

## Natural for Mainframes

TP モニタインターフェイス

バージョン 4.2.5

October 2009

This document applies to Natural バージョン 4.2.5 and to all subsequent releases.

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

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

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

# 目次

1 TP モニタインターフェイス .....	1
2 Natural での TP モニタの使用 .....	3
Natural でサポートされている TP モニタシステム .....	4
TP 環境での Natural の使用 .....	4
3 CICS 環境での Natural - 概要 .....	7
4 Natural CICS インターフェイスの機能 .....	9
Natural CICS インターフェイス .....	10
CICS 環境での Natural ニュークリアス .....	10
CICS 環境でのシステム制御 .....	11
OSCOR/GETVIS - CICS 動的ストレージまたはオペレーティングシステムスト レイジでの Natural コンポーネント .....	11
CICS 環境での Natural ストレージスレッド .....	15
CICS 環境での Natural のロール機能 .....	16
CICS ロール機能 .....	16
CICS 環境での Natural ローカルバッファプール .....	17
CICS 環境での Natural スワッププール .....	17
CICS 一時ストレージにおける Natural CICS インターフェイスシステム制御レ コード .....	18
NCIDIREX - システムディレクトリモジュール名の出口インターフェイス .....	19
NCITIDEX - 端末 ID 出口インターフェイス .....	20
NCIUIDEX - ユーザー ID 出口インターフェイス .....	21
Natural CICS インターフェイスのデバッグ機能 .....	21
Natural CICS インターフェイス CICS TWA の使用 .....	23
5 Natural CICS 生成パラメータ .....	25
NCISCPCB 生成パラメータ .....	27
NCMDIR マクロパラメータ .....	27
NCMTGD マクロパラメータ .....	32
NTSWPRM マクロパラメータ .....	38
NCIPAPM 生成パラメータ .....	38
NCMPRM マクロパラメータ .....	38
6 VSAM RRDS ロールファイルのカスタマイズ .....	59
VSAM RRDS ロールファイル数の増加 .....	60
VSAM RRDS ロールファイル数の減少 .....	61
VSAM RRDS ロールファイルの特性の変更 .....	61
7 CICS MRO 環境での Natural .....	63
YES に設定した NCIPARM パラメータ COMARET .....	64
NO に設定した NCIPARM パラメータ COMARET .....	64
8 Natural に対する CICS ノードエラープログラムに関する考慮事項 .....	67
通常の状況 .....	68
Natural CICS インターフェイス管理外の状況 .....	68
リカバリメカニズム .....	69
特別な考慮事項 .....	69
ダミープログラムの例 .....	70

9 CICS 3270 Bridge のサポート .....	71
CICS 3270 Bridge のデフォルトサポート .....	72
CICS 3270 Bridge の完全なサポート .....	72
NCIXFATU - NCI ソースモジュール .....	72
プロファイルパラメータ DSC=OFF の推奨 .....	72
10 Natural CICS インターフェイスのスレッドセーフに関する考慮事項 .....	73
11 Natural CICS インターフェイスでの CICS チャネルおよびコンテナのサポート .....	75
12 Natural CICS インターフェイスおよび IBM 言語環境 (LE) .....	77
z/OS 用 CICS トランザクションサーバーのバージョン 3.1 以上 .....	78
CICS/TS 3.1 より前のバージョンの CICS トランザクションサーバー .....	78
13 Natural CICS の特殊な機能 .....	81
Natural 以外のプログラムの呼び出し .....	82
CICS 環境の Natural でのダミーの画面 I/O .....	83
NCISTART - Natural CICS ニュークリアス .....	84
14 Natural CICS のサンプルプログラム .....	85
Natural CICS ソースライブラリのサンプルプログラム .....	86
z/VSE で使用するサンプルプログラム .....	88
15 ユーザープログラムからの Natural の起動 .....	89
Natural セッションをアクティブ化するコマンド .....	90
フロントエンドパラメータ .....	91
LINK を介して起動されたフロントエンド .....	92
START を介して起動されたフロントエンド .....	93
XCTL を介して起動されたフロントエンド .....	94
分散プログラムリンク (DPL) を介して起動されたフロントエンド .....	94
フロントエンドプログラムをバックエンドとして起動 .....	94
16 CICS 環境での Natural の非同期処理 .....	95
Natural の非同期処理 .....	96
CICS 環境での非同期 Natural セッション .....	96
テストおよびデバッグ .....	97
17 CICS 環境での Natural セッションのログ .....	99
ロギング機能 .....	100
Natural ログファイルの定義 .....	100
Natural のログレコード .....	101
18 Natural CICS のパフォーマンスに関する考慮事項 .....	105
環境固有の考慮事項 .....	106
ロール機能の選択 .....	106
共有ストレージのスレッドと GETMAIN を実行したスレッドの違い .....	109
CICS のパラメータ設定 .....	112
行圧縮システム .....	113
擬似会話型トランザクションと会話型トランザクションの違い .....	113
Natural と Adabas .....	113
CICS モニタリング製品 .....	113
19 CICS 環境での Natural の出力ファイルおよびワークファイル .....	115
出力ファイルおよびワークファイルの使用方法のカスタマイズ .....	116

CICS 一時ストレージの出力ファイルおよびワークファイル .....	116
CICS 一時データの出力ファイルおよびワークファイル .....	117
20 Com-plete/SMARTS 環境の Natural .....	119
NFMPRM マクロのパラメータ .....	120
異常終了出口の使用 .....	125
ストレージの使用 .....	126
バックエンドプログラムのサポート .....	126
Natural バッチ実行における Com-plete のサポート .....	127
Com-plete/SMARTS 環境での Natural の非同期処理 .....	127
ユーザプログラムからの Natural の起動 .....	128
ストレージスレッドキーの処理 .....	128
セッション初期化中のユーザ出口処理のサポート .....	128
SMARTS サーバ環境の使用 .....	129
Com-plete のリカバリ可能なセッション処理のサポート .....	132
21 IMS/TM 環境の Natural - 概要 .....	133
22 IMS/TM 環境の Natural - 環境 .....	135
IMS/TM インターフェイスの概要 .....	136
IMS/TM 環境 .....	137
ダイアログ指向の環境 .....	138
メッセージ指向の環境 .....	140
バッチメッセージ処理環境 .....	142
Natural WRITE (n) ステートメントのサポート .....	143
SET CONTROL 'N' (端末コマンド %N) .....	146
IMS/TM における Natural メッセージでの TS=ON のサポート .....	146
SENDER の宛先 .....	146
Natural プロファイルパラメータ PROGRAM のサポート .....	147
23 IMS/TM 環境での Natural - コンポーネント .....	149
フロントエンドモジュール .....	150
Natural IMS インターフェイスモジュール NIINTFM .....	151
物理的な入力編集ルーチン .....	152
ユーザメッセージテーブル DFSCMTU0 .....	152
ロールファイルおよびロールサーバ .....	153
認可サービスマネージャ .....	154
Natural 共有ニュークリアス .....	155
Natural バッファプール .....	155
Adabas インターフェイス .....	155
プリロードリスト .....	156
24 IMS/TM 環境での Natural - コンフィグレーションマクロ .....	157
NIMDRIV マクロパラメータ .....	158
NIMPARM マクロパラメータ .....	160
A .....	160
B~C .....	161
E~H .....	162
L~N .....	163
P .....	164

R～S .....	166
T～U .....	167
NIMTRNTG マクロパラメータ .....	168
NIMLPCB マクロパラメータ .....	171
NIMMSGT マクロパラメータ .....	171
NIMPIXT マクロパラメータ .....	172
NIMBOOT マクロパラメータ .....	173
25 IMS/TM 環境の Natural - サービスプログラム .....	175
Natural IMS/TM インターフェイスサービスプログラムについて .....	176
Natural IMS/TM インターフェイスサービスプログラムの説明 .....	177
NIIBRCST - 渡されたメッセージを端末に送信 .....	177
NIICMD - IMS コマンドを IMS に送信 .....	177
NIIDEFT - Natural トランザクションコードへの遅延切り替えを用意 .....	178
NIIDEFTX - 非 Natural のトランザクションコードへの遅延切り替えを用意 .....	178
NIIDIRT - Natural トランザクションコードへの直接切り替えを用意 .....	179
NIIDIRTX - トランザクションコードへの直接切り替えを用意 .....	180
NIEMOD - モジュール出力ディスクリプタの設定変更 .....	181
NIIGCMD - 以前の IMS/TM コマンドの次の応答セグメントを取得 .....	181
NIIGMSG - 次メッセージの最初のセグメントを取得 .....	182
NIIGSEG - 入力メッセージの次のセグメントを取得 .....	182
NIIGSPA - SPA からデータを取得 .....	183
NIIMSIN - IMS 環境情報を取得 .....	183
NIISRTF - 複数セグメントのメッセージを作成 .....	184
NIISRTM - メッセージキューにメッセージセグメントを挿入 .....	185
NIIPCBAD - PSB 名および PCB アドレスの返却 .....	185
NIIPCOM - 応答エリアにデータを移動 .....	186
NIIPMSG - メッセージを送信 .....	186
NIIPSBAD - PSB アドレスの返却 .....	187
NIIPSPA - SPA でのデータ置換 .....	187
NIIPURG - PURG 呼び出しの発行 .....	188
NIIRETRM - メッセージエリアにデータを移動 .....	188
NIISASD - SENDER と OUTDEST の設定を変更 .....	189
NIU3962 - セッションの終了 .....	189
26 IMS/TM 環境の Natural - サービスモジュール .....	191
サービスモジュールの目的 .....	192
サービスモジュールの説明 .....	192
CMCMMND - IMS オペレータコマンドの発行 .....	192
CMDEFSW - Natural トランザクションコードへの遅延トランザクション切り 替え .....	193
CMDEFSWX - Natural 以外のトランザクションコードへの遅延トランザクシ ョ ン切り替え .....	193
CMDIRNMX - メッセージのない別の会話型トランザクションへの切り替え .....	194
CMDIRNMZ - メッセージのない別の会話型トランザクションへの切り替え .....	194
CMDIRSWX - メッセージのある別の会話型トランザクションへの切り替え .....	195
CMDIRSWZ - メッセージのある別の会話型トランザクションへの切り替え .....	196

CMDISPCB - PCB の内容の取得 .....	197
CMEMOD - MOD 名のダイナミックな変更 .....	197
CMGETMSG - 次のメッセージの読み取り .....	198
CMGETSEG - 次のセグメントの読み取り .....	198
CMGETSPA - SPA からのデータ転送 .....	199
CMIMSID - MVS サブシステム ID の取得 .....	199
CMIMSINF - システム環境情報 .....	199
CMPCBADR - PCB アドレスを返す .....	200
CMPRNTR - デフォルトのハードコピー出力先の変更 .....	201
CMPUTMSG - IO-PCB への出力メッセージの挿入 .....	201
CMPUTSPA - SPA へのデータの移動 .....	202
CMQTRAN - 現在のトランザクションコードテーブルエントリの内容 .....	202
CMQUEUE - 代替 PCB へのメッセージの挿入 .....	203
CMQUEUEX - メッセージの内容に対する完全な制御 .....	203
CMSNFPRT - 論理デバイス名の設定 .....	204
CMSVC13D - Natural セッションの終了 .....	205
CMTRNSET - 代替 PCB による SPA の挿入 .....	205
NIIDDEFS - 外部トランザクションへの遅延切り替え .....	205
NIIDPURG - 複数セグメントメッセージの送信 .....	206
NIIDQUMS - 複数セグメントメッセージの作成 .....	206
NIIDSETT - 外部トランザクションコードの取得 .....	207
27 IMS/TM 環境の Natural - ユーザー出口 .....	209
NIIXACCT .....	210
NIIXSTAR .....	210
NIIXMSSP .....	210
NIIXSSTA .....	210
NIIXISRM .....	211
NIIXISRT .....	211
NIIXTGU0 .....	211
NIIXJESA .....	211
NIIXPRT0 .....	211
NIIXRFNU .....	212
NIIXTGN0 .....	212
28 IMS/TM 環境の Natural - 特別な機能 .....	213
必要条件 .....	214
アカウンティング .....	214
モニタリング .....	216
ブロードキャスト .....	216
マルチセッション機能 .....	217
サーバー環境 .....	219
29 IMS/TM 環境の Natural - リカバリの制御 .....	223
システムアベンドとユーザーアベンド .....	224
リバカリ不能エラー .....	224
リカバリ可能エラー .....	225
30 TIAM 環境の Natural .....	227

Natural TIAM インターフェイスの構造 .....	228
マクロ NAMTIAM のパラメータ .....	229
TIAM 環境の共通メモリプール .....	241
Natural 共有ニュークリアス .....	241
31 TSO 環境の Natural .....	243
Natural TSO インターフェイスに関する全般的な情報 .....	244
Natural TSO インターフェイスの生成パラメータ .....	244
TSO 環境の Natural で使用されるデータセット .....	248
TSO 環境の Natural からの TSO コマンドの発行 .....	251
32 openUTM 環境の Natural - 概要 .....	253
33 UTM 環境の Natural - パート 1 .....	257
Natural UTM インターフェイスの構造 .....	258
メッセージのフォーマット - FREXIT .....	259
UTM アプリケーションでの Natural の組み込み .....	261
共通メモリプール .....	261
その他のストレージエリア .....	264
KDCROOT の生成 .....	266
UTM リソースの定義 - KDCDEF .....	267
UTM DC トランザクション出口ルーチン NUERROR .....	269
UTM 起動機能 .....	269
UTM シャットダウン機能 .....	270
34 UTM 環境の Natural - パート 2 .....	275
NATUTM マクロパラメータ .....	276
NATUTM マクロエントリ .....	306
NURENT マクロパラメータ .....	308
35 UTM 環境の Natural - パート 3 .....	313
ユーザー出口 .....	314
UTM 環境での非同期トランザクション処理 .....	318
UTM 環境での印刷 .....	326
Natural 以外のプログラムの呼び出し .....	328
チェーンで結ばれた部分的な UTM プログラムの呼び出し .....	329
Natural UTM アプリケーションでの Natural 以外のプログラムからの Adabas の呼び出し .....	330
UTM タスクの異常終了 .....	330
36 UTM 環境の Natural - パート 4 .....	331
Natural UTM アプリケーションのアカウントティング .....	332
UTM 環境の Natural で使用するユーティリティプログラム .....	333
Software Exchange .....	338
UTM TACCLASS の概念 - プライオリティ制御 .....	340
Natural UTM アプリケーションの生成 .....	351
Natural UTM アプリケーションの最適化 .....	354
複数のアプリケーションと 1 つの共通 Natural .....	356
ダイナミック Natural パラメータの入力と定義 .....	357
UTM ユーザーの再スタート .....	357
Adabas プライオリティ制御 .....	358



索引 ..... 359








---

# 1 TP モニタインターフェイス

---

Naturalでは、Naturalニュークリアスがオンラインランザクション処理のためにTPモニタ、バッチ処理のためにオペレーティングシステム（OS）にアクセスすることを可能にするインターフェイスを提供します。『Naturalシステムアーキテクチャ』ドキュメントの「TP/OSインターフェイス」も参照してください。

このドキュメントでは、サポートされているTPモニタシステムとともにNaturalを操作するための詳細情報を説明します。

 <b>Natural</b> での TP モニタの使用	TPモニタとともにNaturalを使用するための一般的な情報を記載しています。
 <b>CICS</b> 環境での <b>Natural</b>	Natural CICS インターフェイスの機能、および CICS 環境での Natural の操作および個別のコンポーネントについて説明します。
 <b>Com-plete/SMARTS</b> 環境の <b>Natural</b>	Com-plete/SMARTS 環境で Natural を操作する方法について説明します。
 <b>IMS/TM</b> 環境での <b>Natural</b>	IMS/TM 環境で Natural を実行する方法について説明します。
 <b>TIAM</b> 環境での <b>Natural</b>	TIAM 環境で Natural を実行する方法について説明します。
 <b>TSO</b> 環境での <b>Natural</b>	Natural TSO インターフェイスおよびデータセットに関する一般的な情報を記載しています。
 <b>openUTM</b> 環境での <b>Natural</b>	openUTM 環境で Natural を実行する方法について説明します。

CMS 環境での Natural については、Natural の『オペレーション』ドキュメントの「VM/CMS 環境の Natural」を参照してください。

Natural を TP モニタとともに使用するとき適用される Natural プロファイルパラメータの概要については、Natural の『オペレーション』ドキュメントの「プロファイルパラメータの概要」で「TP モニタ」を参照してください。

『TP モニタインターフェイス』ドキュメントの補足となるドキュメントは、次のNaturalドキュメントです。

- インストール
- オペレーション
- メッセージおよびコード

## 2 Natural での TP モニタの使用

---

- Natural でサポートされている TP モニタシステム ..... 4
- TP 環境での Natural の使用 ..... 4

## Natural でサポートされている TP モニタシステム

---

現在 Natural では、次の TP モニタシステムをサポートしています。

- CICS
- CMS
- Com-plete
- IMS/TM
- TIAM
- TSO
- UTM

Natural で特定の TP モニタを使用する際の詳細については、このドキュメントの該当するセクションを参照してください。

参考情報：SYSTP ユーティリティ

Natural ユーティリティ SYSTP では、さまざまな TP モニタ固有の関数を提供します。このユーティリティは、TP モニタ CICS、Com-plete、IMS/TM、TIAM、TSO、および UTM で利用することができます。

## TP 環境での Natural の使用

---

このセクションでは、次のトピックについて説明します。

- TP 環境への Natural の組み込み
- TP モニタでの Natural トランザクションの呼び出し
- Natural セッションの終了
- プログラム例

### TP 環境への Natural の組み込み

TP モニタ環境では、Natural は標準の TP プログラムとして動作し、この TP モニタの管理下で実行されるプログラムに適用されるルールに従います。

Natural コードは完全にリエントラントであるため、すべての Natural ユーザー間で共有され、ワークエリアのみがユーザーごとに存在します（このユーザーの Natural セッションの間のみ）。

Natural ユーザープログラム（トランザクション）は、ネイティブの TP プログラムとともに実行し、Natural プログラムと従来のプログラムの両方を含む総合システムを形成することができます。

## TP モニタでの Natural トランザクションの呼び出し

Natural トランザクションを呼び出すには、Natural という TP プログラムを起動して、システムコマンド LOGON およびスタックで実行される Natural トランザクションの名前を指定します。

複数のコマンド／トランザクションおよびコマンド／トランザクションに対する入力データを、Natural を呼び出すときにスタックを使用して渡すことができます。

## Natural セッションの終了

Natural セッションは TERMINATE ステートメントまたは FIN システムコマンドを実行することによって終了できます。

## プログラム例

Natural ライブラリ SYSEXTP には、特定の TP モニタ環境でのみ適用される特定の関数用の複数のプログラム例が含まれています。

---



# 3 CICS 環境での Natural - 概要

---

このドキュメントでは、Natural CICS インターフェイス（製品コード NCI）の機能、および CICS 環境での Natural の操作および個別のコンポーネントについて説明します。

- **Natural CICS** インターフェイスの機能
- **Natural CICS** 生成パラメータ
- **VSAM RRDS** ロールファイルのカスタマイズ
- **CICS MRO** 環境での **Natural**
- **Natural** に対する **CICS** ノードエラープログラムに関する考慮事項
- **CICS 3270 Bridge** のサポート
- **Natural CICS** インターフェイスのスレッドセーフに関する考慮事項
- **Natural CICS** インターフェイスコンテナのサポート
- **Natural CICS** インターフェイスおよび **IBM** 言語環境（**LE**）
- **Natural CICS** の特殊な機能
- **Natural CICS** のサンプルプログラム
- ユーザープログラムからの **Natural** の起動
- **CICS** 環境での **Natural** の非同期処理
- **CICS** 環境での **Natural** セッションのログ
- **Natural CICS** のパフォーマンスに関する考慮事項
- **CICS** 環境での **Natural** の出力ファイルおよびワークファイル

