel in den als Parameter übergebenen Wert.

Der folgende Parameter wird bereitgestellt:

Name	Format/Länge	Typ	Kommentar
`Destination`	(A8)	Eingabe	Das Hardcopy-Ziel.

Das Modul `CMPRNTR` wird nur aus Kompatibilitätsgründen bereitgestellt. Verwenden Sie stattdessen das Natural-Statement `SET CONTROL` *`hdest-id`*.

CMPUTMSG - Ausgabemeldung in den IO-PCB einfügen

Das Modul `CMPUTMSG` kann verwendet werden, um eine beliebige Ausgabemeldung mit einer gegebenen Länge unter Verwendung eines gegebenen MFS `MOD name` in den IO-PCB einzufügen. Auf diese Weise können Sie MFS-formatierte Ausgabemeldungen an den Absender der Eingabenachricht zurücksenden.

`CMPUTMSG` entnimmt die in der Nachrichtenlänge angegebene Anzahl von Bytes aus dem Nachrichtenbereich und fügt sie mit dem angegebenen `MOD name` in die Nachrichtenwarteschlange ein. Es gibt keine Beschränkung bezüglich der Länge der Nachricht, außer dass sie in den Eingabe-Nachrichtenbereich der umgebungsabhängigen Schnittstelle passen muss.

Die folgenden Parameter werden bereitgestellt:

Name	Format/Länge	Typ	Kommentar
`Message area`		Eingabe	Nachrichtenbereich
`Message length`	(B4)	Eingabe	Nachrichtenlänge
`MOD name`		Eingabe	MOD-Name

Wird in der IO-PCB ein nicht leerer Statuscode zurückgegeben, wird die Natural-Fehlermeldung NAT8272 ausgegeben, die den Statuscode als variablen Teil enthält.

CMPUTSPA - Daten in die SPA verschieben

Das Modul `CMPUTSPA` verschiebt die Daten mit der angegebenen Länge und dem angegebenen Offset in die SPA.

Die folgenden Parameter werden bereitgestellt:

Name	Format/Länge	Typ	Kommentar
`Offset`	(B4)	Eingabe	Versatz
`Length`	(B4)	Eingabe	Länge
`Data`		Eingabe	Daten

Es wird geprüft, ob der angegebene Offset in die Natural Reserved Area (NRA) innerhalb der SPAs zeigt. Wenn ja, wird der Rückgabecode 4 zurückgegeben.

Beispielprogramm: `NIPSPSPA`

CMQTRAN - Inhalt des aktuellen Transaktionscode-Tabelleneintrags zurückgeben

Das Modul CMQTRAN gibt den Inhalt des aktuellen Eintrags in der Transaktionscode-Tabelle zurück.

Die folgenden Parameter werden bereitgestellt:

Name	Format/Länge	Typ	Kommentar
Transaction code		Ausgabe	Der Transaktionscode, unter dem Sie gerade arbeiten.
Offset	(B2)	Ausgabe	Der Versatz der NRA zu der SPA.
Length	(B2)	Ausgabe	Die Länge der NRA.
Uoffset	(B2)	Ausgabe	Nicht verwendet.
PSB name		Ausgabe	Der Name des geplanten PSB.
Number of PCBs		Ausgabe	Die Anzahl der PCBs, deren Adressen Sie über das Modul CMPCBADR erhalten können.

Die logischen Namen, mit denen Sie im Modul CMPCBADR auf PCBs verweisen können, werden aus Security-Gründen nicht zurückgegeben. Sie sollten sich von Ihrem Systemverwalter darüber informieren lassen, auf welche logischen Namen Sie verweisen dürfen.

Beispielprogramm: NIPSQTRA

CMQUEUE - Nachricht in alternativen PCB einfügen

Das Modul CMQUEUE fügt eine Nachricht in den angegebenen alternativen PCB ein.

Die folgenden Parameter werden bereitgestellt:

Name	Format/Länge	Typ	Kommentar
Destination		Eingabe	Ziel
Message		Eingabe	Nachricht
Message length	(B4)	Eingabe	Nachrichtenlänge
TP PCB number	(B4)	Eingabe	Optional

Dieser Aufruf bewirkt einen sofortigen Änderungsaufruf, mit dem das Ziel des angegebenen alternativen PCB auf den im Feld Destination enthaltenen Wert gesetzt wird, woraufhin die Nachricht in den alternativen PCB mit der angegebenen Nachrichtenlänge eingefügt wird.

Der Transaktionscode wird nach den LLZZ-Bytes mit einer Länge von 8 eingefügt.

Nachdem ein PURGE-Aufruf ausgegeben worden ist, wird die Kontrolle an das nächste Statement im Natural-Programm zurückgegeben.

Die Nachricht kann eine beliebige Länge bis zur Größe des Eingabenachrichtenbereichs haben (normalerweise 8000 minus 12 Byte).

Der zu verwendende alternative PCB wird mit dem letzten, optionalen, Parameter angegeben. Wird beim Aufruf keine TP PCB number angegeben, wird der mit dem Subparameter ALTPCB des Makros NTIMSPT (im Natural-Parametermodul) angegebene alternative TP PCB verwendet.

Beispielprogramm: NIPSQLOA

CMQUEUEX - Vollständige Kontrolle über den Nachrichteninhalt erlangen

Das Modul CMQUEUEX ermöglicht Ihnen die vollständige Kontrolle über den Inhalt einer Nachricht, die in die IMS TM-Eingabewarteschlange gestellt werden soll.

Die folgenden Parameter werden bereitgestellt:

Name	Format/Länge	Typ	Kommentar
Destination		Eingabe	Ziel
Message		Eingabe	Nachricht
Message length	(B4)	Eingabe	Nachrichtenlänge
TP PCB number	(B4)	Eingabe	Optional

Dieser Aufruf bewirkt einen sofortigen Änderungsaufruf, um das Ziel des angegebenen alternativen PCB auf den im Feld Destination enthaltenen Wert zu setzen, woraufhin die Nachricht in den alternativen PCB mit der angegebenen Nachrichtenlänge nach den LLZZ-Bytes eingefügt wird. Der Unterschied zu CMQUEUE besteht darin, dass dort der Transaktionscode *nicht* nach den LLZZ-Bytes eingefügt wird.