## CICS テーブルの参照

適切な場合は、CICS テーブル (DCT、FCT、PCT、PPT、TCT など) への参照を、以下に示す対応する項目への参照と見なすことができます。

- アセンブリタイプのリソース定義
- CEDA を介したオンラインリソース定義
- DFHCSDUP を介したバッチリソース定義

以下の項目も参照してください。

- Natural の『インストール』ドキュメントの「*Natural CICS* インターフェイスのインストール」
- Natural の『メッセージおよびコード』ドキュメントの「*CICS* 環境下の *Natural* のアベンドコードとエラーメッセージ」、「*CICS* 環境下の *Natural* の情報メッセージ」、および「*NCISCPRI* 警告およびエラーメッセージ」
- Natural の『メッセージおよびコード』ドキュメントの「*CICS* および *openUTM* 環境下で有効な *Natural* スワッププールマネージャのエラーメッセージ」。
- SYSTP - この *Natural* ユーティリティはさまざまな TP モニタ固有の関数を提供します。
- *CICS* 環境でのサーバーとしての *Natural*

# 4 Natural CICS インターフェイスの機能

---

■ Natural CICS インターフェイス .....	10
■ CICS 環境での Natural ニュークリアス .....	10
■ CICS 環境でのシステム制御 .....	11
■ OSCOR/GETVIS - CICS 動的ストレージまたはオペレーティングシステムストレージでの Natural コンポーネント .....	11
■ CICS 環境での Natural ストレージスレッド .....	15
■ CICS 環境での Natural のロール機能 .....	16
■ CICS ロール機能 .....	16
■ CICS 環境での Natural ローカルバッファプール .....	17
■ CICS 環境での Natural スワッププール .....	17
■ CICS 一時ストレージにおける Natural CICS インターフェイスシステム制御レコード .....	18
■ NCIDIREX - システムディレクトリモジュール名の出口インターフェイス .....	19
■ NCITIDEX - 端末 ID 出口インターフェイス .....	20
■ NCIUIDEX - ユーザー ID 出口インターフェイス .....	21
■ Natural CICS インターフェイスのデバッグ機能 .....	21
■ Natural CICS インターフェイス CICS TWA の使用 .....	23

この章では、Natural CICS インターフェイスの機能について説明します。

## Natural CICS インターフェイス

---

Natural CICS インターフェイスはコマンドレベルのアセンブラで実装されるため、Natural は、CICS マルチプルリージョンオプションおよびデバッグ機能 CEDF と互換性があります。

Natural CICS インターフェイスは、セッションの初期化、ロールイン再スタート（擬似会話型モード）、端末の I/O、データベースアクセス、ABEND 処理、Natural ローカルバッファプール呼び出しと、外部サブルーチンのロード、リンク、および解放を制御します。さらに、すべてのロール I/O 操作は Natural CICS インターフェイスから行われます。

## CICS 環境での Natural ニュークリアス

---

Natural ニュークリアスは、Natural のリエントラントモジュールとさまざまなサポートルーチンの組み合わせです。これらのルーチンは、サイト依存のアセンブリを必要とするソースプログラムおよびロードモジュールとして提供されます。

Natural ニュークリアスの CICS 関連のコンポーネントを以下に示します。

### ■ Natural CICS インターフェイスモジュール NCISTART

これはエントリルーチンです。特に Natural CICS インターフェイス言語環境 (LE) リンクの準備を行います。「[Natural CICS インターフェイスおよび IBM 言語環境 \(LE\)](#)」を参照してください。このモジュールは CICS のバージョンに依存しています。

### ■ Natural CICS インターフェイスモジュール NCIROOT

このモジュールは、すべての CICS 関連のロジックを CICS サービスおよび CICS コントロールブロックアクセスとして保持します。このモジュールは CICS のバージョンに依存しています。

### ■ Natural CICS パラメータモジュール NCIPARM

このモジュールは、Natural CICS インターフェイスランタイムおよびシステム環境生成オプションを保持します。このモジュールは CICS のバージョンに依存していません。ただし、パラメータの一部は CICS のバージョンに基づいて設定する必要があります。

### ■ Natural CICS インターフェイスオブジェクトのみの NCINUC

このモジュールは、Natural ニュークリアスおよび Natural CICS インターフェイスシステム制御ロジックによって呼び出されるサービスルーチンを保持します。これらのルーチンは CICS および CICS のバージョンに依存しておらず、NCIROOT および NCISTART で CICS サービスルーチンを呼び出すことにより CICS を処理します。

## ■ Natural CICS インターフェイスモジュール NCIXCALL

このモジュールは CICS の個別プログラムです。つまり、Natural によって呼び出される 3GL プログラムから EXEC CICS LINK を介して起動されるため、Natural ニュークリアスにリンクされていません。『オペレーション』ドキュメントの「Natural 3GL CALLNAT インターフェイス」を参照してください。このモジュールは CICS のバージョンに依存しています。

## CICS 環境でのシステム制御

CICS 固有の Natural 機能には、スレッドでの動的ストレージの構成、およびこれらのスレッドを処理する追加の機能が含まれているため、Natural CICS システム制御プログラムはより効率的に動的ストレージを処理することができます。

Natural CICS システム制御プログラムは、当初 CICS 環境での GETMAIN の 64 KB 制限を克服するために開発されました。これにより、擬似会話型ユーザーに対するロールファイル I/O 操作および再配置機能を含む完全なストレージの割り当ておよび管理機能が提供されます。

CICS での Natural の擬似会話型処理機能を拡張するために、システム制御プログラムでは、スレッド（ユーザーごとに設定される連続したストレージ）を使用します。この構造により、Natural は CICS の関与を最小限に抑えながら動的ストレージを管理することができます。

システム制御について完全に理解するには、その構造および操作に関する以下の説明を参照してください。CICS 環境での Natural のインストール手順を開始する前に、このメカニズムをよく理解してください。

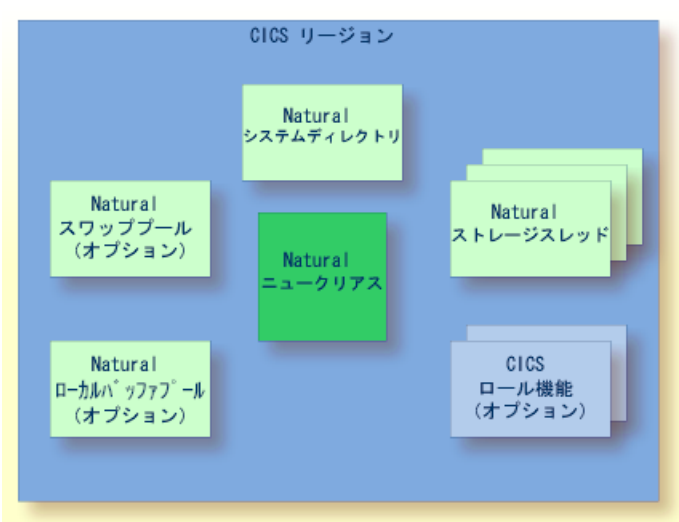
## OSCOR/GETVIS - CICS 動的ストレージまたはオペレーティングシステムストレージでの Natural コンポーネント

シナリオ 1：

単一の CICS リージョン

以下の図は、CICS 動的ストレージに存在する Natural システムのコンポーネントを示しています。コンポーネントについては、次の項目で説明しています。

- CICS 環境での Natural ストレージスレッド
- CICS 環境での Natural ローカルバッファプール
- CICS 環境での Natural スワッププール
- Natural ロール機能

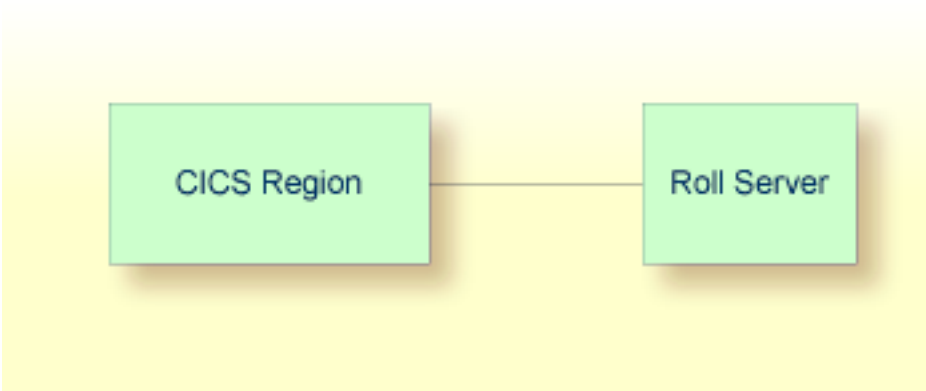


シナリオ 1 は、z/OS または z/VSE 環境において単一の CICS アプリケーションリージョンで Natural をローカルに実行する場合に適用されます。

**注意:** z/OS システムに関する注意事項：この他のシナリオも実現可能です。次の 3 つの図は、z/OS システム、CICS リージョン、Natural ロールサーバー、および Natural 認可サービスマネージャの組み合わせを示しています。

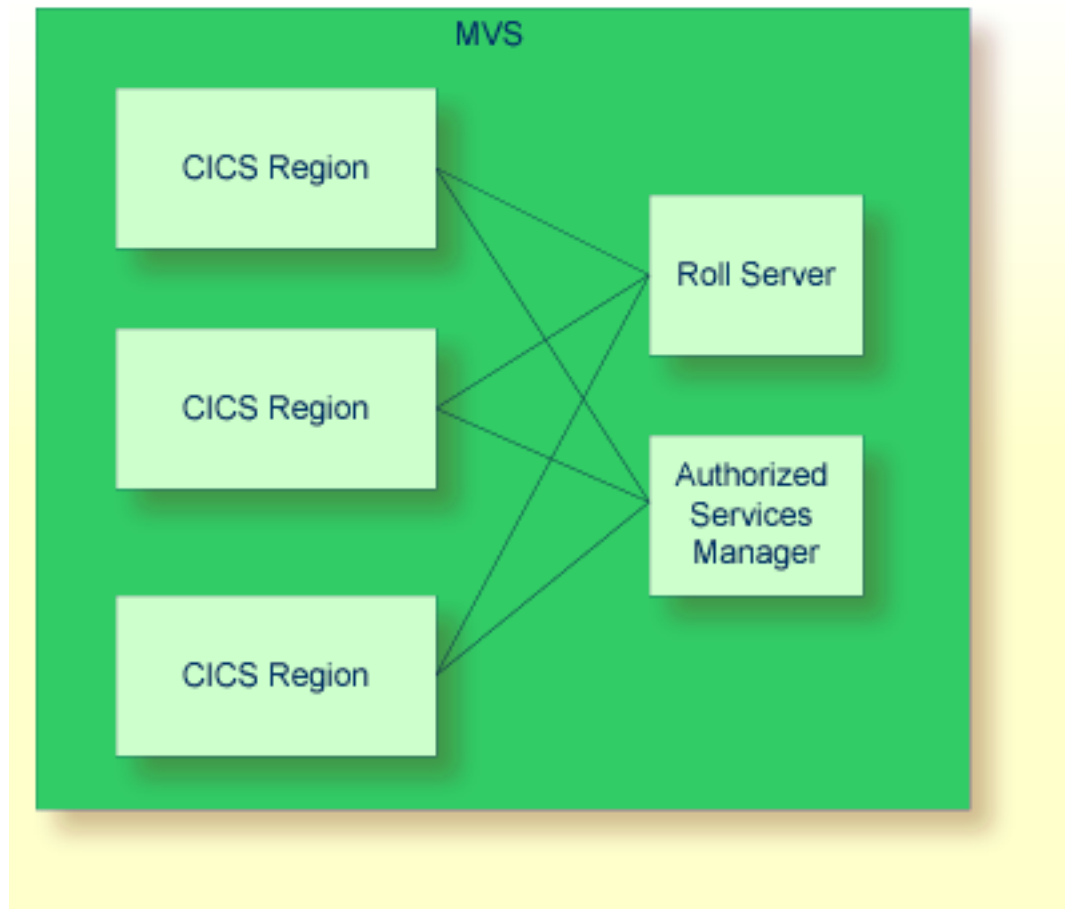
シナリオ 2 :

単一の z/OS、単一の CICS リージョン、単一のロールサーバー



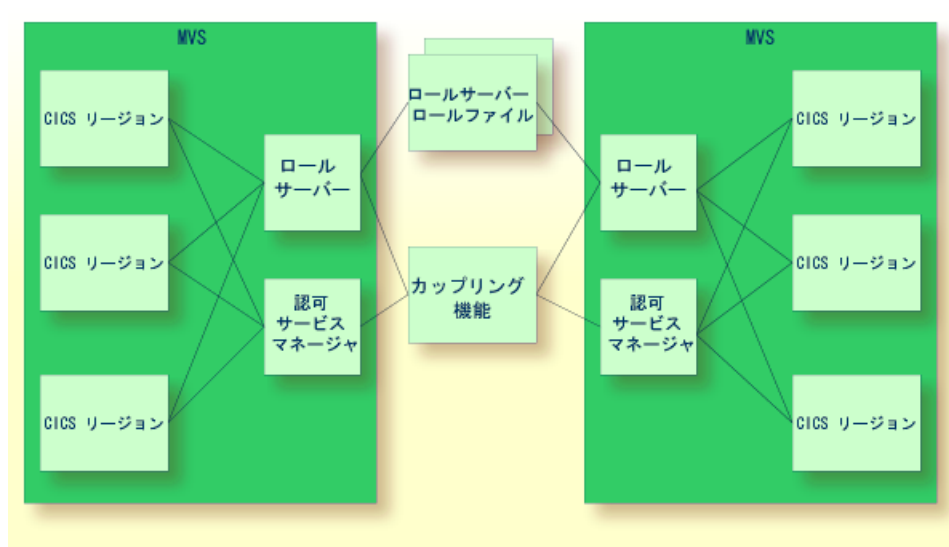
## シナリオ 3 :

単一の z/OS、複数の CICS リージョン、単一のロールサーバー、および（オプションの）認可サービスマネージャ



## シナリオ 4 :

複数の z/OS、複数の CICS リージョン、複数のロールサーバー／認可サービスマネージャ



上記のシナリオに必要なパラメータ設定

モジュール	シナリオ 1	シナリオ 2	シナリオ 3	シナリオ 4
NTBPI (BPI)	TYPE=SWAP、 SIZE= nnn	n/a	n/a	n/a
NCMDIR CICSPLX	NO	NO	YES/MODE	YES/MODE
NCMDIR SIPSERV	NO	NO/YES	YES	YES
NCMDIR ROLLSRV	NO	YES	YES	YES
ロールサーバー	n/a	なし	なし	name
CF 構造名				
認可サービスマネージャ/SIP	n/a	n/a	SIP スロット番号/ サイズ	XCFグループ名/CF 構造名

Natural CICS インターフェイスでは、256 バイトの SIP スロットサイズが必要です。

- 注意:** シナリオ 2、3、4 では、NCI 環境を初期化する最初の Natural のセッションで、SUBSID パラメータを対応するロールサーバーおよび/または認可サービスマネージャの値に設定する必要があります。



## CICS 環境での Natural ストレージスレッド

スレッドとは、Naturalが必要とするすべてのストレージを要求する連続したストレージエリアのことです。これは複数のNaturalユーザーで共有するストレージでも、または、31ビットモード環境では、特定タスク専用の16 MB 境界より上のCICS ユーザーストレージでもかまいません。

各ストレージスレッドは、Natural ユーザーにとっては「アドレススペース」と見なすことができます。Natural ニュークリアスによって発行された各メモリ割り当て要求は、システム制御プログラムに転送され、ストレージスレッドから満たされます。

ストレージスレッドは、Natural CICS インターフェイスが初期化されるときに割り当てられます。ストレージスレッドは、CICS リージョンまたはパーティションに割り当てられる場合には永久（共有）スレッドで、Natural CICS タスクの開始時に割り当てられる場合には排他スレッド（タスク依存のユーザーストレージ）です。

ストレージスレッドの技術は、次の理由から Natural に実装されました。

- 非 31 ビットモードシステムのユーザーストレージにおいて CICS の 64 KB 制限を克服するため。
- ローリングを最適化できるようにするため（以前は、ユーザーストレージの各部分をロール媒体に書き込む必要があったが、現在は連続するストレージエリアがあるため、このエリアはロールアウト前に関連する部分を互いに連続させることにより圧縮される）。
- Natural CICS インターフェイスは、スレッドから Natural セッションのすべての GETMAIN 要求を満たそうと試みます。この方が、CICS サービスコールによる GETMAIN 要求より速く実施することができます。このことは、CICS コマンドレベルの呼び出しに特に当てはまります。これは、CICS EXEC インターフェイスプログラム（EIP）も関係するためです。

スレッドは画面 I/O ごとに所有するタスクによって解放されます。これは会話型タスクと擬似会話型タスクの両方に当てはまります。セッションが再開されると、ストレージが残されていない限り、つまり他のタスクによってスレッドが使用されていない限り、そのストレージは再度スレッドにロールされます。

Natural スレッド選択アルゴリズムは、ロール I/O を最小限にするためにスレッドを使用するバランスを保ちます。つまり、スレッドが多いほど、古いデータが見つかる可能性が高くなり、ロールインを防ぐことができます。しかし、スレッドが多いほど、実ストレージにすべてのスレッドを効率的に保存するために、オペレーティングシステムは頻繁にページングを実行する必要があります。

スレッドはサイズとタイプ、つまり永久的な共有ストレージとして事前に割り当てられているか、または GETMAIN 要求を介しているかに基づいてグループ化されます。使用するスレッドグループの決定は、セッション初期化時に CICS トランザクションコードによって制御されます。同一グループに属するストレージスレッドは、すべて同じサイズです。

スレッドはできるだけ小さく定義する必要があります。Naturalの『ユーティリティ』ドキュメントで Natural ユーティリティ SYSTP のバッファ使用統計機能も参照してください。ただし、スレッドは最大サイズのセッションを保持できる大きさが必要です。

Natural 開発環境と実稼動環境が別にある場合は、実稼動環境では小さいスレッドを使用して（稼動要求にできるだけ速く対応するため）、開発環境では少数の大きいスレッドを使用します（Natural プログラマは通常、大きい Natural のサイズと長い「思考時間」を必要とするため）。

最初の Natural のセッションで、永久的な（共有）スレッドがすべて割り当てられます。

## CICS 環境での Natural のロール機能

---

永久的なストレージスレッドは複数のユーザーによって共有され、GETMAIN を介して割り当てられた大きいスレッドは、長期間保存しない方がよいため、Natural タスクによって端末 I/O ごとにスレッドが解放されます。しかし以前は、端末 I/O が実行された後に Natural セッションを再スタートできるようにユーザーデータを保存する必要がありました。

セッションデータは、以下を使用して保存することができます。

- ローカルロールバッファとロールファイルを含む Natural ロールサーバー
- CICS ロール機能
- Natural スワッププール

さまざまなコンポーネントのシナリオも参照してください。詳細については、Natural の『オペレーション』ドキュメントの「ロールサーバー」を参照してください。

## CICS ロール機能

---


CICS ロール機能はローカルの CICS ストレージ機能です。これらは、CICS のメインまたは補助の一時ストレージ、またはユーザーが以前 CICS に定義した VSAM 相対レコードデータセット（RRDS）です。これらのファイルにより、ロールアウトの発生時に Natural はユーザーの圧縮した動的ストレージを格納することができます。

スワッププールを使用する場合は、CICS ロール機能はスワッププールのバックアップ機能のみを果たします。Natural のパフォーマンスおよび処理能力に影響するため、スワッププールを使用しないときのロール媒体の選択は非常に重要です。

すべての CICS サービス要求により、CICS システムにオーバーヘッドが発生します。このため、ロール機能の CFSIZE/レコードサイズが大きいほど、Natural セッションをロールするための CICS サービスコールが少なくなるため、CPU のオーバーヘッドは小さくなります。一方

で、CISIZE/レコードサイズが大きいと、ロール機能に割り当てられる VSAM バッファスペースも大きくなります。

ロール機能の詳細については、「[パフォーマンスの考慮事項](#)」を参照してください。

 **注意:** ロールサーバーを使用するときは、スワッププールおよび CICS ロール機能は利用できません。

## CICS 環境での Natural ローカルバッファプール

Natural ローカルバッファプールには、Natural モジュールが Adabas または VSAM システムファイルからロードされた後は、Natural モジュールの実行およびコピー時にすべての Natural モジュールが含まれます。

ローカルバッファプールには、Natural プログラムのロード数を最小限にするために、十分なサイズが必要です。しかしローカルバッファプールが大きすぎると、ストレージが浪費されて、ページングのオーバーヘッドが発生する場合があります。

ローカルバッファプールは GETMAIN ストレージとして割り当てられます。つまり、CICS トランザクションサーバーのすべてのバージョンで EXEC CICS GETMAIN SHARED、z/VSE 環境の CICS/VSE では GETVIS 要求です。パーティションまたは関連する CICS DSA で、十分なストレージを利用できる必要があります。

Natural はグローバルバッファプールでも実行できるため、ローカルバッファプールはオプションです。グローバルバッファプールは、バッチモードの *Natural* (z/OS および z/VSE) または *TSO 環境での Natural* または *IMS 環境での Natural* (z/OS のみ) などの他の Natural 環境と共有することができます。

## CICS 環境での Natural スワッププール

Natural スワッププールを使用すると、効率の悪いロール I/O を実行する代わりに、圧縮した Natural セッションをスレッドからメインストレージエリアに「スワップ」することができます。

スワッププールは GETMAIN ストレージとして割り当てられます。つまり、CICS トランザクションサーバーのすべてのバージョンで EXEC CICS GETMAIN SHARED、z/VSE 環境の CICS/VSE では GETVIS 要求です。パーティションまたは関連する CICS DSA で、十分なストレージを利用できる必要があります。

スワップ管理のオプションは、Natural CICS ソースモジュール NCISPCB で、Natural プロファイルパラメータ BPI を使用して設定します。

スワッププールのサイズ、名前、およびキャッシュサイズは、プロファイルパラメータ BPI または Natural パラメータモジュール NATPARM で対応するマクロの NTBPI を使用して指定します。つまり、NCI 環境を初期化する Natural セッションに有効な NTBPI または BPI 設定が使用されます。

スワッププールの詳細については、Natural の『オペレーション』ドキュメントの「Natural スワッププール」、および「[CICS 環境での Natural スワッププールの使用](#)」を参照してください。

### z/OS システムに関する注意事項

スワッププールを使用できるのは、単一の CICS リージョンにおいて CICS 環境の Natural をローカルに実行している場合のみです。しかし、このようなシナリオにおいても、ロールサーバーを代わりに使用の方がよい場合があります。ロールサーバーは CICS リージョンに対して非同期に実行されること、およびロールサーバーは、データスペースにおいてスワッププールより多くのロールバッファを提供できることがその理由です。ロールサーバーを使用するときは、スワッププールおよびロール機能を CICS 環境で利用することはできません。

## CICS 一時ストレージにおける Natural CICS インターフェイスシステム制御レコード

---

Natural CICS インターフェイスは、GETMAIN が実行された永久的なストレージを記憶します。このストレージは、CICS のメイン一時ストレージの NCI システム制御レコードにおいて、EXEC CICS GETMAIN SHARED、または OSCOR/GETVIS ストレージに対するオペレーティングシステムの GETMAIN 要求を介して取得されたストレージです。

これらのシステム制御レコードは次の 2 つの理由から保存されます。

### 1. システムリカバリ

すべての NCI 関連のストレージは NCI システムディレクトリに連鎖されているため、システム制御レコードを使用すると、ストレージ破損の場合にはストレージチェーンを再構築することができます。

### 2. NCI システムディレクトリモジュールの CICS NEWCOPY の後に古い NCI システムをクリーンアップ

NCI システム環境の初期化時に NCI は既存のシステム制御レコードをチェックして、見つかった場合は、新しい環境をインストールする前に関連する永久的なストレージを解放します。

これらの制御レコードのCICS一時ストレージキュー名は *prefixXCR* です。 *prefix* は、Natural CICS コンポーネントに共通の接頭辞で（NCIPARM 生成パラメータ **PREFIX** を参照）、 *X* は 16 進値です。以下のようになります。

x'01'	NCI システムディレクトリ拡張、共有スレッド（TYPE=SHR）、および第 2 の SIR ブロックに関する情報を保持するメインシステム制御レコード（NCMDIR 生成パラメータ <b>USERS</b> を参照）。
x'02'	ファイル入力を介して取得した NCI 共有プロファイルパラメータに関する情報を保持するパラメータシステム制御レコード（NCIPARM 生成パラメータ <b>PRMDEST</b> を参照）。
x'03'	潜在的なスワッププールを含む NCI 環境に属するすべてのローカルプールに関する情報を保持するプールシステム制御レコード。



**重要:** NCI システム制御レコードにローカルの NCI 環境について記述されているため、これらの CICS メイン一時ストレージキューも CICS AOR に保存する必要があります。これは、CICSPlex で Natural を実行する場合に特に当てはまります。

## NCIDIREX-システムディレクトリモジュール名の出口インターフェイス

NCIPRM マクロの **DIRNAME** パラメータを介して明示的に指定されていない限り、Natural CICS インターフェイスシステムディレクトリモジュールの名前は、デフォルトで接頭辞 **CB** になります（NCMPRM マクロの **PREFIX** パラメータを参照）。

NCIDIREX 出口インターフェイスは、ランタイムに Natural CICS インターフェイスのシステムディレクトリモジュールの名前を設定/変更します。これにより、同じ NCI ドライバ/NCIPARM を使用しながら、CICS システム ID、トランザクション ID などに応じて異なるシステムディレクトリモジュールにアクセスすることで、異なる NCI 環境（スレッドグループ/スレッドサイズなど）を使用することが可能になります。

ディレクトリモジュール名の最初の 5 文字も、対応する NCI 環境と関連した CICS 一時ストレージのキュー名の一部として使用されます。このため、CICS リージョンで複数の Natural CICS 環境を実行する場合は、関連するシステムディレクトリモジュール名は最初の 5 文字が異っている必要があります。

NCIDIREX インターフェイス出口は、標準のリンケージ規約を使用して呼び出されますが（レジスタ 13、14、15、および 1）、これに加えて、CICS サービスを呼び出す出口を有効にするために、CICS EIB および EISTG アドレスを保持しているレジスタ 4 および 5 も使用されます。

ソースモジュール **XNCIDIRX** には、サンプルのシステムディレクトリモジュール名の出口が含まれています。

## NCITIDEX - 端末 ID 出口インターフェイス

---

CICS リージョンごとに一意な 4 文字の CICS 端末 ID が、Natural CICS インターフェイスによってセッションキーの一部として使用されます（SIP サーバー、ロールサーバー、CICS 一時ストレージのキュー）。Natural との互換性のために、Natural CICS インターフェイスでは 8 文字のフィールドを使用します。この NCI 端末 ID は、CICS 端末 ID に CICS システム ID を追加することにより、複数の CICS リージョン間で一意にすることができます（NCMPRM マクロの UNITID パラメータを参照）。

または、NCITIDEX 端末 ID 出口インターフェイスを使用して、その NCI 端末 ID を設定することができます。CICS のために（一時ストレージキュー名など）、NCI 端末 ID の最初の 4 文字のみが使用されます。このため、これらの 4 文字の文字列は一意でなければなりません。

NCITIDEX 出口インターフェイスは CICS 環境のセッションマネージャにとって、同じ物理端末で実行されている複数の Natural セッションを識別するために特に重要です。

NCITIDEX 出口によって設定された端末 ID は、Natural CICS インターフェイスによって「外部的」に使用され、Natural 「内部」で使用する Natural システム変数 \*INIT-ID のデフォルトです（\*INIT-ID システム変数は、後で NCIUIDEX/NATUEX1 ユーザー ID 出口インターフェイスによって変更することができます）。

NCITIDEX インターフェイス出口は、標準のリンケージ規約を使用して呼び出されますが（レジスタ 13、14、15、および 1）、これに加えて、CICS サービスを呼び出す出口を有効にするために、CICS EIB および EISTG アドレスを保持しているレジスタ 4 および 5 も使用されます。

ソースモジュール XNCITIDX には、サンプルの端末 ID 出口が含まれています。

### 制限

論理端末 ID の最初の 4 文字が物理端末と一致しない場合、一部の Natural CICS インターフェイス機能は動作しません。

この結果、次のことができなくなります。

- メッセージ交換を使用して論理端末にメッセージを送信することができません。
- 論理端末でセッションをフラッシュするために SYSTP ユーティリティまたは NEP を使用することができません。

## NCIUIDEX - ユーザー ID 出口インターフェイス

---

Natural の NATUEX1 ユーザー出口インターフェイスを使用すると、ユーザーが Natural を使用する権限、およびさまざまな Natural システム変数を設定する権限を与えられているかどうかを判断できます。


Natural ユーザーセッションが開始されると、NATUEX1 インターフェイス出口が標準のリンケージ規約を使用して呼び出されます（レジスタ 13、14、15、および 1）。

CICS 環境では、標準リンケージ規約は CICS サービスコールを発行し、かつ CICS コントロールブロックのアドレス可能性を取得するために十分ではありません。

そのため、Natural CICS インターフェイスでは CICS 環境における NATUEX1 インターフェイス出口としてロードモジュール NCIUEX1 を提供します。このモジュールは CICS でアドレス可能性を設定し、NCIUIDEX インターフェイス出口を呼び出します。この方法として、標準リンケージ規約（レジスタ 13、14、15、および 1）を使用するのに加え、他のレジスタ R4（EIB）、R5（EISTG）、R6（TCTTE）で CICS 関連のアドレスを渡します。

このように、CICS 環境のアドレス可能性を必要とする要求を発行する場合は、標準の NATUEX1 インターフェイスではなく、NCIUIDEX ユーザー ID の出口インターフェイスを使用する必要があります。

ソースモジュール `XNCIUIDX` には、サンプルのユーザー ID 出口が含まれています。

 **重要:** 新しい CICS リリースをインストールするたびに、NCIUIDEX インターフェイス出口を再アセンブルしてリンクする必要があります。

## Natural CICS インターフェイスのデバッグ機能

---

次のトピックについて説明します。

- [TPF パラメータの使用](#)

■ 非同期 Natural セッションの使用

TPF パラメータの使用

対応するオプションに "1" を指定することによって、ドライバに固有のオプションに対してダイナミックパラメータ TPF=(TPF1, TPF2, TPF3, TPF4, TPF5, TPF6, TPF7, TPF8) を設定することができます。

サポートされるオプションは、以下のとおりです。

TPF1	DCI を介してではなく、TWA および CICS COMMAREA の Adabas パラメータを指定した EXEC CICS LINK を介して、Adabas リンクモジュールを起動します。  CEDF を介して Adabas 関連の問題のデバッグを有効にします。
TPF2	Natural スワッププール全体をダンプします。  このパラメータ設定では、Natural スワッププール全体が CICS トランザクションダンプに含まれます。
TPF3	Natural バッファプール全体をダンプします。  このパラメータ設定では、Natural バッファプール全体が CICS トランザクションダンプに含まれます。  <b>注意:</b> 通常、Natural バッファプールはダンプには必要ありません。セッションに関連するバッファプールからのすべてのオブジェクトがダンプされるからです。このため、このオプションは、バッファプールに問題がある場合のみ必要になります。
TPF4	EDITOR バッファプール全体をダンプします。  このパラメータ設定では、EDITOR バッファプールが CICS トランザクションダンプに含まれます。
TPF6	NCI で端末 I/O エラーを処理します。  このパラメータ設定では、NCI は端末 I/O エラーに関して Natural に制御を渡さずに、エラーを単独で処理します。この結果、エラーメッセージ NT06~NT13 のいずれかが発生します。
TPF7	NCI システムエラーの場合に強制的に異常終了します。  このパラメータ設定では、エラーメッセージ NSxx, NIxx, NRxx, または NUSnnnn の場合にプログラムチェックが強制的に実行されます。この設定は、異常終了をインターセプトするデバッグ用ツールがアクティブな場合に特に役に立ちます。その後、エラーをオンラインで直接分析することができます。

0 を指定すると（省略可）、対応するオプションは設定されません。以下に例を挙げます。

TPF=(0,0,0,1) は、TPF=(,,,1) と同じ意味です。



## 非同期 Natural セッションの使用

Natural を開始するダイナミックパラメータ文字列の最初の 5 文字が ASYN, の場合、セッションが端末内かどうかに関係なく、Natural CICS インターフェイスは常に非同期 Natural セッションを設定します。

これはテスト目的の場合、特に EDF または他のデバッグ用ツールがインストールされている場合に役に立ちます。

## Natural CICS インターフェイス CICS TWA の使用

Natural トランザクションは、すべて 128 バイトの TWA サイズで定義されますが、Natural CICS インターフェイスは、CICS トランザクションワークエリア (TWA) の最初の 88 バイトだけを使用して、Natural で以下の機能进行处理します。

- Adabas で Adabas パラメータリスト (32 バイトまで) を呼び出す。Natural CICS インターフェイスは、Adabas を呼び出す前に TWA の内容を保存して Adabas を呼び出した後に復元します。
- 外部プログラムでパラメータリストのアドレスポインタ (20 バイトまで。Natural の CALL ステートメントを参照) を呼び出す。Natural CICS インターフェイスは、外部プログラムを呼び出す前に TWA の内容を保存して、外部プログラムを呼び出した後に TWA の呼び出し部分を復元します。
- 終了メッセージおよび潜在的な終了データ (80 バイト。Natural の『オペレーション』ドキュメントの「バックエンドプログラム呼び出し規則」を参照) のためにバックエンドプログラムを起動する。
- セッション終了時の終了メッセージおよび潜在的な終了データ、および、Natural ダイアログステップの終了時に低い値に完全にリセットされる終了メッセージエリア (セッション終了時およびダイアログステップの終了時に 80 バイト) のために、「LINK」フロントエンドの呼び出し元に制御を返す。
- CICS タスク開始時に LE 情報を渡す (タスク開始時に 88 バイトまで)。

ユーザープログラム (フロントエンド、バックエンド、呼び出された外部プログラム) で CICS TWA を利用して Natural とは別に通信を行うこともできますが、ユーザープログラムでは Natural で使用する TWA の部分を使用しないでください。このような場合は、Natural トランザクションの TWA のサイズを増加して、最初の 128 バイト以外の TWA の部分を使用することをお勧めします。



# 5 Natural CICS 生成パラメータ

---

■ NCISPCB 生成パラメータ .....	27
■ NCMDIR マクロパラメータ .....	27
■ NCMTGD マクロパラメータ .....	32
■ NTSWPRM マクロパラメータ .....	38
■ NCIPAPM 生成パラメータ .....	38
■ NCMPRM マクロパラメータ .....	38

Natural CICS インターフェイスドキュメントのこの部分では、Natural CICS 生成パラメータについて説明します。

### CICS テーブルの参照

適切な場合は、CICS テーブル (DCT、FCT、PCT、PPT、TCT、TST など) への参照を、以下に示す対応する項目への参照と見なすことができます。

- アセンブリタイプのリソース定義
- CEDA を介したオンラインリソース定義
- DFHCSDUP を介したバッチリソース定義

### 関連ドキュメント

- Natural の『インストール』ドキュメントの「*Natural CICS* インターフェイスのインストール」
- さまざまな TP モニタ固有の関数を提供する Natural ユーティリティ SYSTP
- CICS 環境における Natural の操作および個別のコンポーネントの詳細については、『TP モニタインターフェイス』ドキュメントの以下のセクションを参照してください。
  - *Natural* に対するノードエラープログラムに関する考慮事項
  - *CICS 3270 Bridge* の考慮事項
  - *Natural CICS* の特殊な機能
  - *Natural CICS* のサンプルプログラム
  - *NCIUIDEX* ユーザー ID 出口インターフェイス
  - ユーザープログラムからの *Natural* の起動
  - *CICS* 環境での *Natural* の非同期処理
  - *CICS* 環境での *Natural* セッションのログ
  - パフォーマンスの考慮事項
  - *Natural CICS* インターフェイスのデバッグ機能
  - *CICS* 環境での *Natural* の出力ファイルおよびワークファイル

## NCISPCB 生成パラメータ

Natural CICS インターフェイスシステムディレクトリは NCISPCB ソースモジュールのアセンブルおよびリンクによって生成されます。Natural の『インストール』ドキュメントの「Natural CICS インターフェイスのインストール」、「システムディレクトリの作成」を参照してください。

NCISPCB には、次のマクロが含まれています。

- NCMDIR
- NCMTGD
- NTSWPRM

これらのマクロの目的、および NCMDIR および NCMTGD マクロで指定できる個別のパラメータについては、以降のセクションで説明します。

## NCMDIR マクロパラメータ

NCMDIR マクロは必須であり、NCISPCB ソースモジュールで最初のマクロとして指定する必要があります。これには、システムに対するさまざまなオプションが含まれています。NCMDIR マクロで指定できる個別のパラメータを、以下で説明します。

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

### CICSPLX -CICS アプリケーションリージョンの切り替え

このパラメータは、z/OS 環境でのみ適用可能です。


可能な値は次のとおりです。

値	説明：
YES	Natural CICS インターフェイスは、ローカルの CICS アプリケーション所有領域（AOR）外部の擬似会話型画面 I/O を介して、セッション関連のすべてのデータをセッション情報レコード（SIR）およびセッションデータとして保存します。これにより、CICS AOR を切り替えることができます。  このパラメータを YES に設定する場合は、プロファイルパラメータ ADAMODE を 0 より大きい値に設定する必要があります。
MODE	この設定は YES とほぼ同じ意味があります。違う部分は、CICSPLX=MODE によって ADAMODE=0 のプロファイルパラメータ指定が許可されることです。つまり、CICSAOR 切り替えは使用できませんが、Natural セッションは、MRO 環境において CICS AOR が再スタートされても存続されます。

値	説明：
NO	重要な Natural セッションデータはローカルの CICS AOR に保存されます。これによって、CICS AOR の切り替えが事実上無効になります。  これはデフォルト値です。

Natural PLEX サポートが意味するのは、Natural CICS セッションは、CICS タスク終了時に CICS アプリケーションリージョンに存在するすべてのフットプリントを削除するという事です。これは、その後このリージョンを使用しない可能性があるためです。したがって、Natural CICS セッション関連のすべてのデータは、CICS アプリケーションリージョン外部に保存する必要があります。このため CICS 環境の Natural は CICS タスク終了時に、セッション情報レコード (SIR) を認可サービスマネージャの SIP ハンドラに、セッションデータを Natural ロールサーバーに渡します。これに加えて保留されているすべてのモジュール、つまり、Natural にリンクされていないが、RCA モジュールまたは Adabas リンクモジュールとして標準のリンケージ規約を介して直接起動されたモジュールは、CICS タスク終了時に解放する必要があります。また、再スタート情報は、COMARET=YES の場合は CICS 端末所有領域 (TOR) に、COMARET=NO の場合は CICS データ所有領域 (DOR) に (すべての参加 CICS AOR で共有) 保存する必要があります。詳細については、COMARET パラメータを参照してください。

YES または MODE が指定されて NCMDIRSUBSID パラメータが設定されていない場合は、NCI 環境を初期化する Natural セッションに有効な Natural プロファイルパラメータ SUBSID の値が取られます。

 **注意:** このパラメータを YES または MODE に設定すると、SIPSERV および ROLLSRV パラメータが自動的に YES に設定されます。

## ROLLSRV - ロールサーバーのロール

このパラメータは、z/OS 環境でのみ適用可能です。

可能な値は次のとおりです。

値	説明：
NO	CICSPLX=NO および SIPSERV=NO の場合、これがデフォルト値です。CICSPLX または SIPSERV が YES の場合、ROLLSRV=YES に設定されます。
YES	YES を指定すると、Natural CICS インターフェイスは Natural ロールサーバーをロール機能としてのみ使用します。

画面 I/O を介して Natural セッションデータを保存および回復するために Natural ロールサーバーを使用する場合は、CICSPLX および SIPSERV パラメータの両方が NO に設定されているときに、このパラメータを YES に設定する必要があります。YES が (強制的に) 指定されて NCMDIRSUBSID パラメータが設定されていない場合は、NCI 環境を初期化する Natural セッションに有効な Natural プロファイルパラメータ SUBSID の値が取られます。