Nach einem PURGE-Aufruf wird die Kontrolle an das nächste Statement im Natural-Programm zurückgegeben. Die Nachricht kann eine beliebige Länge bis zur Größe des Eingabenachrichtenbereichs haben (normalerweise 8000 minus 12 Byte).

Der zu verwendende alternative PCB wird mit dem letzten, optionalen, Parameter angegeben. Wird beim Aufruf keine TP PCB-Nummer angegeben, wird der mit dem Subparameter ALTPCB des Makros NTIMSPT (im Natural-Parametermodul) angegebene alternative TP PCB verwendet.

Beispielprogramm: NIPSQUEX

CMSNFPRT - Logischen Gerätenamen setzen

Das Modul `CMSNFPRT` setzt den logischen Namen des Geräts, an das die Natural-Nachrichten während des Online-BMP-Laufs gesendet werden.

Der folgende Parameter wird bereitgestellt:

Name	Format/Länge	Typ	Kommentar
`Printer name`		Eingabe	Name des Druckers

Vor dem Aufruf von `CMSNFPRT` müssen Sie mit dem Natural-Profilparameter `SENDER` das Standard-Ausgabeziel definieren.

Beispielprogramm: `NIPSOBMP`

CMSVC13D - Natural-Sitzung abbrechen

Das Modul `CMSVC13D` bricht die Natural-Sitzung mit dem Benutzerabbruchcode U3962 ab und erzeugt einen Dump.

Parameter: Keine

Beispielprogramm: Keines.

CMTRNSET - SPA über den alternativen PCB einfügen

Wenn die Natural-Sitzung normal beendet wird, führt die Natural IMS TM-Schnittstelle einen direkten Programm-Wechsel zum angegebenen Transaktionscode durch und fügt die SPA über den alternativen PCB ein.

Der folgende Parameter wird bereitgestellt:

Name	Format/Länge	Typ	Kommentar
`Trancode`		Eingabe	Transaktionscode

Beispielprogramm: `NIPSEOSS`

NIIDDEFS - Aufgeschobenen Wechsel zu einer Fremdtransaktion durchführen

Das Modul NIIDDEFS ist ähnlich wie das Modul CMDEFSWX. Wenn Sie mit NIIDDEFS einen aufgeschobenen Wechsel zu einer Fremdtransaktion durchführen, wird die aktuelle Natural-Sitzung wie beim Modul CMDIRSWZ unterbrochen. Die unterbrochene Natural-Sitzung kann jederzeit wieder aufgenommen werden, indem Sie eine Nachricht mit der CLEAR-Taste an Natural zurücksenden.

Die folgenden Parameter werden bereitgestellt:

Name	Format/Länge	Typ	Kommentar
Transaction code		Eingabe	Der Transaktionscode, zu dem gewechselt werden soll.
Message		Eingabe	Die Nachricht, die an den fremden Transaktionscode gesendet werden soll.
Message length	(B4)	Eingabe	Die Nachrichtenlänge.
MOD name	(A8)	Eingabe	Der MOD-Name.
Transaction type	(A4)	Eingabe	Eine Variable im Format A4, die den String CONV enthält, wenn die fremde Transaktion konversationell ist, und den String NONC, wenn die fremde Transaktion nicht konversationell ist.

Rückgabecodes:

0	OK
4	Die Nachrichtenlänge ist größer als die Größe des in der Umgebungstabelle definierten Nachrichtenbereichs.
8	Sie haben versucht, einen aufgeschobenen Wechsel mit Suspend von einer konversationellen Natural- zu einer nicht-konversationellen Fremdtransaktion durchzuführen, was nicht möglich ist.
12	Der fünfte Parameter ist ungültig; er enthält weder CONV noch NONC.

Beispielprogramm: NIPSDEFS

NIIDPURG - Multisegment-Nachricht versenden

Das Modul NIIDPURG hat keine Parameter. Es führt einen PURGE-Aufruf unter Verwendung desselben alternativen PCB aus, der beim NIIDQUMS-Aufruf verwendet wurde, und sendet Multisegment-Nachrichten, die mit dem Modul NIIDQUMS erstellt wurden.

Rückgabecodes: Entweder enthalten die Bytes zwei und drei des 4-Byte-Rückgabecodes den Statuscode, oder der Rückgabecode hat den Wert 0.

Beispielprogramm: NIPSQLMS

NIIDQUMS - Multisegment-Nachrichten erstellen

Dieses Modul erstellt Multisegment-Nachrichten. Es hat grundsätzlich die gleiche Funktionalität wie das Modul CMQUEUE, mit dem Unterschied, dass NIIDQUMS keinen PURGE-Aufruf ausgibt.

Die folgenden Parameter werden bereitgestellt:

Name	Format/Länge	Typ	Kommentar
Destination		Eingabe	Ziel
Message		Eingabe	Nachricht
Message length	(B4)	Eingabe	Nachrichtenlänge
TP PCB number	(B4)	Eingabe	Optional

Die Durchführung des PURGE-Aufrufs mit dem Modul NIIDPURG liegt in Ihrer Verantwortung.

Der zu verwendende alternative PCB wird mit dem letzten, optionalen, Parameter angegeben. Wird beim Aufruf keine TP PCB-Nummer angegeben, wird der mit dem Subparameter ALTPCB des Makros NTIMSPT (im Natural-Parametermodul) angegebene alternative TP PCB verwendet.

Beispielprogramm: NIPSQLMS

NIIDSETT - Fremdtransaktionscode abfragen

Um einen korrekten Transaktionswechsel zu einem fremden Transaktionscode durchführen zu können, muss der Typ des Fremdtransaktionscodes bekannt sein. Um diesen Typ zu erhalten, kann das spezielle Modul NIIDSETT verwendet werden. Wird NIIDSETT nicht verwendet, wird angenommen, dass der fremde Transaktionscode vom gleichen Typ ist wie der aufrufende Natural-Transaktionscode. Ist dies nicht der Fall, kommt es zu unvorhersehbaren Ergebnissen oder zu einer abnormalen Beendigung der Sitzung.

Der folgende Parameter wird bereitgestellt:

Name	Format/Länge	Typ	Kommentar
`Transaction type`	(A4)	Eingabe	Mögliche Werte: `CONV` für konversationell, `NONC` für nicht-konversationell.