Natural CICS インターフェイスのために、Natural プロファイルパラメータ SUBSID が適用されるのは、ダイナミックに指定された場合、またはパラメータモジュールで指定された場合に限りです。プロファイルパラメータ SYS または PROFILE によるパラメータ文字列または代替パラメータモジュールで（プロファイルパラメータ PARM で指定したように）指定された場合は無視されます。

## SIPSERV - 認可サービスマネージャのセッション情報プール

このパラメータは、z/OS 環境でのみ適用可能です。

可能な値は次のとおりです。

値	説明：
NO	CICSPLX=NO の場合、これがデフォルト値です。CICSPLX が NO でない場合は、SIPSERV=YES に設定されます。
YES	Natural CICS インターフェイスは、セッション情報レコード (SIR) を、認可サービスマネージャのセッション情報プールに保存します。

このパラメータが YES に（強制的に）設定されると、Natural セッション情報レコードは CICS リージョン外に保存されるため、Natural は擬似会話型画面 I/O 後に CICS アプリケーションリージョンを切り替えることができます。

YES が（強制的に）指定されて NCMDIRSUBSID パラメータが設定されていない場合は、NCI 環境を初期化する Natural セッションに有効な Natural プロファイルパラメータ SUBSID の値が取られません。

Natural CICS インターフェイスのために、Natural プロファイルパラメータ SUBSID が適用されるのは、ダイナミックに指定された場合、またはパラメータモジュールで指定された場合に限りです。プロファイルパラメータ SYS または PROFILE によるパラメータ文字列または代替パラメータモジュールで（プロファイルパラメータ PARM で指定したように）指定された場合は無視されます。

 **注意:** このパラメータが YES の場合、ROLLSRV パラメータは、まだ指定されていない場合は強制的に YES に設定されます。

## SUBSID - サブシステム ID

このパラメータは、z/OS 環境でのみ適用可能です。

可能な値は次のとおりです。

値	説明：
SUBSID=xxxx	Natural ロールサーバーおよび／または認可サービスマネージャのサブシステム ID を定義します。

このパラメータは、Natural ロールサーバーおよび／または認可サービスマネージャに使用される Natural サブシステム ID を定義します。このパラメータを指定しないと、Natural プロファイルパラメータ SUBSID の値が取られません。

Natural CICS インターフェイスのために、Natural プロファイルパラメータ SUBSID が適用されるのは、ダイナミックに指定された場合、またはパラメータモジュールで指定された場合に限りです。プロファイルパラメータ SYS または PROFILE によるパラメータ文字列または代替パラメータモジュールで（プロファイルパラメータ PARM で指定したように）指定された場合は無視されます。

### TSKEY - Natural CICS 一時ストレージキーの接頭辞

このパラメータは、一時ストレージキューの定数の接頭辞を定義します（以下の説明を参照）。

可能な値は次のとおりです。

値	説明：
(xxxx, yyyy)	TSKEY=(xxxx, yyyy) はロールデータの接頭辞を定義しますが、yyyy は擬似会話型の再スタートデータの接頭辞を定義します。
(NAT4, NCOM)	これはデフォルト値です。

CICS 一時ストレージ（メインまたは補助）を、Natural CICS インターフェイスのロール機能または Natural 擬似会話型タスクのための通信エリア（NCMPRM マクロパラメータ COMARET で説明）に使用する場合、タスク依存の一意な一時ストレージのキューに対する名前を指定する必要があります。

これらのキュー名は、定数である 4 バイトのキーおよびタスク関連のキーから構成されます。端末依存タスクの場合はこのタスク関連のキーは端末 ID に対応し、非同期の非端末タスクの場合は CICS の一意なタスク番号に対応します。一時ストレージキュー名の定数の接頭辞は TSKEY パラメータによって定義されます。

Natural CICS インターフェイスでは、4 バイトの接頭辞が、ロールデータ用と擬似会話型の再スタートデータ用に 2 つ必要です。xxxx は、ロールデータの接頭辞を定義します。yyyy は、擬似会話型再スタートデータの接頭辞を定義します。これらの 2 つの接頭辞は互いに異なり、CICS 環境の Natural で排他的でなければなりません。

CICSplex 環境で実行する場合は、Natural セッション再スタート情報用の CICS 一時ストレージの接頭辞は、参加するすべての CICS リージョンにおいてアクセス可能な REMOTE/SHARED として CICS TST で定義する必要があります。



## TSRECSZ - メインおよび補助の一時ストレージのレコードサイズ

このパラメータは、CICS一時ストレージをNatural CICSインターフェイスのロール機能として使用する場合に、データのロールにおける最大レコード長を定義します。

可能な値は次のとおりです。

値	説明：
(nnnnn, mmmmm)	<p>1つ目のサブパラメータ <i>nnnnn</i> は、CICSのメインの一時ストレージに適用されます。この値は、4096～32763の範囲内または0、またはキーワード MAX、YES、または NO のいずれかである必要があります。</p> <ul style="list-style-type: none"> <li>■ 0以外の数値の場合は、その値が無条件に使用されます。</li> <li>■ 0またはNOに設定した場合は、CICSのメインの一時ストレージをNaturalロール機能に使用することはできません。</li> <li>■ MAXまたはYESに設定した場合は、レコードサイズが32763に設定されます。</li> </ul> <p>2つ目のサブパラメータ <i>mmmmm</i> はCICSの補助の一時ストレージに適用されます。この値は、3976～32763の範囲内または0、またはキーワード MAX、YES、または NO のいずれかである必要があります。</p> <ul style="list-style-type: none"> <li>■ 0以外の数値の場合は、その値が無条件に使用されます。MAXに設定すると、レコードサイズが32763に設定されます。</li> <li>■ NOに設定すると、CICSの補助の一時ストレージをNaturalロール機能に使用することはできません。</li> <li>■ 0またはYESに設定すると、Natural CICSインターフェイスは、補助の一時ストレージのコントロールインターバルに適合するレコード長を設定します。つまり、CIサイズからVSAM制御情報とCICS制御情報を引いた長さになります。</li> </ul> <p>ユーザー定義のレコードサイズがCIサイズより大きい場合は、スパンドレコードの書き込みによる追加のCICSオーバーヘッドが生じますが、(論理的)ロールI/Oが少なくなります。</p>
(32748, 0)	これはデフォルト値です。

## USERS - セッション情報レコード

このパラメータは、セッション情報レコードスロット (SIR) の数を指定します。

可能な値は次のとおりです。

値	説明：
( <i>nnnnn</i> , <i>mmm</i> )	<p>サブパラメータ <i>nnnnn</i> は、Natural CICS ディレクトリモジュール自体に保持される SIR の数を定義します。 <i>nnnnn</i> は 1~32767 の範囲で指定する必要があります。ディレクトリの SIR スロットがいっぱいになると、Natural CICS インターフェイスは、<i>mmm</i> によって定義された SIR の数を保持するために十分な大きさの CICS 共有ストレージセグメントを取得します。この範囲は 0~255 の範囲内である必要があります。</p> <p>サブパラメータ <i>mmm</i> が 0 または省略された場合、システムディレクトリに空いている SIR スロットがなければ、システムは SIR 用に追加のストレージを取得しません。この場合、Natural CICS システムは、1 つ目のサブパラメータによって指定されたユーザー数に実質的に制限されます。</p> <p><i>mmm</i> に 0 以外の値が指定された場合、必要に応じて、第 2 のストレージセグメントが自動的に割り当てられます。割り当てられた第 2 のストレージセグメントは、必要がなくなると再度解放されます。</p>
(100,20)	これはデフォルト値です。

Natural CICS インターフェイスは、すべてのアクティブな Natural セッションに関する情報を永続的に保持します。セッションごとに、セッション情報レコード (SIR) が保存されます。

これらの SIR は、以下の場所に保存されます。

- 並列シスプレックス環境で実行する場合はカップリング機能
- 単一の z/OS システム内の複数の CICS リージョンで実行する場合は *Natural* 認可サービスマネージャのデータスペース
- 単一の CICS AOR で (ローカルに) 実行する場合は CICS リージョンのメインストレージ

ただし、Natural セッションが CICS リージョンでアクティブな場合は常に、現在のアプリケーションリージョンの SIR スロットが使用されます。

単一の CICS AOR でローカルに実行する場合、**USERS** パラメータは、すべての Natural セッションに適用されます。CICSplex 環境で実行する場合、**USERS** は、参加している CICS AOR のそれぞれで現在アクティブな Natural セッションのサブセットに適用されます。

## NCMTGD マクロパラメータ

NCMTGD マクロは必須であり、スレッドグループごとに指定する必要があります。Natural CICS インターフェイスにより、スレッドのグループを定義することができます。これらのグループは、セッション初期化時に CICS トランザクション ID によって制御/選択されます。共通のスレッドサイズはグループごとに異なる場合があり、また各グループは異なるオプションを持つことができます。スレッドグループの定義は、単一のセッションだけでなくシステム全体に関係するため、Natural CICS システムディレクトリの一部となっています。

NCMTGD マクロで指定できる個別のパラメータを、以下で説明します。

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

### PFKEY - スレッドグループの PF/PA キー


このパラメータは、単一の CICS トランザクションまたは CICS トランザクションのリストを定義します。

可能な値は次のとおりです。

値	説明：
xxx	xxx に指定できる値は、PF1～PF24、PA1～PA3 です。
(xxx,xxx,...)	キーのリストも指定することができます。リストは、PFKEY=(PF12, PF14) のようにカッコで囲む必要があります。

デフォルト値はありません。

セッションを開始すると、Natural CICS インターフェイスは、すべてのスレッドグループの定義をスキャンして、現在のトランザクション ID、または PF または PA キーを探します。見つからない場合は、最初のスレッドグループがデフォルトになります。

 **注意:** 少なくとも 1 つのトランザクション ID (文字または 16 進形式) またはアテンション識別子を開始する 1 つのトランザクションを、デフォルトグループとして使用される最初のグループを除く、すべてのグループに対して指定する必要があります。

### PRIMERF - Natural CICS の主なロール機能

このパラメータは、関連するスレッドグループで定義されているすべてのタスクに対して、Natural CICS インターフェイスの主なロール機能を定義します。このため、このパラメータは `TYPE=NONE` のスレッドグループには適用されません。

可能な値は次のとおりです。

値	説明：
VSAM	Natural CICS インターフェイスの VSAM RRDS ロールファイルが、主なロール機能として使用されます。CICS の補助の一時ストレージは第 2 のロール機能と見なされます。これは、主なロールファイルのすべてがいっぱいになった場合、または利用できない場合に使用されることを意味します。
AUX	CICS の補助の一時ストレージが、Natural CICS インターフェイスの主なロール機能として使用されます。
MAIN	CICS のメインの一時ストレージが、Natural CICS インターフェイスの主なロール機能として使用されます。


値	説明:
NONE	関連するセッションがロールされません。NONE は、TYPE=SHR グループ、およびTYPE=SHR グループを再定義する TYPE=ALIAS のグループには無効です。

デフォルト値はありません。

Natural ロールサーバーを使用する場合、このパラメータは無視されます。ロールサーバーとの Natural セッションをロールなしの会話型で強制的に実行すると、値 NONE が使用されます。

注意するポイント:

- VSAM RRDS ロールファイルが CICS システムで利用できない場合は、PRIMERF=VSAM と PRIMERF=AUX は同じ効果があります。
- 補助の一時ストレージが CICS システムで定義されていない場合は、PRIMERF=AUX と PRIMERF=MAIN は同じ効果があります。
- 補助の一時ストレージが CICS システムで定義されていない場合に、PRIMERF=VSAM を指定すると、VSAM RRDS ロールファイルが利用できないかいっぱいになったとき、CICS のメインの一時ストレージが第 2 のロール機能と見なされることを意味します。
- CICS のメインの一時ストレージをロール機能として使用する場合は、レコードサイズは TSRECSZ パラメータで定義されます。

 **重要:** PRIMERF=NONE に定義されているスレッドグループに関するセッションは、ロール機能がなくロールができないため、設計上会話型になります。

## THRDSZE - スレッドサイズ

このパラメータは、TYPE=GETM および TYPE=SHR グループに共通するスレッドサイズを定義します。

可能な値は次のとおりです。

値	説明:
nnn	スレッドサイズ nnn は、40 以上に設定することができます。

デフォルト値はありません。

このパラメータは、Natural で利用できる論理的なスレッドサイズを定義することに注意してください。ただし Natural CICS インターフェイス NCI は、内部管理のために論理的なスレッドサイズに 2 KB をさらに追加します。これは、GETMAIN 要求スレッドの物理的なスレッドサイズまたは長さが、THRDSZE 値より 2 KB 大きいことを意味します。

TYPE=GETM の場合は、先頭および末尾の CICS ストレージアカウンティングエリア (SAA) 用に追加の 16 バイトが必要です。

**重要な注意：**

1. GETMAIN が 512 KB を超える場合は、CICS によってこれらのストレージが MB 境界で位置合わせされます。
2. トランザクションアイソレーションを使用する場合は、CICS は、内部的に EUDSA で 1 MB 「ページ」を使用します（詳細については『CICS Performance Guide』を参照）。

これらの2つの事実からストレージのフラグメントが発生するため、CICS で適切な EDSALIM を設定するときには注意してください。

**THREADS - スレッドグループごとのスレッドまたはタスクの数**

このパラメータは、以下に説明するようにスレッドまたはタスクの数を指定します。

可能な値は次のとおりです。

値	説明：
<i>nnn</i>	スレッド数は、510 以下に設定することができます。

デフォルト値はありません。

TYPE=SHR のスレッドグループについては、THREADS パラメータは必須であり、インストール時に GETMAIN (CICS バージョンに応じて SVC または SHARED) を介して割り当てられるスレッド数を定義します。

TYPE=GETM および TYPE=NONE のスレッドグループについては、THREADS パラメータはオプションであり、スレッドグループごとに同時にアクティブにできる Natural タスクの最大数を決定します。

スレッドグループタイプについては、THREADS パラメータは、TYPE=SHR のスレッドグループとは違いストレージの使用状況を制御しません（「[ストレージ使用の制御](#)」も参照）。

スレッドグループごとのスレッドの数またはタスクの数は、スレッドコントロールブロック (TCB) を指定して定義します。

TYPE=SHR のスレッドグループについては、各スレッドは TCB に密接に接続されています。スレッドは関連する TCB でキューに入れることにより共有されます。TYPE=GETM および TYPE=NONE のスレッドグループは、アクティブ化するためにのみ TCB でキューに入れられます。

TYPE=SHR のスレッドグループを持つセッションはスレッドを使用しますが、他のセッションタイプは、スレッドがすでに割り当てられている (TYPE=GETM) またはスレッドが割り当てられていない (TYPE=NONE) TCB を使用します。

THREADS パラメータが 0 でない場合、Natural プロファイルパラメータ DBROLL および MAXROLL、および CMROLL への呼び出しでは、TYPE=GETM/NONE のスレッドグループに対する処理が異なります。この場合スレッドを解放できないため、保持されている TCB リソースが解放され、こ

れによりストレージに保存されているセッションデータを持つセッションがアクティブになります。

### TRAN - スレッドグループのトランザクション ID

このパラメータは、単一の CICS トランザクションまたは CICS トランザクションのリストを定義します。

可能な値は次のとおりです。

値	説明：
(下記を参照)	Natural の PCT に定義された 1 つ以上の CICS トランザクションコード。

デフォルト値はありません。

TRAN パラメータのトランザクション ID は、文字フォーマットである必要があります。英数字以外の文字を含むトランザクション ID は、アポストロフィで囲む必要があります。

セッションを開始すると、Natural CICS インターフェイスは、すべてのスレッドグループの定義をスキャンして、現在のトランザクション ID、または PF または PA キーを探します。見つからない場合は、最初のスレッドグループがデフォルトになります。

トランザクション ID のリストは、TRAN=(NATU, XYZ) のようにカッコで囲む必要があります。

**注意:** 少なくとも 1 つのトランザクション ID (文字または 16 進形式) またはアテンション識別子を開始する 1 つのトランザクションを、デフォルトグループとして使用される最初のグループを除く、すべてのグループに対して指定する必要があります。

### TYPE - グループのスレッドタイプ

このパラメータは、特定のグループに対して使用されるスレッドのタイプを定義します。

可能な値は次のとおりです。

値	説明：
SHR	<p>CICS 共有ストレージスレッドが使用されます。スレッドグループに利用できるスレッドは、このグループに対して定義されたすべての CICS トランザクションによって共有されます。CICS タスクを開始するときのスレッド選択は、ENQUEUE/DEQUEUE 技術によって行われます。現在利用できるスレッドがない場合は、このスレッドグループに対する待機キューが保持されます。</p> <p>これはデフォルト値です。</p> <p>SYSplex 環境で実行する場合は、Natural パラメータ RELO=OFF により TYPE=SHR スレッドを持つセッションが会話型に設定され、CICS リージョンの切り替えを防ぎます。</p>

値	説明：
GETM	<p>GETMAINを介して割り当てられたスレッドが使用されます。つまりスレッドは、実質的にはCICS GETMAIN 操作の EXEC CICS GETMAIN FLENGTH を実行して、スレッドグループの共通スレッドサイズで取得されます。GETMAIN を介して割り当てられたスレッドを使用することにより、各 Natural タスクは終了するまで、つまり擬似会話型タスクの場合は画面I/Oから画面I/Oの間、排他的なスレッドストレージを利用することができます。</p> <p>Natural パラメータ RELO=OFF または PSEUDO=OFF が指定されると、GETMAIN を介して割り当てられたスレッドを使用するタスクは会話型に設定されます。これは、スレッドに対して FREEMAIN を実行した後に、その後の GETMAIN によってメモリ内に同じストレージを取得できることが保証されないためです。しかし、GETMAIN を介して割り当てられたスレッドストレージは所有するタスクに排他的に属しているため、このようなタスクはロール不能として定義することができます (PRIMERF パラメータを参照)。つまり、特定のスレッドは、Natural セッションが終了するまで特定のタスクに属することになります。この場合、タスクは設計上会話型になり、ロールは実施されません。</p>
NONE	<p>スレッドはこのスレッドグループに定義されたトランザクションによって使用されず、すべての Natural GETMAIN 要求は CICS に直接渡されて、EXEC CICS GETMAIN FLENGTH 要求として処理されます。設計上、このようなタスクはロールできないため会話型になります。</p>
ALIAS	<p>現在の NCMTGD マクロでは、以前の NCMTGD のマクロ指定によって定義されたものとは異なるスレッドグループ用のオプションを指定できます。ただし、TYPE=GETM および TYPE=SHR のスレッドグループのみ、1つ以上の NCMTGD TYPE=ALIAS のマクロ要求によって再定義することができます。</p> <p>99 個までのスレッドグループがサポートされます。これは ALIAS 以外の TYPE を持つ 99 個までの NCMTGD マクロ指定が認識されることを意味します。</p>

## XTRAN - スレッドグループに対する 16 進形式のトランザクション ID


このパラメータは TRAN パラメータと等価ですが、トランザクション ID が 16 進形式である必要があります。

可能な値は次のとおりです。

値	説明：
(下記を参照)	<p>設定可能値：Natural の PCT に定義されている 1つ以上の CICS トランザクションコード。</p>

デフォルト値はありません。

16 進形式のトランザクション ID のリストは、XTRAN=(D5C1E3E4, E7E8E9) のようにカッコで囲む必要があります。

-  **注意:** 少なくとも 1つのトランザクション ID (文字または 16 進形式) またはアテンション識別子を開始する 1つのトランザクションを、デフォルトグループとして使用される最初のグループを除く、すべてのグループに対して指定する必要があります。

## NTSWPRM マクロパラメータ

---

NTSWPRM マクロは、スワッププールのさまざまな側面を定義するために使用します。スワッププールを使用しない場合は、このマクロは省略してください。詳細については、Naturalの『オペレーション』ドキュメントの「Natural スワッププール」を参照してください。