27 Natural unter IMS TM - User Exits

- NIIXACCT 188
- NIIXSTAR 188
- NIIXSSTA 188
- NIIXISRM 189
- NIIXISRT 189
- NIIXTGU0 189
- NIIXJESA 189
- NIIXPRT0 189
- NIIXRFNU 190
- NIIXTGN0 190

Dieses Kapitel enthält eine Übersicht über die User Exits, die bei der Natural IMS TM-Schnittstelle zur Verfügung stehen. Für jeden Exit wird ein gleichnamiges Quellcodemodul bereitgestellt. Jedes Quellcodemodul enthält eine Beschreibung der Parameterliste und der Registerkonventionen.

NIIXACCT

Der Exit wird aufgerufen, bevor ein Abrechnungssatz (Accounting Record) in das IMS-TM Log (Protokoll) oder in das SMF geschrieben wird. Er ermöglicht es also, den Inhalt eines Abrechnungssatzes zu ändern. Wenn `NIIXACCT` ein Register 15 ungleich Null zurückgibt, wird kein Abrechnungssatz geschrieben.

NIIXSTAR

Der Exit wird bei jedem Transaktionsschritt aufgerufen, nachdem die SPA und die Nachricht abgerufen wurden und der Natural Thread verschoben (Roll-in) und dekomprimiert wurde. Innerhalb dieses Exits sind der Natural IOCB und der Treiberarbeitsbereich zugänglich.

Wenn nach der Rückkehr von `NIIXSSTA` im Register 15 der Wert 12 steht, wird die Natural IMS TM-Schnittstelle gezwungen, die Natural-Sitzung zu beenden. Jeder andere Wert ungleich Null in Register 15 zwingt die Schnittstelle zur Ausgabe des Natural IMS TM Interface-Fehlers 3517 mit dem Reason Code, der den Wert in Register 15 enthält.

Note: Dieser Exit wird nicht aufgerufen, wenn eine neue Natural-Sitzung gestartet wird.

NIIXSSTA

Dieser Exit wird aufgerufen, wenn eine neue Natural-Benutzersitzung gestartet wurde und die SPA und der Natural IOCB initialisiert wurden. Innerhalb dieses Exits sind der Natural IOCB und der Treiberarbeitsbereich zugänglich.

Wenn nach der Rückkehr von `NIIXSSTA` im Register 15 der Wert 12 steht, wird die Natural IMS TM-Schnittstelle gezwungen, die Natural-Sitzung zu beenden. Jeder andere Wert ungleich Null in Register 15 zwingt die Schnittstelle zur Ausgabe des Natural IMS TM Interface-Fehlers 3509 mit dem Reason Code, der den Wert in Register 15 enthält.

NIIXISRM

Dieser Exit wird vor dem Einfügen der Nachricht in den IOPCB aufgerufen.

NIIXISRT

Dieser Exit wird vor dem Einfügen der SPA in den IOPCB aufgerufen, und zwar selbst bei Ende der Natural-Sitzung. Das Ende der Sitzung kann am leeren Transaktionscode erkannt werden.

NIIXTGU0

Dieser Exit wird aufgerufen, wenn das Servicemodul `CMGETMSG` verwendet wird. `NIIXTGU0` erhält die Kontrolle unmittelbar nach dem GU-Aufruf gegen den IOPCB, und zwar unabhängig vom Statuscode.

NIIXJESA

Dieser Exit wird aufgerufen, wenn die JES API zum Schreiben von Reports verwendet wird. Er wird aufgerufen, nachdem der Optionsstring erstellt wurde und kann zur Änderung des Optionsstrings verwendet werden.

NIIXPRT0

Dieser Exit wird aufgerufen, wenn Reports direkt auf IMS TM-Drucker geschrieben werden. Er kann verwendet werden, um die Codes für Formularvorschub (Form Feed) und Zeilenumbruch (New Line) zu setzen.

NIIXRFNU

Dieser Exit wird aufgerufen, wenn die neue Natural Session einem Roll File zugewiesen wird. Er kann verwendet werden, um die Nummer des Roll File zu berechnen, das für diese Sitzung verwendet werden soll.

NIIXTGN0

Dieser Exit wird aufgerufen, wenn das Servicemodul `CMGSEGO` oder `CMGETSEG` verwendet wird. `NIIXTGNO` erhält die Kontrolle unmittelbar nach dem Abrufen des Nachrichtensegments, und zwar unabhängig vom Statuscode.

28 Natural unter IMS TM - Spezielle Funktionen

- Voraussetzungen 192
- Accounting (Abrechnung) 192
- Monitoring (Überwachung) 194
- Broadcasting (Rundsenden) 194
- Server-Umgebung 195

Dieses Kapitel beschreibt die Verwendung von speziellen Funktionen, die mit der Natural IMS TM-Schnittstelle zur Verfügung stehen.

Voraussetzungen

Für einige dieser Funktionen ist der Natural Authorized Services Manager (ASM) erforderlich.

- Wenn der ASM erforderlich ist, muss er gestartet sein, bevor die Funktion verwendet wird.
- Das vom ASM verwendete Natural-Subsystem muss das gleiche sein wie das von der Natural-Sitzung verwendete.
- Für Accounting (Abrechnung) und Monitoring (Überwachung) muss zusätzlich der SIP-Server aktiviert sein.

Accounting (Abrechnung)

Die Abrechnungsfunktion ist nur in dialogorientierten Umgebungen verfügbar. Sie wird aktiviert, indem im Makro `NTIMSPE` des Natural-Parametermoduls der Schlüsselwort-Subparameter `ACTACTV` der Umgebungstabelle auf `ON` gesetzt wird.

Bei jeder Terminal-E/A werden Informationen über die jeweilige Natural-Sitzung in das IMS-TM Log (Protokoll) oder in das SMF geschrieben, je nach Einstellung des Schlüsselwort-Subparameters `ACTLOG` im `NTIMSPE`-Makro.