## NCIPAPM 生成パラメータ

---

Natural CICS インターフェイスのパラメータモジュールは、NCIPARM ソースモジュールをアセンブルして生成します。Naturalの『インストール』ドキュメントの「Natural CICS インターフェイスのインストール」、「Natural CICS インターフェイスパラメータモジュールの作成」を参照してください。このパラメータはNCMPRM マクロ定義を保持します。

NCMPRM マクロ、およびこのマクロで指定できるパラメータの目的については、次のセクションで説明しています。

## NCMPRM マクロパラメータ

---

NCMPRM マクロは、CICS 環境で関連するすべての Natural セッションのオプションを決定します。このマクロは Natural CICS パラメータモジュールの一部です。この作成手順は、Naturalの『インストール』ドキュメントにある、CICS 環境での Natural のインストール手順の該当項目に記載されています。

すべてのデフォルト値を含むサンプルの NCMPRM マクロ定義は、データセット NCI $nnn$ .SRCE の NCIPARM ソースモジュールに含まれています。

NCMPRM マクロの個別のパラメータを、以下で説明します。

BACKEND | BACKOUT | CHAP | CNTCALL | COMACAL | COMAMSG | COMARET | CONSOLE | DIRNAME | DUPTID  
| FLDLEN | LOGDEST | MSGDEST | MSGPFX | MSGTRAN | PREFIX | PRMDEST | PSTRNID | RCVASYN |  
RESENDC | RESENDS | RJEDEST | RJEUSER | SIGNON | SLCALL | SLNOHLD | SNDLAST | TERMVAR | TRANCHK  
| UCTRAN | UNITID



## BACKEND - バックエンドプログラムの起動制御

このパラメータは、セッションが正常に終了または異常終了した後に、指定されたバックエンドプログラムまたはトランザクションを起動するかどうかを定義します。

BACKEND パラメータには2つのサブパラメータがあります。2つ目のサブパラメータはオプションです。このサブパラメータは、端末エラーが発生した場合にバックエンドプログラムを起動するかどうかを制御します。これには、NEP によって開始されるセッションクリーンアップタスクも含まれます。

設定可能な値はいずれのサブパラメータも YES/NO ですが、デフォルト値は異なります。

値	説明：
YES	BACKEND=(YES,NO) と同じです。  BACKEND パラメータを省略した場合は、これがデフォルトです。潜在的なバックエンドプログラムまたはトランザクションが、特にタスクが異常終了した後は常に起動されますが、端末エラーの場合には起動されません。  バックエンドプログラムが起動されると、Natural 終了メッセージおよびリターンコードが CICS トランザクションワークエリア (TWA) に渡されます。さらに、COMAMSG パラメータで説明しているように、同じ情報を CICS COMMAREA に渡すことができます。
(YES,YES)	BACKEND=(,YES) と同じです。潜在的なバックエンドプログラムまたはトランザクションが、特に端末エラーを含む異常終了の後に常に起動されます。
NO	BACKEND=(NO,NO) に設定します。潜在的なバックエンドプログラムまたはトランザクションが、Natural セッションが正常に終了した場合、つまり Natural の終了メッセージが出力される場合のみ起動されます。

## BACKOUT - 回復不可能な異常終了の場合のトランザクションのバックアウト

このパラメータは、Natural CICS インターフェイスが EXEC CICS SYNCPOINT ROLLBACK 呼び出しによってトランザクションのバックアウトを実行するかどうかを定義します。

可能な値は次のとおりです。

値	説明：
YES	保留中のすべてのファイル更新がバックアウトされます。  これはデフォルト値です。
NO	保留中のすべてのファイル更新がコミットされます。

異常終了による出口のために、Natural CICS インターフェイスはすべての異常終了をインターセプトします。異常終了がリカバリ可能でない場合は、異常終了するセッションのすべてのリソースが解放され、セッションは EXEC CICS RETURN を介して終了します。つまり、CICS に関

しては「正常に」終了します。したがって、タスクの終了時に「保留中」のファイル更新は、CICSによって自動的にバックアウトされません。

### CHAP - タスクのディスパッチ優先度の変更

このパラメータは、Natural CICS インターフェイスで DBROLL および/または MAXROLL 呼び出し制限に達している長期間にわたるタスクを処理する方法を定義します。可能な値は次のとおりです。

値	説明：
YES	DBROLL および/または MAXROLL 呼び出し制限に達するたびに、タスクのディスパッチ優先度が1ずつ減らされます。元のタスクディスパッチ優先度は、次の画面 I/O で再設定されます。
NO	セッションが中断されます。  これはデフォルト値です。

### CNTCALL - データをコンテナに自動的に渡す CICS 呼び出し

SET CONTROL 'P=C' を指定すると、CALL ステートメントのパラメータデータは、パラメータデータポインタではなく EXEC CICS LINK の CICS COMMAREA に渡されます。CICS COMMAREA は 32 KB に制限されているため、32 KB を超える COMMAREA が指定された EXEC CICS LINK は LENGERR 条件のために失敗します。

CNTCALL パラメータを指定すると、渡されるデータが COMMAREA の最大長である 32 KB を超えた場合に、EXEC CICS LINK で自動的にコンテナを使用できます。この機能は、チャンネルおよびコンテナをサポートする CICS バージョンでのみ動作します (CICS トランザクションサーバー 3.1)。

このとき、Natural CALL ステートメントの前にアプリケーションプログラミングインターフェイス USR4204N を介して明示的に指定されていなければ、デフォルトのコンテナ名は "NCI-COMMAREA" です。

可能な値は次のとおりです。

値	説明：
YES	COMMAREA データが 32 KB を超えると、Natural CICS インターフェイスは EXEC CICS LINK で自動的に CICS コンテナを使用します。このとき、デフォルト名として NCI-COMMAREA を使用します。
NO	COMMAREA データが 32 KB を超えると、Natural CALL ステートメントは失敗し、NAT0920 メッセージおよび理由コード LENGERR (16 進数で 16) が出力されます。

## COMACAL - サブルーチン呼び出しでの CICS COMMAREA の使用

このパラメータは、EXEC CICS LINKで外部サブルーチンプログラムを起動するときに、Natural CICS インターフェイスがCICS コマンドレベル COMMAREA 機能を利用するかどうかを定義します。

可能な値は次のとおりです。

値	説明：
YES	Natural パラメータリストのアドレスは、CICSTWA (Naturalバージョン1と同じ)、および CICS COMMAREA (Naturalバージョン2と同じ) に渡されます。COMMAREA の長さ設定は、FLDLEN パラメータの設定に従い 8 または 12 になります。YES を指定すると、特に CICS コマンドレベルで書かれた外部サブルーチンプログラムとの通信が向上します。これは、これらのサブルーチンにとって CICS TWA より CICS COMMAREA にアクセスする方が簡単だからです。また、オーバーヘッドも減少します。  これはデフォルト値です。
NO	Natural は、Natural 要求パラメータリストのアドレスを、CICS TWA の外部サブルーチンプログラムにのみ渡します。  COMMAREA の長さ設定は 0 です。

COMACAL パラメータは、呼び出された外部サブルーチンにデータを渡す方法に関して、Natural バージョン 1 に実質的な「互換モード」を提供することができます。

呼び出しオプションの SET CONTROL 'P=C' を利用する場合は、このパラメータを YES に設定します。『端末コマンド』ドキュメントの端末コマンド %P の説明も参照してください。

## COMAMSG - 終了メッセージでの CICS COMMAREA の使用

このパラメータは、CICS COMMAREA で、Natural 終了メッセージおよびリターンコードを潜在的なバックエンドプログラムまたはトランザクションに渡すかどうかを制御します。

可能な値は次のとおりです。

値	説明：
YES	Natural バックエンドパラメータエリアおよび潜在的な終了データが、CICS COMMAREA で渡されます。  これはデフォルト値です。
NO	この設定を行うと、Natural はセッションの正常な終了時または異常終了時に、Natural バックエンドパラメータエリア (Natural の『オペレーション』ドキュメントの「バックエンドプログラムの呼び出し規則」も参照) を CICS TWA の潜在的なバックエンドプログラムだけに渡します。  潜在的な終了データも、COMMAREA で渡されます。利用できる終了データがない場合は、COMMAREA は渡されません。

## COMARET - タスク制御での CICS COMMAREA の使用

このパラメータは、擬似会話型タスクを終了および再スタートするときに、Natural CICS インターフェイスが CICS コマンドレベル COMMAREA 機能を利用するかどうかを定義します。

可能な値は次のとおりです。

値	説明：
YES	擬似会話型の Natural タスクは、EXEC CICS LINK または同等の CICS のマクロ要求によって起動されている場合を除き、再スタート情報を CICS COMMAREA に保存します。  これはデフォルト値です。
NO	Natural は再スタート情報を CICS のメインの一時ストレージに配置します。この結果、この情報を配置および取得するために必要な CICS サービスコールが発生するため、オーバーヘッドが増加します。  使用される CICS 一時ストレージキーは、接頭辞の文字列 (NCMDIR パラメータ <b>TSKEY</b> で定義) と端末 ID から構成されます。CICSplex 環境で実行する場合、CICS 一時ストレージキーの接頭辞を CICS TST で、参加するすべての CICS リージョンでアクセス可能な REMOTE/SHARED として定義する必要があります。

COMARET パラメータは、擬似会話型の再スタートデータを置く場所という点で、実質的に Natural バージョン 1 との互換性を提供します。

## CONSOLE - オペレータコンソールの CICS 端末 ID

このパラメータは、オペレータコンソールとのメッセージ交換に Natural によって使用される端末 ID を指定します。

可能な値は次のとおりです。

値	説明：
XXXX	XXXX には、任意の 4 文字の端末 ID を指定できます。
CN01	これは、z/OS 環境でのデフォルト値です。
CNSL	これは、z/VSE 環境でのデフォルト値です。

コンソール端末は、端末エントリでトランザクションの自動開始 (ATI) を許可する必要があります。

## DIRNAME - Natural CICS インターフェイスのシステムディレクトリモジュールの名前

このパラメータは、Natural CICS インターフェイスのシステムディレクトリモジュールの名前を指定します。

可能な値は次のとおりです。

値	説明：
(下記を参照)	任意の有効なモジュール名
<i>prefix</i> CB	<i>prefix</i> は、プログラムおよびファイルに共通の接頭辞です。PREFIXパラメータを参照してください。  これはデフォルト値です。

ディレクトリモジュール名の最初の5文字も、対応するNCI環境と関連したCICS一時ストレージのキュー名の一部として使用されます。このため、CICSリージョンで複数のNatural CICS環境を実行する場合は、関連するシステムディレクトリモジュール名は最初の5文字が異っている必要があります。

指定した、またはデフォルトのNatural CICSインターフェイスのシステムディレクトリモジュール名は、NCIシステムディレクトリモジュール名の出口インターフェイスNCIDIREXを介してランタイムに変更できることに注意してください。これにより、同じNCIドライバ/NCIPARMを使用しながら、CICSシステムID、トランザクションIDなどに応じて異なるNCI環境（スレッドグループ/スレッドサイズなど）を使用することが可能になります。

## DUPTID - 重複する端末IDの処理

端末IDはセッション情報レコード（SIR）のキーなので、Natural CICS インターフェイスでは端末IDは一意である必要があります。これは通常、単一のCICSリージョンに対しては保証されていますが、同じSIPサーバーを共有する複数のCICSリージョンでは保証されていない場合があります。

DUPTIDパラメータは、端末IDが重複する場合、つまり、新しいセッションを開始するときに、この端末IDに対してすでにSIRが存在する場合、Natural CICS インターフェイスがどのように処理するかを決定します。

可能な値は次のとおりです。

値	説明：
YES	重複する端末 ID が見つかった場合、Natural CICS インターフェイスは、古いセッションを内部的に終了し、その後で新しいセッションを開始します。  これはデフォルト値です。
NO	新しいセッションの端末 ID に対して SIR がすでに存在する場合、Natural CICS インターフェイスは新しいセッションを終了し、メッセージ NS19 を発行します。説明および対処措置については、『メッセージおよびコード』ドキュメントの「CICS 環境下の Natural のメッセージ」、「SCP 処理のエラー」を参照してください。

端末 ID の出口インターフェイスでは、例えば 4 文字の CICS システム ID を物理的な 4 文字の CICS 端末 ID に追加することによって、一意な 8 文字の端末 ID を作成することができます。この結果、論理的な Natural 端末 ID が得られます。

### FLDLEN - 外部プログラム呼び出しにおけるフィールド長リストの指定

このパラメータは、EXEC CICS LINK を介して外部サブルーチンを起動するときに、フィールド長リストのアドレスを渡すかどうかを定義します。

可能な値は次のとおりです。

値	説明：
NO	FLDLEN=(NO,NO) と同じです。  COMACAL が NO に設定されている場合以外は、パラメータアドレスリストのアドレスおよびフィールド説明リストのアドレス (CALL ステートメントで説明されている R1 および R2) のみが、CICS TWA および CICS COMMAREA で渡されます。  これはデフォルト値です。
YES	FLDLEN=(YES,NO) と同じです。  フィールド長リストのアドレス (CALL ステートメントで説明されている R3) も、CICSTWA および COMMAREA でそれぞれ渡されます。
(YES,YES)	FLDLEN=(,YES) と同じです。  フィールド長リストのアドレスおよび大きいフィールド長リストのアドレス (CALL ステートメントで説明されている R4) も、CICS TWA および COMMAREA でそれぞれ渡されます。2つ目のサブパラメータを YES に設定すると、1つ目のサブパラメータに YES 条件が設定されます。

次の表に、COMACAL および FLDLEN で可能な組み合わせ、および関連する CALL パラメータ値を示します。

NCIPARM パラメータ		CALL パラメータ		
COMACAL	FLDLEN	TWA パラメータのアドレス	COMMAREA パラメータのアドレス	COMMAREA の長さ
NO	NO,NO	2	n/a	0
NO	YES,NO	3	n/a	0
NO	YES,YES	4	n/a	0
YES	NO,NO	2	2	8
YES	YES,NO	3	3	12
YES	YES,YES	4	4	16

いずれの場合でも、最後に渡されたアドレスには、リストの最後のアドレスであることを示すフラグが付加されます。このフラグはアドレスフィールドの上位ビットに設定されます。

### LOGDEST - Natural CICS のログの出力先

このパラメータは、Natural CICS インターフェイスがセッションログレコードを書き込む、CICS の出力先の名前を指定します。

可能な値は次のとおりです。

値	説明:
<i>name</i>	任意の有効な宛先名
NLOG	これはデフォルト値です。

CICS 宛先管理テーブルのエントリは、オプションの Natural CICS ログデータセットに対して定義する必要があります。

### MSGDEST - Natural エラーメッセージログの出力先 ID

このパラメータは、Natural CICS インターフェイスで NCI 情報メッセージに使用され、セッションが異常終了した場合に Natural セッション終了メッセージを記録する CICS の出力先の名前を指定します。

可能な値は次のとおりです。

値	説明：
<i>name</i>	任意の有効な宛先名
NERR	これはデフォルト値です。

これらのメッセージは文字フォーマットなので、新しく定義する代わりに、すでに利用できる CICS の出力先（例えば CSSL）を使用することもできます。

### MSGPFX - WTL メッセージに対する NCI メッセージの接頭辞の生成

Natural CICS インターフェイスは、MSGDEST 出力先に送信するすべてのメッセージに対して接頭辞を使用します。この接頭辞の長さは約 48 バイトで、次の情報を含んでいます。

- NCI メッセージ番号
- CICS リージョン SYSID
- 端末 ID または非端末タスクの文字列の ASYN
- ユーザー ID
- トランザクション ID
- 日時

デフォルトでは、メッセージの接頭辞は CMWTL を介して出力されるメッセージにも追加されません。

可能な値は次のとおりです。

値	説明：
YES	NCI メッセージの接頭辞が、CMWTL を介して発行されるすべてのメッセージに追加されます。 これはデフォルト値です。
NO	NCI メッセージの接頭辞が、CMWTL を介して発行されるメッセージに追加されません。メッセージは変更されることなく発行されます。



## MSGTRAN - 内部メッセージ交換トランザクション ID

このパラメータは、Naturalのメッセージ交換および非同期セッションフラッシュ機能によって内部的に使用されるトランザクション ID を指定します。

可能な値は次のとおりです。

値	説明：
(下記を参照)	任意の有効な CICS トランザクション ID。
NMSG	これはデフォルト値です。

このトランザクション ID は、Natural を起動するために使用されるトランザクション ID とは異なっている必要があります。また、CICS で定義されている必要があります。

## PREFIX - プログラムおよびファイルの共通の接頭辞

このパラメータは、Natural CICS コンポーネントで共通のモジュールの接頭辞を、Natural CICS システムディレクトリ *prefixCB*、CICS 3270 Bridge XFAINTU 出口 *prefixXFA*、VSAM ロールファイル *prefixRn* ( $n=1\sim 9$ ) のように定義し、また、CICS のメイン一時ストレージ内の、NCI によるすべての永久的な GETMAIN ストレージに関する情報を保持するシステム制御レコードを、ローカルプールおよび共有スレッドとして定義します。TS 制御レコードキーは、*prefixXCR* の形式で、*x* は出力不能文字です。この共通の接頭辞をすべての NCI 関連のプログラムに対して使用するのはいくつかの慣習です。例えば、Natural CICS インターフェイスニュークリアスの場合は *prefixDRV*、Natural CICS インターフェイスノードエラープログラムの場合は *prefixNEP* を使用します。

可能な値は次のとおりです。

値	説明：
XXXXX	<i>prefix</i> は長さ 1~5 バイトで、プログラムおよびファイルの命名規則に一致する必要があります。

デフォルト値はありません。

## PRMDEST - Natural CICS プロファイルパラメータの入力先の名前

このパラメータは、Natural ダイナミックプロファイルパラメータを含む CICS の宛先の名前を指定します。

可能な値は次のとおりです。

値	説明：
<i>name</i>	任意の有効な宛先名
NPRM	これはデフォルト値です。

システムの初期化時に、Natural CICS インターフェイスは、Natural ダイナミックプロファイルパラメータを取得し、その環境に保存します。セッションの開始時に、潜在的な他のプロファイルパラメータ（端末入力経由またはフロントエンドの呼び出し元によって入力）は、PRMDEST の宛先から取得されたパラメータ文字列の末尾に連結されます。つまり、明示的なダイナミックプロファイルパラメータを、PRMDEST から読み込んだ Natural CICS インターフェイスシステムのプロファイルパラメータを上書きするために使用することができます。

CICS の宛先管理テーブルのエントリは、Natural CICS インターフェイスのオプションのプロファイルパラメータ入力先に対して定義する必要があります。通常これは追加パーティションの宛先です。

### PSTRNID - \*INIT-PROGRAM 変数設定の制御

Natural タスクがフロントエンドプログラムによってアクティブにされると、PSTRNID パラメータは Natural システム変数 \*INIT-PROGRAM の設定方法を決定します。

可能な値は次のとおりです。

値	説明：
YES	*INIT-PROGRAM は、Natural CICS 擬似会話型タスク処理に使用される実際のトランザクション ID に設定されます。これは、最初に Natural セッションを開始したタスクのトランザクション ID である必要はありません。  これはデフォルト値です。
NO	*INIT-PROGRAM は、最初に Natural セッションを開始したタスクのトランザクション ID に設定されます。

### RCVASYN - 非同期セッションの回復

このパラメータは、Natural CICS インターフェイスによる非同期セッションの処理方法を定義します。

可能な値は次のとおりです。

値	説明：
YES	<p>これはデフォルト値です。</p> <p>Natural CICS インターフェイスは、端末以外のセッションの一部の Natural プロファイルパラメータ設定が、予期しない入力または NT06、NT11 などの I/O エラーによる異常終了を防ぐように設定します。</p> <p>RCVASYN=YES は、以下のパラメータ設定を強制します。</p> <ul style="list-style-type: none"> <li>■ CM=OFF, MENU=OFF, PC=OFF</li> <li>■ TTYPE=ASYL (SENDER 指定が空白、指定されていない、または NCIPARM MSGDEST 指定と同じ場合)</li> <li>■ SENDER='msgdest' (SENDER 指定が空白または指定されていない場合)</li> <li>■ OUTDEST='sender' (OUTDEST 指定が空白または指定されていない場合)</li> </ul>
NO	<p>Natural CICS インターフェイスは、端末以外のセッションに対して何も行いません。ユーザーは、非同期 Natural セッションに対して適切な Natural プロファイルパラメータを設定する必要があります。「Natural の非同期処理」を参照してください。</p>

### RESENDC - サブルーチン呼び出し後の画面再送信チェック

Natural はデフォルトで 3270 の出力データストリームを最適化します。Natural で使用される画面イメージ技術により、Natural は最後に送信したマップを記憶することができます。このため新しいマップを送信するときは、実際には古いマップの「更新部分」のみが送信されます。このロジックでは、画面イメージは、Natural によって呼び出された画面 I/O を実行する 3GL プログラムによって破壊される場合があります。

可能な値は次のとおりです。

値	説明：
YES	<p>Natural CICS インターフェイスは、呼び出された 3GL プログラムが画面 I/O を実行したかどうかをチェックします。実行されている場合は、次の画面 I/O でフルスクリーンが送信されます。</p> <p>これはデフォルト値です。</p>
NO	<p>Natural CICS インターフェイスにより、更新部分のみが送信されます。</p>

## RESENDS - 擬似会話型セッション再開後の画面再送信チェック

Natural はデフォルトで 3270 の出力データストリームを最適化します。Natural で使用される画面イメージ技術により、Natural は最後に送信したマップを記憶することができます。このため新しいマップを送信するときも、「更新部分」のみが送信されます。このロジックでは、画面イメージは、例えば擬似会話型画面 I/O 時のメッセージ交換（CICS CMSG トランザクション）によって破壊される場合があります。

可能な値は次のとおりです。

値	説明：
YES	Natural セッション中に、Natural CICS インターフェイスは外部からの画面 I/O も認識し、最後に発行された画面を再送信します。  これはデフォルト値です。
NO	新しいマップを送信するときは、「更新部分」のみを送信します。


## RJEDEST - Natural CICS サブミット先の名前

このパラメータは、z/OS タイプのオペレーティングシステムにのみ適用されます。

可能な値は次のとおりです。

値	説明：
(下記を参照)	宛先の名前。
NRJE	これはデフォルト値です。

RJEDEST は、JES 内部リーダー機能を介してジョブをサブミットするために NATRJE ユーティリティによって使用される CICS の追加パーティションの宛先の名前を指定します。

-  **注意:** 適切な CICS の宛先を CICS DCT および起動 JCL で定義する必要があります。Natural の『インストール』ドキュメントの「Natural CICS インターフェイスのインストール手順」で対応する手順も参照してください。  
ファンクションコード L または B (NATRJE CALL ステートメントの *parm3*) を最後の NATRJE 呼び出しに対して設定する必要があります。  
L が指定されて、*nrje* が追加パーティションの宛先である場合は、この宛先は閉じられ、次に内部リーダーが起動されます。  
B が指定されて、*nrje* が間接の宛先である場合は、この宛先は閉じられません。この場合、内部リーダーを起動するために、後続の /\*EOF カードをサブミットする必要があります。

Natural NATRJE ユーティリティの詳細については、Natural の『ユーティリティ』ドキュメントを参照してください。

## RJEUSER - POWER へのサブミットを行うユーザー ID の設定

このパラメータは、POWER スプーリングシステムを使用する z/VSE オペレーティングシステムにのみ適用されます。

可能な値は次のとおりです。

値	説明：
YES	RJEUSER=(YES,CICS) と同じです。  これはデフォルト値です。  Natural システム変数 *INIT-USER が XPCC ユーザー ID として使用されます。また、POWER JECL をユーザーが適切に設定する必要があります。
(YES,NAT)	Natural システム変数 *USER が XPCC ユーザー ID として使用されます。また、POWER JECL をユーザーが適切に設定する必要があります。
NO	ユーザー ID 'R000' が XPCC ユーザー ID として、Natural CICS インターフェイスによってサブミットされたすべてのジョブに使用されます。

z/VSE オペレーティングシステムでは、CICS 環境の Natural は XPCC マクロ要求によってジョブのサブミットを行います。

XPCC マクロではユーザー ID を指定する必要があります。つまり、適切な LDEST/PDEST パラメータが \* \$\$ JOB ステートメントで指定されている場合、または適切な DEST パラメータが \*\$\$ LST または \*\$\$ PUN ステートメントで指定されている場合を除き、サブミットされたジョブのリストまたはパンチ出力へのアクセス権は、サブミットしたユーザーにのみ与えられます。

しかし、特殊なユーザー ID 'R000' を使用すると、JECL で適切なターゲットの宛先をコード化することなく、サブミットされたジョブのリストまたはパンチ出力に誰でもアクセスすることができます。

## SIGNON - サインオンの動作

このパラメータは、CICS 環境の Natural が Natural セッションにおいて CICS ユーザー ID を処理する方法を定義します。

可能な値は次のとおりです。

値	説明：
NO	これはデフォルト値です。 EXEC CICS ASSIGN USERID (...) によって返された空白以外の値はすべて有効と見なされます。
YES	EXEC CICS ASSIGN USERID (...) によって返された空白以外の値が、CICS デフォルトユーザー ID とは異なっていて、ユーザーが CICS にサインオンしている場合は（端末内のセッションのみ）、有効と見なされます。

その他の処理：

CICS ユーザー ID が無効な場合、

- 編集された（アンパック形式の）CICS タスク番号が端末以外の CICS セッション（つまり、非同期や DPL など）に対して取得されます。
- 端末内のセッションに対しては、3 バイトの CICS オペレータ ID が空白以外の場合はこの ID が使用されます。空白の場合は、CICS 端末 ID が使用されます。



#### 注意:

1. CICS 端末 ID は CICS リージョン内で一意ですが、CICS ユーザー ID およびオペレータ ID は一意でない場合もあります。しかし、CICS 端末 ID は他の CICS リージョンで重複している可能性があり、この場合、Adabas 内でユーザー ID が重複します。
2. Natural ユーザー ID 出口 NATUEX1 または Natural CICS ユーザー ID 出口インターフェイス `NCIUIDEX` は、\*INIT-USER をカスタマイズするために使用できます。

## SLCALL - 標準的なリンク呼び出し

Natural CALL ステートメントは、CICS 規則を使用して、つまり EXEC CICS LINK を介して Natural 以外のダイナミックなプログラムを起動します。Natural 以外のダイナミックなプログラムは、CALL ステートメントが実行される前に適切なインジケータが Natural プログラムで設定されていれば、標準のリンケージ規約（BALR/BASR/BASSM 14、15 など）でも起動することができます。端末コマンド %P=S、%P=SC、%P=L、および %P=LS も参照してください。



**注意:** 端末コマンド %P=S、%P=SC、%P=L、および %P=LS は、SLCALL が特定のリンケージ規約を自動的に使用する動作をバイパスします。

SLCALL により、特定のリンケージ規約を自動的に使用することができます。このことは、CICS マクロレベルの API がサポートされていない CICS システムに特に関係しています。CICS バージョン 3.2 以上およびすべての CICS/TS バージョンの場合がこれに該当します。

可能な値は次のとおりです。

値	説明：
YES	Natural CICS インターフェイスは、モジュールのロードポイントで文字列 DFH を探すことによって、呼び出されるモジュールが有効な CICS コマンドレベルプログラムかどうかを判断します。DFH が見つければ、プログラムは EXEC CICS LINK を介して起動されます。DFH が見つからなければ、モジュールは標準のリンケージ規約に従って処理され、BALR/BASSM 14、15 を介して起動されます。
NO	リンケージ規約は使用されません。これはデフォルト値です。

### SLNOHLD - 標準のリンケージ規約を介して起動される外部プログラムのロードオプション

このパラメータは、Natural CICS インターフェイスによる、非 CICSplex 環境で標準リンケージ規約によって起動される非 LE 外部プログラム（つまり、ダイナミックな非 CICS プログラムおよび RCA プログラム）の処理方法を定義します。

可能な値は次のとおりです。

値	説明：
YES	これはデフォルト値です。  Natural CICS インターフェイスによって、標準リンケージ規約によって起動されるすべての非 LE 外部プログラム（RCA プログラムを含む）が EXEC CICS LOAD を介して HOLD オプションなしでロードされます。したがって、擬似会話型画面 I/O での Natural セッションの中断/待機中にこれらのプログラムの NEWCOPY が許可されます。  SLNOHLD=YES は、Natural CICS インターフェイスによって LE プログラム全般および CICSplex 環境の非 LE プログラムに対して行われる処理に該当します。
NO	これは Natural の従来からの動作です。  Natural CICS インターフェイスによって、標準リンケージ規約によって起動されるすべての非 LE 外部プログラム（RCA プログラムを含む）が EXEC CICS LOAD HOLD を介してロードされます。つまり、そのようなプログラムは、DELETE プロファイルパラメータ設定に応じて、しばらくの間ストレージ内に固定されます。RCA プログラムはセッション終了までです。

## SNDLAST - EXEC CICS SEND コマンドの LAST オプションの使用

このパラメータは、ブラケットプロトコルを使う SNA 端末（LUTYPE2）で、擬似会話型画面 I/O に対して「終了ブラケット」を強制するのに役立ちます。

可能な値は次のとおりです。

値	説明：
YES	LAST オプションは、タスクが擬似会話型モードで終了する前に EXEC CICS SEND コマンドに使用されます。  これはデフォルト値です。
NO	LAST オプションは使用されません。

## TERMVAR - Natural ワークファイルの端末 ID 変数


このパラメータを指定すると、Natural ユーザーは、端末 ID を知らなくても CICS 環境で排他的な Natural ワークファイルを持つことができます。

可能な値は次のとおりです。

値	説明：
xxxx	変数 xxxx は 4 文字の文字列です。以下の説明を参照してください。
&TID	これはデフォルト値です。

端末 ID は CICS システムにおいて一意であるため、CICS 一時ストレージの排他的なワークファイルは通常、CICS 端末 ID を含んでいます。パラメータ TERMVAR を指定すると、変数を定義することができます。この変数がワークファイル名に見つかり、実際の端末 ID に置き換えられます。非英数字を含む文字列は、アポストロフィ（'）で囲む必要があります。

端末以外のセッションに対しては、パックされた CICS タスク番号が論理的な端末 ID として使用されることに注意してください。

 **注意:** 変数の文字列にサブストリング '\*\*' を含めることはできません。このサブストリングは Natural によってワークファイル番号に置き換えられるため、端末 ID を挿入できなくなります。



## TRANCHK - トランザクション ID に対する入力マップのチェック

セッションを終了する前に、CICS セッションへの接続が失われるか切断された場合は（VM 環境下またはセッションマネージャのインストール時など）、CICS を呼び出すときに、別のユーザーがこのオープンセッションに入る可能性があります。通常、CICS 環境でのユーザーの最初のアクションはトランザクション ID を入力することです。

可能な値は次のとおりです。

値	説明：
YES	Natural CICS インターフェイスは、ユーザーによって入力されたトランザクション ID の最初の 4 バイトが Natural トランザクション ID と一致するかどうかをチェックします。一致する場合は、接続が失われるか切断された後の「再スタート」であると認識されます。「古い」セッションのリソースはすべて解放され、新しいセッションが開始されます。
NO	ユーザーによって入力されたデータは、Natural トランザクション ID と比較してチェックされません。  これはデフォルト値です。

## UCTRAN - Natural での小文字／大文字小文字の混在のサポート

このパラメータは、Natural CICS インターフェイス (NCI) による小文字／大文字小文字の混在のサポートを有効または無効にします。

可能な値は次のとおりです。

値	説明：
YES	UCTRAN=(YES,YES) と同じです。NCI の小文字／大文字小文字の混在のサポートは完全に有効化されます。  これはデフォルト値です。
NO	UCTRAN=(NO,YES) と同じです。NCI の小文字／大文字小文字の混在のサポートは、擬似会話型画面 I/O に対して無効になります。


1 つ目のサブパラメータは擬似会話型の画面 I/O 後の NCI の大文字小文字の混在のサポートを制御し、2 つ目のサブパラメータは会話型の画面 I/O 後の NCI の大文字小文字の混在のサポートを制御します。後者には NTC のアップロードも含まれます。

### 1 つ目のサブパラメータ（擬似会話型画面 I/O）

擬似会話型 Natural セッションでの小文字／大文字小文字の混在のサポートを有効にするには、Natural ニュークリアスが制御を得る前に、端末入力が大文字に変換されていない必要があります。このため、Natural CICS インターフェイスはデフォルトで、UCTRAN=YES で定義された端末を、Natural セッションの存続期間は混在モード (UCTRAN(TRANID)) に切り替えます。

セキュリティの理由から CICS 定義／コントロールブロックの変更が望ましくない場合には、この NCIPARM パラメータ UCTRAN を NO に設定することにより、Natural CICS インターフェイスで端末の大文字変換ステータスが変更されないようにすることができます。その場合には、ユーザーは端末を「小文字」（CICS TYPETERM パラメータ UCTRAN(TRANID/NO)）で実行するように定義して、Natural の小文字／大文字小文字の混在のサポートを使用できるようにする必要があります。

現在の Natural バージョンでサポートされているすべての CICS バージョンで、トランザクションの PROFILE での UCTRAN パラメータを介したトランザクションレベルでの「大文字小文字の変換」が可能のため、この NCIPARM パラメータは NO に設定して、小文字／大文字小文字の混在のサポートは CICS で行われるようにします。

 **注意:** CICS では、TYPETERM および PROFILE 定義での UCTRAN パラメータの組み合わせにより、CICS での擬似会話型トランザクションの端末入力の処理方法を決定します（詳細については、『CICS Resource Definition Manual』その他を参照）。このため、必要な大文字変換ステータスをトランザクションに関連する PROFILE で主に定義して、アプリケーションが TYPETERM 大文字変換モードの変更に影響されないようにすることをお勧めします。

## 2 つ目のサブパラメータ（会話型の画面 I/O）

会話型 I/O で小文字／大文字小文字の混在がサポートされるということは、Natural CICS インターフェイスが CICS 端末入力要求で「as is」オプションを使用することを意味します（CONVERSE/RECEIVE ASIS）。2 つ目のサブパラメータが NO に設定されると、Natural CICS インターフェイスは「as is」オプションなしで会話型の CICS 端末入力要求を行います。

## UNITID - 一意の端末 ID の設定

このパラメータは、複数の CICS リージョンにおいて一意な Natural の CICS 端末 ID を作成するのに役立ちます。

可能な値は次のとおりです。

値	説明：
YES	Natural CICS インターフェイスは、CICS システム ID（MRO がいない場合はローカル SYSID、それ以外の場合は TOR SYSID）を 4 バイトの CICS 端末 ID に追加して、8 バイトの論理端末 ID を作成します。
NO	Natural CICS インターフェイスは CICS 端末 ID を変更せずに使用します。 これはデフォルト値です。

このパラメータは、特に非 CICSplex において、リソースが複数の CICS リージョンによって SIP サーバーまたはロールサーバーとして共有される場合に重要です。同じ端末 ID が複数の CICS 環境で使用される場合は、このパラメータは Natural に一意の端末 ID を提供するのに役立ちます。Natural CICS インターフェイス内部では、Natural 端末 ID は 8 バイトのフィールドです。

8 バイトの端末 ID および 8 バイトの CICS ユーザー ID の組み合わせは、SIP およびロールサーバーのキーとして使用されます。

このパラメータの結果は、Natural CICS インターフェイスによってセッションキーおよびロールサーバーキーに、および Natural によってシステム変数 \*INIT-ID に使用されます。

**注意:**

1. 端末 ID 出口 (NCITIDEX) は、その論理端末 ID を後処理する場合があります。
2. また、ユーザー ID 出口 (NCIUIDEX および NATUEX1) は、\*INIT-ID システム変数を後処理する場合があります。
3. このパラメータは、*Natural Advanced Facilities* (NAF) プリンタにも適用されます。つまり、NAF スプーリングおよびレポート管理システム NATSPOOL でプリンタを適切に定義するか、またはユーザー ID 出口をプリンタの \*INIT-ID の後処理に使用する必要があります。
4. 非端末のセッションでは、Natural CICS インターフェイスは常に、パックされた CICS タスク番号および CICS システム ID から構成される 8 バイトの論理端末 ID を設定します。つまり、UNITID=YES が、端末 ID として使用された CICS タスク番号を持つ非同期タスクに対して設定されます。



## 6 VSAM RRDS ロールファイルのカスタマイズ

---

- VSAM RRDS ロールファイル数の増加 ..... 60
- VSAM RRDS ロールファイル数の減少 ..... 61
- VSAM RRDS ロールファイルの特性の変更 ..... 61

Natural CICS インターフェイスドキュメントのこの部分では、VSAM RRDS ロールファイルのカスタマイズについて説明します。

- **VSAM RRDS** ロールファイル数の増加
- **VSAM RRDS** ロールファイル数の減少
- **VSAM RRDS** ロールファイルの特性の変更

Natural ロールサーバーを使用している場合は、このセクションの内容は適用されません。

### CICS テーブルの参照

適切な場合は、CICS テーブル（DCT、FCT、PCT、PPT、TCT など）への参照を、以下に示す対応する項目への参照と見なすことができます。

- アセンブリタイプのリソース定義
- CEDA を介したオンラインリソース定義
- DFHCSDUP を介したバッチリソース定義

この章では、次のトピックについて説明します。

## VSAM RRDS ロールファイル数の増加

---

VSAM RRDS ロールファイルは9つまで割り当てることができます。各ロールファイルには、ユーザー定義の接頭辞と、後に続く固定された接尾辞から構成される ID が付けられています。接頭辞の長さは1~9文字です。接尾辞は R1 から R9 までの2文字から構成されます。

新しい VSAM ロールファイルを追加するには、次の手順に従います。

1. ローカルサイトの標準に準拠する空の VSAM RRDS を作成します。次にバッチプログラム NCISCPRI を使用してデータセットを初期化します。このバッチプログラムは、Natural のインストール時にアセンブルしておく必要があります。SPACE および RECORDSIZE 属性はロールファイルごとに異なる場合があります。このため、環境に最適な値を見つけるために必要に応じて変更することができます。
2. FCT エントリを作成し、CICSJCL を適切に変更します。このとき、これらに対して接頭辞/接尾辞を使用します。

Natural CICS インターフェイスを再度初期化すると、新しいロールファイルを利用できるようになります。

## VSAM RRDS ロールファイル数の減少

---

次の手順に従います。

1. Natural が稼動していないことを確認します。
2. FCT および JCL の定義を削除するか、またはファイルを削除します。

Natural CICS インターフェイスを再度初期化すると、ロールファイル数が調整されます。

## VSAM RRDS ロールファイルの特性の変更

---

次の手順に従います。

1. ロールファイルの数を減らすには、上記の手順に従います。
2. ロールファイルの数を増やすには、該当する手順に従います。





# 7 CICS MRO 環境での Natural

---

- YES に設定した NCIPARM パラメータ COMARET ..... 64
- NO に設定した NCIPARM パラメータ COMARET ..... 64

Natural CICS インターフェイスドキュメントのこの部分では、CICS マルチプルリージョン (MRO) 環境における Natural の機能について説明します。

### CICS テーブルの参照

適切な場合は、CICS テーブル (DCT、FCT、PCT、PPT、TCT など) への参照を、以下に示す対応する項目への参照と見なすことができます。

- アセンブリタイプのリソース定義
- CEDA を介したオンラインリソース定義
- DFHCSDUP を介したバッチリソース定義

CICS マルチプルリージョン (MRO) 環境で Natural を実行する場合には、特別な考慮事項があります。

この章では、次のトピックについて説明します。

## YES に設定した NCIPARM パラメータ COMARET

---

NCIPARM パラメータ **COMARET** を YES に設定すると、Natural セッションデータは、2つの異なる CICS リージョンに保存されます。

- セッション再スタート情報は、CICS 端末所有領域 (TOR) の端末エントリにリンクされた COMMAREA に保存されます。
- 実際のセッションデータは CICS アプリケーション所有領域 (AOR)、つまり、スレッド、スワッププール、またはロール機能に保存されます。