- Wenn der Schlüsselwort-Subparameter `ACTACTV` auf `CMD` gesetzt ist, wird ein `/LOG`-Kommando ausgegeben, das den Abrechnungssatz in das IMS TM Log (Protokoll) schreibt. Alle Transaktionscodes müssen daher das Kommando `/LOG` verwenden dürfen. Am Anfang eines jeden Datensatzes wird ein 8-Byte-Header eingefügt. Dieser Header erleichtert die Auswahl der Abrechnungssätze mit Hilfe der IMS TM Utility `DFSERA10`. Der Header-String wird durch den Schlüsselwort-Subparameter `ACTAHDR` der Umgebungstabelle definiert (Makro `NTIMSPE`).
- Wenn der Schlüsselwort-Subparameter `ACTACTV` auf `LOG` gesetzt ist, wird der Abrechnungssatz mit dem `LOG`-Aufruf in das IMS TM Log (Protokoll) geschrieben. Mit dem Schlüsselwort-Subparameter `ACTARID` (Makro `NTIMSPE`) können Sie den zu verwendenden Protokollcode angeben.
- Wenn der Schlüsselwort-Subparameter `ACTACTV` auf `SMF` gesetzt ist, wird der Abrechnungssatz mit Hilfe des Authorized Services Manager in das SMF geschrieben. Mit dem Schlüsselwort-Subparameter `ACTARID` können Sie den zu verwendenden SMF-Satztyp angeben.

Bei jeder Terminal-Ein-/Ausgabe werden die folgenden Informationen über jede Natural-Benutzersitzung gespeichert:

- IMS TM-Kennung des IMS TM-Systems, in dem der Benutzer aktiv ist,

- LTERM-Name des IMS TM-Terminals, an dem die Sitzung gestartet wurde,
- Benutzerkennung des Benutzers der Natural-Sitzung (aus dem IOPCB),
- Anzahl der derzeit durchgeführten Dialogschritte,
- derzeit aktiver Transaktionscode,
- derzeit aktiver PSB-Name,
- aktueller Name der Natural Library, bei der der Benutzer angemeldet ist,
- derzeit aktiver Natural-Programmname,
- Nicht-Natural-Transaktionscode, bei dem die Sitzung möglicherweise unterbrochen ist,
- Uhrzeit und Datum des Starts der Sitzung,
- Uhrzeit und Datum der letzten ENTER-Operation,
- Datenbankkennung (DBID) und Dateinummer (FNR) der Natural-Systemdatei (FNAT) für diese Sitzung,
- DBID und FNR der Natural-Benutzerdatei (FUSER) für diese Sitzung,
- DBID und FNR der Natural-Dictionary-Datei (FDIC) für diese Sitzung,
- DBID und FNR der Natural Security-Systemdatei (FSEC) für diese Sitzung,
- DBID und FNR der Natural-Spooldatei (FSPOOL) für diese Sitzung,
- DBID und FNR der Super Natural-Systemdatei für diese Sitzung,
- Nummer des zuletzt aufgetretenen Natural-Fehlers,
- komprimierte Thread-Länge der letzten Terminalausgabe.

Die Informationen werden durch den DSECT `NIMACTR` gemapped. Es gibt zwei Bereiche für die Speicherung der DBID und FNR der verwendeten Natural-System-Dateien. Im ersten Bereich wird aus Kompatibilitätsgründen für jede DBID und FNR ein Byte verwendet. Im zweiten Bereich wird für jede DBID und FNR ein Vollwort verwendet, um Adabas Version 6 oder höher zu unterstützen. Dem Abrechnungssatz wird ein Längen- und Versionsfeld vorangestellt.

Bevor der Abrechnungssatz in das IMS-TM Log (Protokoll) bzw. in das SMF geschrieben wird, wird der User Exit `NIIXACCT` aufgerufen. Mit diesem User Exit können Sie den Abrechnungssatz an Ihre Erfordernisse anpassen. Sie können auch Informationen an den Abrechnungssatz anhängen. In diesem Fall müssen Sie das Length-Feld auf die neue Länge setzen.

Da der Abrechnungssatz im Command Buffer (Kommando-Pufferspeicher) angelegt wird, darf die Gesamtlänge den mit dem Schlüsselwort-Subparameter `CMBSIZE` (Makro `NTIMSPE` des Natural-Parametermoduls) angegebenen Wert minus 17 Byte nicht überschreiten. Die maximal zulässige Länge wird als Parameter übergeben.

Wenn `NIIXACCT` mit einem Wert ungleich Null in Register 15 zurückkehrt, wird kein Abrechnungssatz geschrieben.

Monitoring (Überwachung)

Die Überwachungsfunktion ist nur in dialogorientierten Umgebungen verfügbar. Sie wird aktiviert, indem der Schlüsselwort-Subparameter `MONACTV` in der Umgebungstabelle auf `ON` gesetzt wird (Makro `NTIMSPE` im Natural-Parametermodul), und verwendet die SIP-Funktion des Authorized Services Manager. Das Natural-Subsystem muss dasselbe sein, das auch von der zu überwachenden Natural-Sitzung verwendet wird.

Sie können die laufenden Aktivitäten aller Natural-Sitzungen, die dasselbe Natural-Subsystem verwenden, mit Hilfe der Funktion **Monitoring** (`M`) des Dienstprogramms SYSTP verfolgen. Weitere Informationen zu diesem Dienstprogramm finden Sie unter *SYSTP* in der *Debugger und Dienstprogramme (Utilities)*-Dokumentation. Für SYSTP-Sitzung muss ebenfalls dasselbe Natural-Subsystem verwendet werden.

Broadcasting (Rundsenden)

Die Broadcasting-Funktion ist nur in dialogorientierten Umgebungen verfügbar. Sie wird aktiviert, indem der Subparameter `BROACTV` in der Umgebungstabelle auf `ON` gesetzt wird (Makro `NTIMSPE` im Natural-Parametermodul), und verwendet die SIP-Funktion des Authorized Services Manager.

Ist das Broadcasting aktiviert, ist es möglich, Broadcast-Nachrichten an bestimmte Benutzer eines bestimmten Natural-Subsystems zu senden. Solche Benutzer können sein:

- alle Benutzer des Natural-Subsystems, mit dem der Absender verbunden ist,
- alle Benutzer des Natural-Subsystems innerhalb desselben IMS TM-Systems wie der Absender der Nachricht,
- alle Benutzer des Natural-Subsystems innerhalb desselben IMS TM-Systems wie der Absender der Nachricht, jedoch zusätzlich eingeschränkt auf einen bestimmten Transaktionscode,
- alle Benutzer des Natural-Subsystems innerhalb desselben IMS TM-Systems wie der Absender der Nachricht, jedoch zusätzlich eingeschränkt auf eine Natural-Anwendung,
- alle Benutzer des Natural-Subsystems innerhalb desselben IMS TM-Systems wie der Absender der Nachricht, jedoch zusätzlich beschränkt auf eine Natural-Anwendung und eine bestimmte FUSER-Systemdatei.

Wenn es in einer Sitzung zu einer Terminalausgabe kommt, wird geprüft, ob die Sitzung eine Nachricht empfangen muss oder nicht. Wenn nicht, wird die normale Natural-Ausgabe gesendet. Wenn ja, wird die Nachricht anstelle der normalen Ausgabe gesendet, und beim Drücken von ENTER wird der Natural-Nukleus angewiesen, den letzten Bildschirm erneut zu senden. Auf diese Weise sehen Sie zuerst die Nachricht und erhalten danach den normalen Natural-Ausgabebildschirm.

Wenn mehr als eine Broadcast-Nachricht vorhanden ist, werden die Nachrichten nacheinander angezeigt, bis die letzte Nachricht angezeigt wurde. Danach wird der normale Natural-Ausgabebildschirm angezeigt.

Eine Broadcast-Nachricht wird nur angezeigt, wenn die bei der Erstellung der Nachricht angegebene Verfallszeit nicht überschritten wurde.

Wenn eine Broadcast-Nachricht gesendet wurde, müssen Sie RESET drücken, bevor Sie erneut ENTER drücken können. Alle möglichen Achtungskennungen (Attention-IDs) haben die gleiche Wirkung wie das Drücken von ENTER.

Mit dem Dienstprogramm SYSTP können Sie Broadcast-Nachrichten erstellen und den Inhalt aller aktiven Nachrichten zusammen mit der LTERM/IMSID des Absenders anzeigen. Der Text einer Nachricht ist auf 72 Bytes begrenzt.

Die zu sendenden Nachrichten werden in einem Pool gespeichert, der vom SIP-Server verwaltet wird. Sie verbleiben dort, bis Sie sie mit dem Dienstprogramm SYSTP löschen oder bis Sie den Authorized Services Manager herunterfahren.

Caution: Wenn eine Rundspruchnachricht gelöscht oder erstellt wird, werden auch alle abgelaufenen Nachrichten gelöscht.

Server-Umgebung

Die Server-Umgebung ermöglicht es 3GL-Anwendungen, Natural-Programme über eine Aufrufschnittstelle (Call Interface) auszuführen. Sie ist in allen unterstützten IMS TM-Umgebungen verfügbar und besteht aus dem Natural IMS TM-Treiber `NIISRVD`, der Server-Aufrufschnittstelle `NIIBOOTS` und der Service-API `NIIPCOM`.

`NIISRVD` und `NIIBOOTS` werden als Quellcodemodule geliefert und müssen bei Ihnen vor Ort assembliert und verlinkt werden. Einzelheiten finden Sie unter *Natural IMS TM-Schnittstelle auf z/OS installieren* in der Natural *Installation für z/OS*-Dokumentation.

In der Serverumgebung können Sie eine Natural-Sitzung starten, indem Sie `NIIBOOTS` aus einem beliebigen 3GL-Programm aufrufen. Nachdem die Natural-Sitzung gestartet wurde, kehrt sie zu dem aufrufenden 3GL-Programm zurück und wartet auf weitere Eingaben. Die Eingabe wird normalerweise von `CMSYNIN` kommend erwartet, was bedeutet, dass das 3GL-Programm den primären Eingabe-Dataset von Natural simulieren muss.

Es wird dringend empfohlen, das Server-Natural immer auf die `NEXT`-Zeile zu legen. Dadurch kann der nächste Aufruf von `NIIBOOTS` entweder ein Natural-Kommando oder ein Natural-Programm ausführen. Andernfalls würde der nächste Aufruf von `NIIBOOTS` als Eingabe für ein Natural-Programm behandelt werden, das durch einen früheren Aufruf von `NIIBOOTS` gestartet wurde.

Ähnlich wie bei der nachrichtenorientierten Schnittstelle werden alle Ausgaben, die normalerweise in `CMPRINT` geschrieben werden, an das mit dem Natural-Profilparameter `SENDER` angegebene IMS TM-Ziel gesendet. Einzelheiten zu den speziellen Zielen, die von der Natural IMS TM-Schnittstelle verwendet werden, finden Sie unter ***Sender-Ziel*** im Abschnitt *Natural unter IMS TM - Umgebungen*.

Caution: In einer MPP-Umgebung wird standardmäßig für alle in dieser Region geplanten Transaktionen dasselbe Server-Natural verwendet. Wenn Sie mehrere Server-Naturals in derselben MPP-Region verwenden möchten, müssen Sie mehrere Server-Call-Schnittstellen generieren. Jede Server-Call-Schnittstelle muss mit einem eindeutigen Namen generiert werden, der mit dem `NIMBOOT`-Parameter `SERVERN` angegeben wird, und muss unter einem eindeutigen Namen verlinkt werden. Es wird empfohlen, das Lademodul nach dem mit `SERVERN` angegebenen Namen zu benennen.

Aufruf-Schnittstelle NIIBOOTS

`NIIBOOTS` ist der Standardname, wie er in der Dokumentation und in den mitgelieferten Beispielprogrammen verwendet wird. Dieser Standardname kann während der Installation geändert werden.

`NIIBOOTS` benötigt die folgenden Parameter:

- die PSB-Adresse (die Adresse der PCB-Adressliste),
- den Kommandobereich,
- den Antwortbereich.

Im Kommandobereich kann Folgendes übergeben werden:

- die Startup-Parameter,
- jedes Natural-Kommando, gefolgt von seinen Eingabedaten,
- die `NIIBOOTS`-spezifischen Kommandos, wie `STAT` und `REFR` (in Kombination mit den Startup-Parametern).