これにより、例えば AOR を再スタートすると不整合が生じる場合がありますが、TOR には古い「保留中」の Natural セッションが含まれているため、このようなセッションを再開すると、対応するエラーメッセージが表示されます。

## NO に設定した NCIPARM パラメータ COMARET

---

**COMARET** を NO に設定すると、すべての Natural セッションデータは AOR に保存され、上記の不整合を防ぐことができます。

しかし、端末を TOR から削除すると (VTAM に戻る、セッションマネージャを切り替える、または電源をオフにする) セキュリティの問題が生じる場合があります。また、この TOR にダイヤルする別の端末は、削除された端末の ID を受信し、Natural トランザクションコードを入力します。次にこの端末は、先に削除された端末のセッションを再開しますが、これは AOR の一時ストレージに再スタート情報が含まれているからで、ここにはキュー名の一部として端末 ID が含まれています。

このような状況を防ぐために、ノードエラープログラム（NEP）をインストールすることができます（「[Natural に対する CICS ノードエラープログラムに関する考慮事項](#)」および「[Natural CICS のサンプルプログラム](#)」を参照）。これにより、関連する端末が削除されたときに Natural セッションが終了されます。



# 8 Natural に対する CICS ノードエラープログラムに関する考慮事項

---

■ 通常の状態 .....	68
■ Natural CICS インターフェイス管理外の状態 .....	68
■ リカバリメカニズム .....	69
■ 特別な考慮事項 .....	69
■ ダミープログラムの例 .....	70

この章では CICS ノードエラープログラムに関する考慮事項について説明します。

以下の項目も参照してください。

- CICS ノードエラープログラムのインストール。CICS ノードエラープログラムのインストールの詳細については、対応する CICS バージョンの『*CICS Customization Guide*』で、ユーザーが交換可能なプログラムに関するセクションを参照してください。

## 通常の状態

---

Natural セッションがアクティブな場合は常に、スレッドストレージ、ロール機能エントリ（つまり VSAM RRDS ファイルまたは CICS 一時ストレージキュー内のレコード）、スワッププールスロットなどの CICS リソースが使用されます。

これらのリソースが Natural CICS インターフェイスに制御されている場合は、セッションが正常終了しても異常終了しても、リソースは正常に解放されます。

## Natural CICS インターフェイス管理外の状態

---

次の場合は、Natural CICS インターフェイスによって制御することはできません。

1. Natural によって呼び出された Natural 以外のプログラムが、EXEC CICS ABEND CANCEL コマンドまたは同等の CICS マクロ要求を発行します。すべてのセッションリソースを適切に解放するため、CICS タスクは、Natural CICS インターフェイスが制御を受け取ることなくキャンセルされます。
2. 一部の CICS モニタ製品には CICS タスクを削除するツールがあり、アプリケーションによって設定された異常終了出口をバイパスします。Natural タスクがこのようにキャンセルされた場合、Natural CICS インターフェイスは、まだセッションに所有されているリソースを解放することができません。
3. Natural セッションが CICS においてアクティブでないときに（擬似会話型画面 I/O）、ユーザーが、電源をオフにするか、セッションマネージャの該当する機能を使用して CICS リージョンから端末を切断します。

## リカバリメカニズム

Natural CICS インターフェイスには、このような状況から回復するためのリカバリメカニズムがあります。以下に例を示します。

新しい Natural セッションを開始する場合は常にテーブルをスキャンして、同じ端末 ID を持つ別の Natural セッションがすでにアクティブになっていないかを確認します。このようなセッションが存在する場合は論理的に終了し、そのすべてのリソースを、新しいセッションを開始する前に解放します。

しかし、セッションを論理的に終了してからリソースを解放するまでに長い時間がかかる場合があります。また、セキュリティの問題がある場合もあります。

NCIPARM 生成パラメータ **COMARET** を NO に設定すると、Natural セッションを再開する情報は CICS 一時ストレージレコードに保存され、このレコードの端末 ID は一時ストレージキュー名の一部になります。別の CICS ユーザーがこの端末 ID で Natural を開始しようとする、新しいセッションは開始されずに古い Natural セッションが再開されます。

上記のリストで3つ目のケースが最も重大です。CICSのノードエラープログラム (NEP) 出口インターフェイスを使用すると、このような場合に、失われたセッションを Natural CICS インターフェイスで終了することができます。NCIZNEP という適切なプログラムが、Natural CICS ソースライブラリで提供されています（「[Natural CICS のサンプルプログラム](#)」を参照）。このプログラムは DFHZNEP ノードエラープログラムによって呼び出す必要があります。

## 特別な考慮事項

この他にもいくつかの考慮事項があります。

- Natural CICS ソースライブラリには、CICS/VSE バージョン 2.3 用の XNCINEP1 および CICS/TS 用の XNCINEP2 という2つのサンプルのノードエラープログラムが含まれています。どちらのサンプルプログラムも Natural CICS 環境に対して特別な処理を実行することはなく、単に (LINK を介して) NCIZNEP プログラムを呼び出し、そのプログラムが CICS 環境の Natural を処理します。
- DFHZNEP は、特定のインストールに対してすでにカスタマイズされている場合があります。ノードエラープログラムは1つに制限されているため、関連する XNCINEP<sub>x</sub> プログラムのロジックを既存の DFHZNEP ロジックに適合させる必要があります。
- MRO 環境では、DFHZNEP は TOR にインストールする必要があります。
- CICS 3.3 以上では、NCIZNEP プログラムを EXECCKEY(CICS) で定義する必要があります。
- 上記の項目 3 で説明した場合において、DFHZNEP はさまざまな内部エラーコードによって、複数回制御を受け取る場合があります。これは、各内部エラーコードは特定の CICS エラー

メッセージと関連していますが、ある特定のアクションから複数のエラーメッセージが生じる場合があるためです。

- 完了時に、NCIZNEP はパラメータ入力をクリアすることによって、作業、つまり Natural セッションクリーンアップタスクの起動が正常に完了したかどうかを呼び出し元（通常は DFHZNEP）に示します。
- ノードエラープログラムが起動されるたびに、CICS コントロールブロック群は変更されている可能性があります。例えば、TCTTE の COMMAREA および NEXTRANSID 情報は、特定のノードエラーイベントの後に失われている場合があります。この場合、NCIPARM パラメータ COMARET を適切に設定する必要があります。これは、CICS によってすでにクリーンアップされた CICS COMMAREA の Natural 擬似会話型セッション再スタートデータを渡すときに起動されるノードエラープログラムのノードエラーイベントを選択できないことを意味します。

## ダミープログラムの例

---

特定のアクションで起動される DFHZNEP の回数とエラーコード、および TCTTE の状態を調べるには、ダミーのノードエラープログラムを記述します。このプログラムは、要求された情報を示す CICS トレース要求のみを発行します。

次のサンプルでは、DFHZNEP エラープロセッサは、DFHZNEP に渡される可能なすべてのエラーコードに対して、制御を受け取ることができます。

```
.  
  
DFHSNEP TYPE=INITIAL  
  
ORG NEPTT  
  
DC 256X'03' invoke error processor '3' for ALL error codes  
  
ORG ,  
  
DFHSNEP TYPE=ERRPROC, GROUP=3, CODE=49  
.  
  
set up requested information and issue trace request(s)
```



## 9 CICS 3270 Bridge のサポート

---

- CICS 3270 Bridge のデフォルトサポート ..... 72
- CICS 3270 Bridge の完全なサポート ..... 72
- NCIXFATU - NCI ソースモジュール ..... 72
- プロファイルパラメータ DSC=OFF の推奨 ..... 72

Natural CICS インターフェイスドキュメントのこの章では、CICS 3270 Bridge のサポートについて説明します。

### CICS 3270 Bridge のデフォルトサポート

---

デフォルトでは、Natural CICS インターフェイスは、「ブリッジされた装置」、つまり CICS 3270 Bridge 出口経由でエミュレートされる端末を処理する機能により CICS 3270 Bridge をサポートしています。

### CICS 3270 Bridge の完全なサポート

---

CICS 3270 Bridge の完全なサポートが必要な場合は、NCI ソースモジュール NCIXFATU をインストールする必要があります。『インストール』ドキュメントの対応する手順「Natural/CICS インターフェイス XFAINTU 出口のアセンブル」を参照してください。

### NCIXFATU - NCI ソースモジュール

---

NCIXFATU モジュールは、実際には CICS XFAINTU グローバルユーザー出口 (GLUE) です。その目的は、ブリッジ機能の保持期間 (keep-time) の期限が切れ、関連する Natural セッションがまだ終了していない場合に、Natural リソースを解放することです。

NCIXFATU モジュールは、ノードエラープログラム (NEP) が「実際の」端末に提供するのと同じ機能を Natural に提供します。

### プロファイルパラメータ DSC=OFF の推奨

---

CICS 3270 Bridge を使用している場合は、プロファイルパラメータ DSC=OFF (3270 タイプ端末のデータストリーム圧縮が無効) で Natural セッションを開始して、Natural で以前の画面との差分ではなく常にフルスクリーンが送信されるようにすることをお勧めします。

# 10 Natural CICS インターフェイスのスレッドセーフに関する考慮事項

---

Natural CICS インターフェイスはスレッドセーフに定義して、スループットを向上させるために OTE TCB を利用することができます。

つまり、QR TCB で実行していない場合は、Natural CICS インターフェイスは CICS ENQ/DEQ を介して追加のシリアライゼーションを提供する必要があります。

これらのシリアライゼーションの影響を最小限にするために、

- **THREADS** パラメータを指定せずに（または **THREADS=0**）、**TYPE=GETM** スレッドを使用して、CICS のロール機能ではなく
- *Natural* ロールサーバーを使用することをお勧めします。



## 注意:

1. CICS の Adabas もスレッドセーフである必要があります。つまり、この機能をバージョン 8 より前の Adabas バージョンとともに使用することはできません。
2. **CSTATIC** として定義されるすべてのユーザープログラムは、スレッドセーフである必要があります。
3. 標準のリンケージ規約を介して明示的に（つまり、端末コマンド **%P=S**、**%P=SC**、**\*P=L**、または **%P=LS** を使用）または暗黙的に（つまり、**NCIPARM** 生成パラメータ **SLCALL** が **YES** に設定されている場合）起動されるすべてのダイナミックユーザープログラムは、スレッドセーフである必要があります。



# 11 Natural CICS インターフェイスでの CICS チャネル およびコンテナのサポート

---

CICS トランザクションサーバーのバージョン 3.1 では、IBM は、EXEC CICS LINK 用のチャネルおよびコンテナを導入しました。この点で、CICS コンテナは 32 KB 制限のない名前付きの COMMAREA と考えることができます。

Natural CICS インターフェイスは、次の 2 つの方法で CICS コンテナをサポートします。

1. 特別な SET CONTROL 'P=CC' を介して、CALL ステートメントパラメータデータがコンテナに渡されます。
2. NCIPARM システム生成パラメータ **CNTCALL** を YES に設定すると、CALL ステートメントを使って渡されたパラメータデータが 32 KB を超えている場合に、%P=C CALL は CICS COMMAREA の代わりに CICS コンテナを自動的に使用します。

いずれの場合も、コンテナ名が「実際の」CALL ステートメントに先立ってアプリケーションプログラミングインターフェイス USR4204N を介して明示的に定義されていない場合は、デフォルトのコンテナ名は NCI-COMMAREA です。



# 12 Natural CICS インターフェイスおよび IBM 言語環境 (LE)

---

- z/OS 用 CICS トランザクションサーバーのバージョン 3.1 以上 ..... 78
- CICS/TS 3.1 より前のバージョンの CICS トランザクションサーバー ..... 78

Natural CICS インターフェイスは LE プログラムをサポートします。このドキュメントでは、CICS 環境の Natural での LE の有効化について説明します。

## z/OS 用 CICS トランザクションサーバーのバージョン 3.1 以上

---

z/OS 用 CICS トランザクションサーバーのバージョン 3.1 以上では、Natural CICS インターフェイスは単独で LE に対応しています。つまり、次の場合に、Natural CICS インターフェイスタスクは、C、COBOL、PL/I などの言語で記述された LE プログラムを直接 CALL (CICS LINK ではなく標準のリンケージ規約) することができます。

- SET CONTROL 'P=LS' が指定された場合
- SET CONTROL 'P=S' が指定された場合
- NCIPARM パラメータ `SLCALL=YES` が指定され、呼び出されるプログラムが CICS プログラムではない場合

CICS/TS 3.1 で Natural CICS インターフェイスを LE 対応にするには、Natural CICS インターフェイスのスタートモジュール NCISTART および Natural CICS インターフェイスの RPC サーバフロントエンドモジュール NCISFED をインストールするときに CICS トランスレータオプション LEASM を設定します (「Natural CICS インターフェイスのインストール」を参照)。この CICS トランスレータオプションは、他の Natural CICS インターフェイスコンポーネントまたはユーザー出口に対して設定しないでください。

## CICS/TS 3.1 より前のバージョンの CICS トランザクションサーバー

---

z/OS および z/VSE において CICS/TS 3.1 より前のバージョンの CICS トランザクションサーバーでは、Natural CICS インターフェイスは LE 対応ではありません (これらの CICS バージョンの LE 対応アセンブラをサポートしていないため)。しかし、LE 対応の 3GL フロントエンドプログラムによって起動する場合、Natural CICS インターフェイスに、同じ LE 機能が提供されます (サンプルプログラム XNCIFRCN (COBOL)、XNCIFRPN (PL/I)、または XNCIFRDN (C))。

「[Natural CICS のサンプルプログラム](#)」も参照してください。

または、Natural CICS インターフェイスニュークリアスを LE 対応にするには、提供される 3 つの LE 対応フロントエンドスタブ NCILEFC (COBOL)、NCILEFP (PL/I)、または NCILEFD (C) のいずれかを Natural CICS インターフェイスニュークリアスにリンクします。この場合、3GL フロントエンドスタブは NCISTART モジュールの前に INCLUDE する必要があります。

基礎をなす LE エンクレーブに関して Natural CICS インターフェイスに通知するために、LE 対応の 3GL フロントエンドプログラムは、CICS TWA のアドレスを渡すことによって、最初に NCI LE スタブプログラム NCILESTB を呼び出す必要があります (上記のサンプルを参照)。このプログラムは、LE 対応で実行されているかどうか、つまり LE エンクレーブが存在するかどうかをチェックして、CICS TWA で Natural CICS インターフェイスに対して適切な情報を設定



します。したがって Natural CICS インターフェイスを LE 対応で実行するには、フロントエンドの呼び出し元は、CICS TWA でフロントエンドパラメータを渡してはいけません。

Natural CICS インターフェイスニュークリアスの LE 対応に加えて、DFHEAI スタブモジュールではなく、CICS 提供の EXEC インターフェイススタブ DFHELII をインストールに使用する必要があります（「Natural CICS インターフェイスのインストール」を参照）。



# 13 Natural CICS の特殊な機能

---

- Natural 以外のプログラムの呼び出し ..... 82
- CICS 環境の Natural でのダミーの画面 I/O ..... 83
- NCISTART - Natural CICS ニュークリアス ..... 84

Natural CICS インターフェイスドキュメントのこの章では、Natural CICS の特殊な機能について説明します。

### Natural 以外のプログラムの呼び出し

---

Natural タスクが開始時に行う最初のアクションの1つは、異常終了処理の出口をアクティブにすることです。この出口は、異常終了の際にスレッドを含むすべてのリソースを解放するために使用します。このため、Natural 以外のプログラムで EXEC CICS ABEND CANCEL または同等のマクロレベル要求を発行してはいけません。これは、このような要求によって現在のセッションがキャンセルされ、アクティブな出口が無視されるためです。この場合、Natural はリソースをクリーンアップすることができず、スレッドおよびロール機能は解放されません。

Natural プログラムがアクティブな場合は常に、スレッドは Natural タスクに割り当てられます。これは、Natural 以外のプログラムが（CICS リンケージ規約後に）呼び出された場合も該当します。

このため、このようなプログラムは、間に Natural が制御を受け取ることなく、過度の I/O および他の処理を行うべきではありません。Natural 以外のプログラムが会話型の画面 I/O を実行している場合は、呼び出しステートメントの前に、Natural 以外のプログラムを呼び出す Natural プログラムに SET CONTROL 'P=V' ステートメントをコード化することができます。これは、Natural 以外のプログラムを呼び出す前に、パラメータデータはスレッドからコピーされ、セッションはロールアウトされることを示します。

### 標準リンケージ規約を介した Natural 以外のプログラムの呼び出し

Natural 以外のプログラムは、基礎となるオペレーティングシステムおよび／または TP モニタシステム内でプログラムが起動される方法で、Natural によって起動（CALL）されます。

CICS では、Natural 以外のプログラムは EXEC CICS LINK 要求によって起動されます。しかし、例えば同じサブルーチンプログラム（CICS またはオペレーティングシステム要求を発行しない）が、バッチおよびオンライン処理の両方に使用される場合、Natural 以外のプログラムも CICS 標準リンケージ規約を使用して、つまり BALR R14、R15 を介して起動される場合があります。

詳細については、『端末コマンド』ドキュメントで端末コマンド %P=S を参照してください。NCMPRM マクロの SLCALL パラメータも参照してください。

## COMMAREA のパラメータ値による Natural 以外のプログラムの呼び出し

デフォルトでは、Natural 以外のプログラムは、TWA および COMMAREA で渡された、要求パラメータおよびフィールドディスクリプタリスト (R1 および R2。Natural の『ステートメント』ドキュメントで CALL ステートメントの説明も参照) のアドレスを使って呼び出されます (NCIPARM パラメータ COMACAL の設定に依存)。

より CICS に近いのは CICS COMMAREA にパラメータ値を渡す方法で、特に呼び出されるプログラムが別の CICS リージョンに存在する場合 (分散プログラムリンク (DPL)) がそうです。この場合、「呼び出している」リージョン内のアドレスには「呼び出される」リージョンからはアクセスできないためです。

詳細および制限については、『端末コマンド』ドキュメントで端末コマンド %P=C を参照してください。

必要条件：

この機能を使用するには、NCIPARM で COMACAL を YES に設定する必要があります。

パラメータ値が CICS COMMAREA に渡されるとき、CICS TWA の最初の 2 ワードは低い値です。つまり、R1 および R2 の情報は渡されません。

## CICS 環境の Natural でのダミーの画面 I/O

SET CONTROL '00' ステートメントが Natural ステートメントの前に配置され、画面 I/O が実行されると、端末出力は CICS 環境の Natural によって実行されません。したがって、Enter キーおよびユーザー入力も Natural に渡されません。

この機能は、次のような状況に便利です。

1. 擬似会話型画面 I/O を終了し、Natural によって呼び出された Natural 以外のプログラムに移る場合。Natural 以外のプログラムは EXEC CICS SEND 操作を実行し、Natural に戻ります。SET CONTROL '00' ステートメントのために、次の Natural 画面 I/O は、擬似会話型セッションのタスクを終了します。画面入力時に、Natural は制御を受け取り、Natural 以外のプログラムを再度起動して、次に EXEC CICS RECEIVE を実行します。

2. 端末オペレータの介入を必要とせず、「実行中に」 Natural 擬似会話型のトランザクション ID を変更する場合。

```
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 'QQ'
INPUT 'DUMMY' /* dummy I/O, which you will never see
WRITE 'HELLO' *INIT-PROGRAM /* now the new transaction ID is active
```

3. 別の CICS AOR (アプリケーション所有領域) においても、端末オペレータの介入を必要とせず、Natural 外のアプリケーションに切り替える場合。

```
* 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 'QQ'
INPUT 'DUMMY' /* dummy I/O, which you will never see
WRITE 'HELLO' *INIT-PROGRAM /* now the new transaction ID is active
```

この場合、Natural の再スタートデータを CICS COMMAREA に渡すときにスタックすることは、起動されるアプリケーションの責任です。COMMAREA は、新しい (擬似会話型) アプリケーションでも使用される可能性が高いためです。

## NCISTART - Natural CICS ニュークリアス

---

NCISTART (つまりエントリポイントとして NCISTART を持つ Natural CICS ニュークリアス) は CICS 静止段階 1 または 2 の実行のために CICS PLTSD に配置するのに適しています。