Die Startup-Parameter werden in zwei zusammenhängenden 80-Byte-Bereichen übergeben. Der erste Bereich enthält den Namen der Umgebungstabelle und den Namen des zu verwendenden Transaktionscodes wie folgt:

```
ENV-TAB=environment-table-name
TRNCODE=transaction-code-name
```

Der Transaktionscode wird nur berücksichtigt, wenn im Natural-Parametermodul im Makro `NTIMSP TRNCODE=ON` angegeben ist. Einzelheiten zur Verwendung des Transaktionscodes finden Sie im Makro `NIMBOOT` im Abschnitt ***Natural unter IMS TM - Konfiguration***).

Der zweite Bereich enthält die dynamischen Natural-Parameter, mit denen die Natural-Sitzung gestartet werden soll.

Der Antwortbereich ist der Bereich, in dem eine Antwort aus dem ausgeführten Natural-Programm über die Service-API NIIPCOM eingegeben werden soll.

NIIBOOTS prüft jedes Mal, wenn es aufgerufen wird, ob das Server-Natural initialisiert worden ist.

- Wenn Natural nicht initialisiert wurde, wird eine neue Natural-Sitzung gestartet und das empfangene Kommando wird als dynamischer Parameter an Natural übergeben.
- Wenn Natural initialisiert wurde, wird der im Kommandobereich empfangene String als Natural-Kommando oder als Natural-Programm an Natural übergeben.

Die NIIBOOTS-spezifischen Kommandos STAT und REFR haben die folgende Funktion:

- STAT gibt im Antwortbereich COLD zurück, wenn Natural noch nicht initialisiert wurde, und WARM, wenn es bereits initialisiert wurde.
- REFR erzwingt die Initialisierung/Reinitialisierung von Natural, unabhängig vom aktuellen Zustand von Natural.

Die Benutzung einer ON ERROR-Routine wird empfohlen

Es wird dringend empfohlen, eine ON ERROR-Routine in den ausgeführten Natural-Programmen zu verwenden, um dem aufrufenden 3GL-Programm einige Informationen im Antwortbereich mittels NIIPCOM zurückzugeben.

Rückgabecodes

NIIBOOTS übergibt beim Beenden von Natural den von Natural bereitgestellten Rückgabecode

Beispielprogramme

Um die Verwendung von NIIBOOTS und NIIPCOM zu veranschaulichen, werden die Beispielprogramme NIPBOOTS und NIPPCOM mitgeliefert. NIPBOOTS spielt die Rolle des aufrufenden 3GL-Programms, NIPPCOM ist ein Natural-Beispielprogramm, das in der Serverumgebung ausgeführt wird und den String NIISRVR in den Antwortbereich schreibt. Die ON ERROR-Routine stellt die Natural-Fehlernummer in den Antwortbereich.

Mit den Beispielprogrammen können Sie das folgende Szenario durchspielen:

1. Übergeben Sie das Kommando STAT. Die Zeichenkette COLD wird in den Antwortbereich zurückgegeben.
2. Übergeben Sie das Kommando: STACK=(LOGON SYSEXTP),SENDER=S0201, wobei S0201 der LTERM-Name des zugewiesenen Druckergeräts im Server-Natural ist.

Natural wird initialisiert und ist bereit, ein Natural-Kommando in der Library `SYSEXTP` zu empfangen. Die Nachricht über die erfolgreiche Anmeldung wird auf dem zugewiesenen Drucker ausgegeben. Im Antwortbereich wird nichts zurückgegeben.

3. Übergeben Sie das Kommando `STAT`. Die Zeichenkette `WARM` wird in den Antwortbereich zurückgegeben.
4. Übergeben Sie das Kommando `NIPPCOM`. Das Programm `NIPPCOM` wird ausgeführt und die Zeichenkette `NIPSRVR` wird in den Antwortbereich zurückgegeben. Natural ist bereit, das nächste Kommando in der Library `SYSEXTP` entgegenzunehmen.
5. Übergeben Sie das Kommando: `REFR STACK=(LOGON SYSEXTP;NIPPCOM),SENDER=S0201`

   Natural wird reinitialisiert und das Programm `NIPPCOM` in der Library `SYSEXTP` wird ausgeführt. Der Antwortbereich enthält die Zeichenkette `NIPSRVR`.
6. Übergeben Sie das Kommando `FIN`.

   Natural wird beendet und es werden keine Informationen an den Antwortbereich übergeben. Der Rückgabecode enthält den Rückgabecode der Natural-Beendigung. Die Natural-Beendigungsnachricht wird auf dem zugewiesenen Druckergerät ausgegeben.
7. Übergeben Sie das Kommando `STAT`. Die Zeichenfolge `COLD` wird an den Antwortbereich zurückgegeben.

29 Natural unter IMS TM - Wiederherstellungsbehandlung

- System- und Benutzerabbrüche 200
- Nicht behebbare Fehler 200
- Behebbare Fehler 201

Dieses Kapitel beschreibt die Behandlung von Wiederherstellungsvorgängen in der Natural IMS TM-Schnittstelle.

System- und Benutzerabbrüche

Die Natural IMS TM-Schnittstelle ist durch eine ESTAEX-Umgebung geschützt, die im Falle eines Abbruchs die Kontrolle übernimmt.

- Wird ein Benutzerabbruch erkannt, werden die Ressourcen bereinigt und der Abbruch abgefangen, ohne dass Natural die Kontrolle erhält.
- Wird ein Systemabbruch festgestellt, wird Natural über den Abbruch informiert und je nach Einstellung des Natural-Profilparameters `DU` setzt Natural die Sitzung mit einer Fehlermeldung fort oder beendet sie.
- Wenn die Unterstützung von IBMs Language Environement (LE) aktiviert ist und der Abbruch auftritt, während ein LE-Programm die Kontrolle hat, werden vom Benutzer geschriebene oder LE-spezifische Condition Handler angewendet, und Natural wird nur dann über den Abbruch informiert, wenn die Bedingung von allen LE-Condition Handlers abgefangen wird. In diesem Fall wird der Abbruch von Natural gemäß den folgenden Schritten behandelt, bevor die standardmäßige Abbruchbehandlung stattfindet:
  - Die entsprechende LE-Fehlermeldung wird nach `SYSOUT` geschrieben,
  - ein LE-Snap-Dump wird gemäß der LE-Laufzeitoption `TERMTHDACT` nach `CEEDUMP` geschrieben,
  - die LE wird angewiesen, die Verarbeitung nach dem Natural-`CALL`-Statement wieder aufzunehmen,
  - eine spezielle Natural-Fehlermeldung (NAT0950 bei `DU=OFF` oder NAT9967 bei `DU=ON`) wird ausgegeben, die die LE-Fehlernummer angibt.

  In allen Fällen können Sie einen Dump erzeugen, der die Situation zum Zeitpunkt des Auftretens des Fehlers darstellt (Registerinhalte, PSW usw.). Der Dump wird erzeugt, wenn `DU=ON` oder `DU=SNAP` gesetzt ist oder wenn der Benutzerabbruch dies angefordert hat.

Nicht behebbare Fehler

Ein nicht behebbarer Fehler ist ein logischer Fehler, der von der Natural IMS TM-Schnittstelle erkannt wird und von Natural nicht behandelt werden kann. Diese Situationen treten typischerweise während des Starts, der Beendigung oder der Terminal-E/A auf. In allen Fällen ist die Natural-Laufzeit nicht aktiv und kann daher nicht auf den Fehler reagieren.

Wenn ein nicht behebbarer Fehler festgestellt wird, gibt die Natural IMS TM-Schnittstelle einen NII-Fehler aus und beendet die Sitzung. Die Fehlermeldung wird auch in das IMS TM Log (Pro-

tokoll) und in das Systemprotokoll geschrieben. Je nach der Dump-Option in der Fehlermeldungstabelle wird ein Snap-Dump erstellt.

Wenn Sie nicht möchten, dass eine Nachricht in das IMS TM-Protokoll geschrieben wird, müssen Sie den Schlüsselwort-Subparameter `ERRLHDR` des Makros `NTIMSPE` (im Natural-Parametermodul) explizit auf Null setzen, d. h. Sie müssen `ERRLHDR=,` angeben.

Wenn es nicht möglich ist, die Fehlermeldung zu senden (z.B. wenn der GU fehlgeschlagen ist), wird die Sitzung abgebrochen (User Abend).

Behebbare Fehler

Wenn die Natural IMS TM-Schnittstelle einen logischen Fehler feststellt, der von Natural behandelt werden kann, z. B. ein ungültiges Ziel für einen Report, wird eine Natural-Fehlermeldung ausgegeben und Natural fährt mit seiner Standard-Fehlerbehandlung fort.

V Natural unter TSO

30 Natural unter TSO

- Allgemeine Informationen über die Natural-TSO-Schnittstelle 206
- Treiberparameter für die Natural TSO-Schnittstelle 206
- Von Natural unter TSO verwendete Datasets 206
- TSO-Kommandos aus Natural unter TSO absetzen 210

Dieses Kapitel beschreibt die Funktionalität der Natural TSO-Schnittstelle (Produktcode NTI) sowie den Betrieb und die einzelnen Komponenten von Natural unter TSO im Betriebssystem z/OS.

Verwandte Dokumente

Natural TSO-Schnittstelle auf z/OS installieren in der Natural *Installation für z/OS*-Dokumentation.

Allgemeine Informationen über die Natural-TSO-Schnittstelle

Die Natural-TSO-Schnittstelle (Load-Module `NATTSO` und `NATTSOL`) besteht aus einer Anzahl von Service-Routinen, die eine Schnittstelle zum Betriebssystem z/OS bilden. Siehe auch *Natural TSO-Schnittstelle auf z/OS installieren* in der Natural *Installation für z/OS*-Dokumentation. Das Modul `NATTSOL` enthält Funktionen für die IBM LE-Umgebung. Sie können `NATTSO` entweder mit dem Natural-Nukleus verlinken oder separat ausführen, indem Sie es über den Profilparameter `NUCNAME` mit einem gemeinsam genutzten Natural-Nukleus verbinden.

`NATTSO` ist vollständig reentrant und kann oberhalb der 16-MB-Grenze laufen. Mehrere Natural-Sitzungen können innerhalb einer TSO-Region parallel gestartet werden. Sie können dann mit einem `SWAPKEY` zwischen den Sitzungen umschalten (siehe den entsprechenden Subparameter des Profilparameters `TSOP` in der *Parameter-Referenz*-Dokumentation).

Treiberparameter für die Natural TSO-Schnittstelle

Informationen zu den Treiberparametern, die für die Natural TSO-Schnittstelle zur Verfügung stehen, finden Sie in der Beschreibung des Profilparameters `TSOP` oder des Parametermakros `NTTSOP` in der *Parameter-Referenz*-Dokumentation.

Von Natural unter TSO verwendete Datasets

Die folgenden Datasets werden benötigt, wenn bestimmte Funktionen während einer Natural TSO-Sitzung verwendet werden:

Dataset	Funktion
CMEDIT	Software AG Editor-Arbeitsdatei
CMHCOPY	Hardcopy-Druckausgabe
CMPLOG	Dynamische Profilparameter-Reportausgabe
CMPRMIN	Dynamische Profilparameter-Eingabe
CMPRT*nn*	Zusätzliche Reports 01-31
CMTRACE	Externe Trace-Ausgabe
NATRJE	Job Submit-Ausgabe
STEPLIB	Load Library für externe Module
CMWKF*nn*	Arbeitsdateien 01-32

Diese Datasets werden im Folgenden beschrieben.

Wenn nicht anders angegeben, sind die Standard-DCB RECFM/LRECL-Informationen wie folgt:

RECFM=FB und LRECL=80 für sequenzielle Eingabe-Datasets

RECFM=FBA und LRECL=133 für sequenzielle Ausgabe-Datasets

CMEDIT - Software AG Editor-Arbeitsdatei

Der VSAM-Dataset für die Software AG Editor-Arbeitsdatei ist erforderlich, wenn ein lokaler oder globaler Software AG Editor Buffer Pool verwendet werden soll. Wenn er nicht in der JCL oder durch das TSO-Kommando ALLOC definiert ist, wird der Name der Editor-Arbeitsdatei, der im Subparameter DSNAME des Profilparameters EDBP oder des Parametermakros NTEDBP angegeben ist, von Natural verwendet, um die dynamische Zuordnung für die Editor-Arbeitsdatei durchzuführen.

Alternativ kann der Profilparameter EDPSIZE verwendet werden, um mit einem Editor-Hilfs-Buffer-Pool zu arbeiten, für den keine Editor-Arbeitsdatei erforderlich ist. Weitere Informationen über die Installation des Software AG Editor finden Sie unter *Software AG Editor installieren* in der Natural *Installation für z/OS*-Dokumentation.

CMHCOPY - Hardcopy-Druckausgabe

Der Standardname des Dataset für die Hardcopy-Druckausgabe ist CMHCOPY. Er kann durch eine der folgenden Möglichkeiten geändert werden:

- Schlüsselwort-Subparameter DEST des Profilparameters PRINT für Druckdatei 0
- Profilparameter HCDEST, der gleichbedeutend ist mit PRINT=((0),DEST=...),
- Wert der Systemvariablen *HARDCOPY während der Sitzung
- Terminalkommando %H während der Sitzung

Mit den Schlüsselwort-Subparametern des Profilparameters PRINT für die Druckdatei 0 können Sie die Standardwerte für den Hardcopy-Datataset ändern. Der Standard-Dataset-Name CMHCOPY impliziert CLOSE=FIN für den Hardcopy-Print-Dataset, d.h. nachdem der Dataset zur Ausgabe geöffnet wurde, wird eine nachträgliche Änderung des Hardcopy-Print-Ausgabe-Dataset-Namens nicht beachtet. Wird zum Zeitpunkt des Öffnens ein anderer Name definiert, wird der Hardcopy-Dataset bei der nächsten Terminal-E/A geschlossen.

Während der Sitzung kann der Hardcopy-Dataset freigegeben und neu zugewiesen werden (vor dem Öffnen oder nach dem Schließen), indem er dynamisch zugeordnet wird (siehe Natural Anwendungsprogrammierschnittstelle USR2021N).

CMPLOG - Dynamische Profilparameter-Reportausgabe

Wenn der Profilparameter PLOG auf ON gesetzt ist und der Dataset CMPLOG verfügbar ist, werden die ausgewerteten dynamischen Profilparameter während der Sitzungsinitialisierung in diesen Dataset geschrieben. Wenn der Dataset CMPLOG nicht verfügbar ist, werden die ausgewerteten dynamischen Profilparameter im Line-Mode auf dem TSO-Terminal geschrieben.

CMPRMIN - Dynamische Profilparameter-Eingabe

Falls vorhanden, wird dieser Dataset während der Sitzungsinitialisierung gelesen, um die dynamischen Profilparameter zu erhalten. Nur die ersten 72 Positionen jedes Datensatzes werden zum Bau eines dynamischen Profilparameter-Strings verwendet.

Alle anderen Profilparameter, die direkt für den Start des Natural-Nukleus übergeben werden, z. B. durch das TSO CALL-Kommando, werden an das Ende des Parameterstrings angehängt, der aus der Eingabe von CMPRMIN gebildet wird, d. h. diese können zum Überschreiben der Parameter aus CMPRMIN verwendet werden.

CMPRTnn - Zusätzliche Reports 01-31

Diese Datasets können von Natural Print File-Statements wie WRITE (*nn*) verwendet werden. Wenn keine DCB-Informationen (z.B. RECFM, LRECL, BLKSIZE) vorhanden sind, werden die Standardwerte durch den Profilparameter PRINT oder durch das Makro NTPRINT im Natural-Parametermodul definiert. Die Namen der Druckdatei-Datasets können durch den Schlüsselwort-Subparameter DEST überschrieben werden.

CMTRACE - Externe Trace-Ausgabe

Wenn der Profilparameter `ETRACE` auf `ON` gesetzt ist oder wenn das entsprechende Terminalkommando `%TRE+` ausgegeben wurde, wird jede Natural-Trace-Ausgabe während der Sitzung in den Dataset `CMTRACE` geschrieben. Um die Natural-Komponenten zu definieren, die einem Trace unterzogen werden sollen, muss der Profilparameter `TRACE` verwendet werden.

Wenn der Dataset `CMTRACE` nicht verfügbar ist, wird er dynamisch zugeordnet als

```
//CMTRACE DD SYSOUT=*
```

wenn der erste Trace-Datensatz geschrieben werden soll:

NATRJE - Job-Submit-Ausgabe

Dieser Dataset wird zur Natural-Job-Übergabe verwendet. Wenn er nicht definiert ist, wird er dynamisch zugeordnet als

```
//NATRJE DD SYSOUT=(A,INTRDR)
```

wenn der erste Job übergeben wird:

STEPLIB - Load Library für externe Module

`STEPLIB` ist der Standard-Load-Library-Name zum Laden externer Module, z.B. des gemeinsam genutzten Nukleus (siehe Profilparameter `NUCNAME`), eines separaten Adabas-Linkroutinenmoduls (siehe Profilparameter `ADANAME`), des Session-Backend-Programms (siehe Profilparameter `PROGRAM`) und aller externen Subprogramme, die nicht mit dem Natural-Parametermodul verlinkt sind.

Der Name der Load-Library kann durch den Profilparameter `LIBNAM` überschrieben werden. Der angegebene Load-Library-Name muss in der TSO-Job-Control oder durch ein `ALLOC`-Statement, z. B. in der CLIST, die die Natural-Sitzung startet, definiert werden.

CMWKFnn - Arbeitsdateien 01-32

Diese Datasets können von Natural-Workfile-Statements wie z.B. `READ WORK FILE` *`nn`* und `WRITE WORK` *`nn`* verwendet werden. Wenn in der JCL oder im VTOC-Eintrag für den Dataset keine DCB-Informationen (`RECFM`, `LRECL`, `BLKSIZE` usw.) vorhanden sind, werden die Standardwerte durch den Profilparameter WORK oder durch das Makro `NTWORK` im Natural-Parametermodul definiert. Die Workfile-Dataset-Namen können durch den Schlüsselwort-Subparameter `DEST` überschrieben werden.

TSO-Kommandos aus Natural unter TSO absetzen

Sie können das Natural-Beispielprogramm TSO in der Library `SYSEXTP` verwenden, um TSO-Kommandos abzusetzen, zum Beispiel:

```
TSO LISTALC STATUS
```

Wenn Sie TSO ohne Parameter eingeben, werden Sie über ein Menü zur Eingabe eines TSO-Kommandos aufgefordert.

Um das Menü zu verlassen, können Sie an der ersten Stelle einen Punkt (.) eingeben oder Sie können PF3 drücken.