- 静止段階 1 で実行する場合、NCISTART は SYSTP スナップショット機能を実行します (Natural の『ユーティリティ』ドキュメントの「SYSTP ユーティリティ」で説明)。
- 静止段階 2 で実行する場合、NCISTART は SYSTP スナップショット機能を実行する前にアクティブなすべての Natural セッションを強制終了します。

NCISTART は、CICS COMMAREA (CICS/TS) または TWA の初めに (CICS/VSE バージョン 2.3)、関連する CICS 端末エントリアドレスを持つノードエラープログラムによって (CICS LINK を介して) 呼び出されるロジックを保持しています。

# 14 Natural CICS のサンプルプログラム

---

- Natural CICS ソースライブラリのサンプルプログラム ..... 86
- z/VSE で使用するサンプルプログラム ..... 88

Natural CICS インターフェイスドキュメントのこの部分では、Natural CICS サンプルプログラムの概要について説明します。

## Natural CICS ソースライブラリのサンプルプログラム

次のサンプルプログラムが、Natural CICS ソースライブラリで提供されています。

- フロントエンドプログラム
- バックエンドプログラム
- ユーザー出口
- サブプログラムの呼び出し
- ノードエラープログラム
- 他のプログラム

### フロントエンドプログラム

名前	言語	機能
XNCIFRNP	アセンブラ	CICS 起動時に Natural CICS 環境を初期化する初期化プログラム。
XNCIFRNL	アセンブラ	EXEC CICS LINK を介して Natural を起動するフロントエンドプログラム。
XNCIFRNS	アセンブラ	EXEC CICS START を介して Natural を起動するフロントエンドプログラム。
XNCIFRNX	アセンブラ	EXEC CICS XCTL を介して Natural を起動するフロントエンドプログラム。
XNCIFRCL	COBOL	EXEC CICS LINK を介して Natural を起動するフロントエンドプログラム。
XNCIFRCN	COBOL	これは、LE 対応のために EXEC CICS LINK を介して Natural を起動するダミーのフロントエンドプログラムです。
XNCIFRCS	COBOL	EXEC CICS START を介して Natural を起動するフロントエンドプログラム。
XNCIFRCX	COBOL	EXEC CICS XCTL を介して Natural を起動するフロントエンドプログラム。
XNCIFRPL	PL/1	EXEC CICS LINK を介して Natural を起動するフロントエンドプログラム。
XNCIFRPN	PL/1	これは、LE 対応のために EXEC CICS LINK を介して Natural を起動するダミーのフロントエンドプログラムです。
XNCIFRPS	PL/1	EXEC CICS START を介して Natural を起動するフロントエンドプログラム。
XNCIFRPX	PL/1	EXEC CICS XCTL を介して Natural を起動するフロントエンドプログラム。
XNCIFRDN	C	これは、LE 対応のために EXEC CICS LOAD および BASR を介して Natural を起動するダミーのフロントエンドプログラムです。



## バックエンドプログラム

名前	言語	機能
XNCIBACK	アセンブラ	終了データのダンプ：このバックエンドプログラムは、Natural 終了メッセージおよび終了データをダンプ形式で表示します。非同期タスクから起動された場合、Natural 終了メッセージがオペレータコンソール上で発行され、潜在的な終了データがダンプされます。NCIBACK もバックエンドトランザクション（STR=xxxx または RET=xxxx）によって起動することができます、ここで、xxxx は XNCIBACK に関連するトランザクションコードです。

## ユーザー出口

名前	言語	機能
XNCIDIRX	アセンブラ	システムディレクトリモジュール名の出口：このソースモジュールはサンプルのシステムディレクトリモジュール名の出口を含んでいます（「 <a href="#">NCIDIREX - 出口インターフェイスによるシステムディレクトリモジュール名</a> 」も参照）。
XNCIRDC1	アセンブラ	SYSRDC の出口：このプログラムは SYSRDC ユーティリティのサンプルの出口を提供します。『ユーティリティ』ドキュメントの関連するセクションを参照してください。
XNCIUIDX	アセンブラ	ユーザー ID 出口：このプログラムは、ユーザー ID をテスト/設定するためのサンプルのユーザー出口を提供します（「 <a href="#">NCIUIDEX - ユーザー ID 出口インターフェイス</a> 」も参照）。
XNCITIDX	アセンブラ	端末 ID 出口：このプログラムは、端末 ID のテストおよび/または論理端末 ID またはセッション ID の設定を行うためのサンプルのユーザー出口を提供します（「 <a href="#">NCITIDEX - 端末 ID 出口インターフェイス</a> 」も参照）。

## サブプログラムの呼び出し

名前	言語	機能
XNCI3GC1	COBOL	このプログラムは、CICS 環境の Natural サブプログラムに対するサンプルの COBOL 呼び出しを提供します。
XNCI3GC2	COBOL	このプログラムは、CICS 環境の Natural サブプログラムに対するサンプルの COBOL 呼び出しを提供します。
XNCI3GP1	PL/1	このプログラムは、CICS 環境の Natural サブプログラムに対するサンプルの PL/1 呼び出しを提供します。
XNCI3GP2	PL/1	このプログラムは、CICS 環境の Natural サブプログラムに対するサンプルの PL/1 呼び出しを提供します。

## ノードエラープログラム

名前	言語	機能
XNCINEP1	アセンブラ	このノードエラープログラムは、CICS マクロレベルを使用して NCIZNEP を呼び出します。
XNCINEP2	アセンブラ	このノードエラープログラムは、CICS コマンドレベルを使用して NCIZNEP を呼び出します。

## 他のプログラム

名前	言語	機能
XNCIUCTR	アセンブラ	大文字小文字の切り替え：このプログラムは、端末を大文字／小文字モードに切り替えます。
XNCIGNIT	アセンブラ	「おやすみ」プログラム：このサンプルプログラムは、Natural セッションのクリーンアップのために NCIZNEP を呼び出します。

## z/VSE で使用するサンプルプログラム

z/VSE については、アセンブラで記述されたサンプルプログラムは A book として提供されています。COBOL で記述されたサンプルプログラムは C book として提供されています。PL/1 で記述されたサンプルプログラムは P book として提供されています。C で記述されたサンプルプログラムは H book として提供されています。

# 15 ユーザープログラムからの Natural の起動

---

▪ Natural セッションをアクティブ化するコマンド .....	90
▪ フロントエンドパラメータ .....	91
▪ LINK を介して起動されたフロントエンド .....	92
▪ START を介して起動されたフロントエンド .....	93
▪ XCTL を介して起動されたフロントエンド .....	94
▪ 分散プログラムリンク (DPL) を介して起動されたフロントエンド .....	94
▪ フロントエンドプログラムをバックエンドとして起動 .....	94

Natural CICS インターフェイスドキュメントのこのセクションでは、ユーザープログラムから Natural を起動するさまざまな方法について説明します。

## Natural セッションをアクティブ化するコマンド

---

このセクションでは、次のトピックについて説明します。

- EXEC CICS XCTL または EXEC CICS LINK の使用
- EXEC CICS START の使用
- 分散プログラムリンク (DPL) の使用
- サンプルプログラム
- 外部サブルーチン CMTASK の使用

Natural セッションは、次のコマンドのいずれかを使ってユーザーのフロントエンドプログラムからアクティブ化することができます。

- EXEC CICS XCTL
- EXEC CICS LINK
- EXEC CICS START

または同等の CICS マクロレベル要求。

### EXEC CICS XCTL または EXEC CICS LINK の使用

EXEC CICS XCTL/LINK を使用する場合、Natural によって使用されるパラメータは、CICS COMMAREA または TWA で渡すことができます。

- Natural は、セッション初期化時に、渡された COMMAREA の長さを検査することによって起動パラメータの場所を決定します。
- 長さが 22 の場合は、Natural は COMMAREA でパラメータを見つけようとします。これ以外の場合は、TWA に渡されたと見なされます。

フロントエンドプログラムを適切に識別するには、フロントエンドパラメータリストの最初の 4 バイトが現在のトランザクション ID を示している必要があります。

フロントエンドプログラムに関連するトランザクション ID には、Natural の TWA サイズ以上の TWA サイズが必要です。「*ncitransact*」も参照してください (Natural の『インストール』ドキュメントの「Natural CICS インターフェイスのインストール」の手順 20)。

## EXEC CICS START の使用

EXEC CICS START を使用する場合、Natural によって使用されるフロントエンドおよびダイナミックパラメータは、EXEC CICS START コマンドの FROM (...) および LENGTH (...) で渡すことができます。これらのパラメータについては、次のページで説明します。

## 分散プログラムリンク (DPL) の使用

リモートリージョンをポイントする SYSID パラメータで EXEC CICS LINK を使用する場合、Natural によって使用されるフロントエンドおよびダイナミックパラメータは CICS COMMAREA で渡す必要があります。また、Natural の要件を満たす TWA サイズのミラートランザクションのトランザクションコードを指定する TRANSID パラメータも指定する必要があります。

非同期 Natural セッションと同じ制限が、DPL を介して起動される Natural セッションに適用されることに注意する必要があります。つまり、入力は不可能であり（したがって、同じダイナミックパラメータ設定を推奨）、呼び出し元は Natural セッションの終了後に制御を受け取ります。

## サンプルプログラム

さまざまなプログラミング手法に対応するサンプルプログラムを、Natural CICS ソースライブラリで提供しています。「[Natural CICS のサンプルプログラム](#)」も参照してください。

## 外部サブルーチン CMTASK の使用

外部サブルーチン CMTASK を呼び出すことによって、Natural プログラムから Natural セッションを開始することができます。ライブラリ SYSEXTP で Natural のサンプルプログラム ASYNCICS を参照してください。

## フロントエンドパラメータ

ユーザーのフロントエンドプログラムから Natural を起動するためには、次のパラメータリストを入力します。

場所	内容
1 - 4	トランザクション ID の起動  この値は現在のトランザクション ID と等しい必要があります。呼び出しているトランザクション ID から、Natural はユーザーのフロントエンドプログラムによって呼び出されたことを識別します。

場所	内容
	XCTL で呼び出されている場合は、リターンプログラム名が指定されていない限り、トランザクションは RETURN と TRANS ID により Natural セッションの終了時に再スタートされます (5 つ目のパラメータを参照)。
5 - 8	<p>ダイナミックパラメータ文字列のアドレス/オフセット</p> <p>ダイナミックパラメータの上書きが評価される場合は、この値は、ダイナミックパラメータ割り当て文字列の 12 バイト前に位置するアドレスに設定する必要があります。</p> <p>START または DPL で呼び出される場合は、フィールドは、ダイナミックパラメータ割り当て文字列の 12 バイト前に位置するアドレスのオフセット (フロントエンドパラメータリストの開始位置に相対的) に設定する必要があります。</p>
9 - 10	<p>ダイナミックパラメータ文字列の長さ</p> <p>ゼロはパラメータが渡されないことを示します。許可される最大長は 32760 です。最大値を超えると、セッションが終了して対応するエラーメッセージが表示されます。</p>
11 - 14	<p><b>Natural トランザクション ID</b></p> <p>指定された値は、START または XCTL で呼び出されるときに、擬似会話型 Natural セッションの制御に使用されるトランザクション ID です。このトランザクションは、Natural セッションが擬似会話型モードで、つまり各端末 I/O で再スタートされるたびに起動されます。</p> <p>Natural トランザクション ID が指定されていない場合は、Natural は、現在の CICS タスクを開始したトランザクション ID を再スタートして、フロントエンドプログラムは各擬似会話型 I/O の後に制御を回復します。</p>
15 - 22	<p>バックエンドプログラム名</p> <p>この 8 バイトの値は、Natural セッションの終了時に制御が転送されるプログラム名です。この場合、RETURN と TRANSID を使ってトランザクション ID の呼び出しを再開するのではなく、CICS XCTL コマンドが使用されます。</p> <p>このフィールドの最初のバイトが数値の場合は、Natural はバックエンドをアクティブにすることなく単に RETURN を実行します。このフィールドの代わりに、Natural プロファイルパラメータ PROGRAM を使用することもできます。</p> <p>Natural 以外のバックエンドプログラムを呼び出すための規則については、Natural の『オペレーション』ドキュメントを参照してください。</p>

## LINK を介して起動されたフロントエンド

フロントエンドに戻るときに、Natural は TWA で、セッションが終了したかどうかを示します。セッションが終了した場合は、TWA に通常のバックエンド情報が保持されます (『オペレーション』ドキュメントの「バックエンドプログラムの呼び出し規則」を参照)。それ以外の場合は、Natural は TWA の最初の 4 バイトに NEXTTRANSID を出力します。

Natural が擬似会話型モード（プロファイルパラメータ PSEUDO を ON に設定）で実行されており、EXEC CICS LINK（または同等の CICS マクロレベル要求）によって起動された場合、Natural が端末に書き込みを行い入力を待つたびに、オリジナルの呼び出し側のトランザクションが起動されます。つまり、Natural は再スタート情報を CICS 一時ストレージに書き込んだ後に「論理」CICS RETURN TRANSID (...) を発行します。

呼び出し側のトランザクションはこの状況を認識して（例えば NEXTTRANSID が送信されたかどうかをチェックしたり、NCOMxxxx TS レコードの存在によって認識する。NCOM は Natural CICS パラメータ生成オプションで、xxxx は端末 ID）、制御を Natural に渡す必要があります。

この方法の利点として、セッション中にフロントエンドプログラムは、制御を別のアプリケーション（例えば COBOL）に渡して後で Natural セッションを再開することを決定できます。

詳細については、『パラメータリファレンス』ドキュメントで PSEUDO パラメータの説明を参照してください。

設計によって、Natural はセッション終了時に LINK フロントエンドプログラムをバックエンドプログラムとして処理します。つまり、バックエンドプログラムの呼び出し規則が適用されません。

## CICSplex 環境

参加しているすべての CICS AOR が、（例えば適切な CICS TST 定義を介して）NCOMxxxx TS レコードにアクセスできることを確認してください。

または、LINK フロントエンドプログラムが、タスク終了時に CICS 一時ストレージ内の NCI セッション再スタート情報を取得し、それを CICS COMMAREA で渡すこともできます。そのような COMMAREA は、セッション再開のために Natural を起動する前に、CICS 一時ストレージに再度出力される必要があります。

## START を介して起動されたフロントエンド

Natural セッションが開始されたタスクである場合は（つまり、START を実行されたフロントエンドユーザープログラムによる EXEC CICS START または EXEC CICS LINK/XCTL コマンドによって起動された場合）、Natural は、最初に COMMAREA によって提供された起動パラメータ、次に TWA のパラメータをスキャンして、最後に EXEC CICS RETRIEVE コマンドによって必要なパラメータを取得しようとします。

## XCTL を介して起動されたフロントエンド

---

Natural セッションが XCTL を使ってフロントエンドプログラムから開始され、リターンプログラムが指定されていない場合（つまりセッション起動パラメータの 5 番目のパラメータの指定も、Natural ダイナミックパラメータまたは NTPRM マクロの PROGRAM 指定もされていない場合）、Natural は内部的に PROGRAM='RET=xxx' 指定をシミュレートすることによって RETURN と TRANSID を介してセッション終了時にユーザーフロントエンドトランザクションを再スタートします。ここで xxx はフロントエンドトランザクションコードです。

ループ状況を回避するために、新しいセッションを開始するか、古いセッションを再開するかを決定するロジックがユーザーのフロントエンドルーチンに含まれている必要があります。

## 分散プログラムリンク（DPL）を介して起動されたフロントエンド

---

Natural セッションが DPL を介して起動された場合、Natural は最初に、サーバーリージョンで直接起動されたか、ローカルフロントエンドプログラムによって EXEC CICS LINK/XCTL を介して間接的に起動されたかを判別します。直接起動された場合は、Natural は CICS COMMAREA から起動パラメータを取得します。間接的に起動された場合は、Natural は COMMAREA で提供された起動パラメータをスキャンし、次に TWA 内のパラメータをスキャンします。リターン時に、ローカル LINK フロントエンドプログラムが使用可能な場合は、Natural は通常のバックエンドデータを TWA で渡します。それ以外の場合は、終了メッセージと潜在的なバックエンドデータをリモートクライアントの COMMAREA で返します。

## フロントエンドプログラムをバックエンドとして起動

---

Natural セッションがフロントエンドプログラムから開始され、このプログラムがリターンプログラムとしても指定されている場合、ユーザーのフロントエンドは、開始するトランザクション ID もチェックする必要があります。

この状況は特に、フロントエンドプログラムが擬似会話型モードではなく、Natural が会話型モードである場合に該当します。

この場合、Natural は終了されずに再度起動されますが、起動パラメータの最初のパラメータは Natural トランザクション ID なので、このときはフロントエンドプログラムによって呼び出されたことは検出されません。



# 16 CICS 環境での Natural の非同期処理

---

■ Natural の非同期処理 .....	96
■ CICS 環境での非同期 Natural セッション .....	96
■ テストおよびデバッグ .....	97

この章では、CICS 環境で非同期 Natural を使用している場合に適用される特別な考慮事項を説明します。

## Natural の非同期処理

Natural の非同期処理については、『オペレーション』ドキュメントの「非同期処理」で概説していますが、CICS 環境で実行するときは、追加の考慮事項があります。これについて、以降のセクションで説明します。

## CICS 環境での非同期 Natural セッション

適切な SENDER および OUTDEST の宛先が、非同期 Natural セッションに対して指定されていることを確認してください。指定されていないと、出力（予期しないエラーメッセージなど）のために異常終了します。


また、適切なメッセージ交換トランザクション ID (`MSGTRAN`) が Natural パラメータモジュール NATPARM で指定され、CICS に定義されていることを確認してください。

SENDER および OUTDEST に対する CICS 端末 ID および一時的なデータの宛先に加えて、Natural CICS インターフェイスでは次のキーワードがサポートされます。

DUMMY	すべての出力が無視されます。
CONSOLE	すべての出力がオペレータコンソールにルーティングされます。  内部的には、NCMPRM マクロの <code>CONSOLE</code> パラメータを介して定義された端末 ID が使用されます。コンソールを処理するときは、プロファイルパラメータ <code>TTYPE</code> または <code>ASYL</code> に設定した端末コマンド <code>%T=</code> などを使用して、端末タイプを適切に切り替える必要があります。

デフォルトでは、3270 データストリームプロトコルが、CICS 環境での非同期 Natural セッションの出力に使用されます。

Natural 出力データを 3270 端末またはプリンタ制御情報なしで、例えば CSSL などの CICS メッセージの宛先に送信することもできます。これを行うには、`SET CONTROL 'T='` ステートメントを使用して行モードに切り替えるか、プロファイルパラメータ `TTYPE=xxxx` で開始します。xxxx は BTCH または ASYL です。こうすると、すべての Natural 出力が 1 行ずつ送信され、Natural プロファイルパラメータ `EJ` が ON に設定されていると行の先頭に ASA 制御文字が付きます。EJ=OFF が設定されていると、制御文字は送信されません。

 **注意:** `SET CONTROL 'T=xxxx'` または `SET CONTROL '+'` を使用する場合、またはパーソナルコンピュータのサポートが有効である場合（プロファイルパラメータ `PC` を ON に設定）、Natural システム変数 `*DEVICE` が変更されます。つまり、この変数を使用して非同期 Natural セッションを判別することができなくなります。

非同期 Natural セッションの一部のパラメータ設定は、NCMPRM 生成パラメータ RCVASYN を YES に設定することによって強制することができます。

## テストおよびデバッグ

最近の CICS バージョンでは、CICS での非同期タスクの追跡を有効にするトランザクション CEDX が提供されています。以前の CICS バージョンにはこの機能はなかったため、このようなデバッグは端末内のタスクでのみ可能でした。

Natural CICS インターフェイスでは、このような場合のための支援機能を提供しています。非同期 Natural セッションを、端末からそのセッションを開始することによってテストすることができます。このとき、ダイナミックパラメータ文字列の最初の 5 文字として ASYN を使用するか、またはプロファイルパラメータ TTYPE=xxxx を使用します。xxxx は ASYN または ASYL です。次に、Natural CICS インターフェイスは非同期 Natural セッションを設定します。

このエミュレーションは Natural に関してのみ 100 パーセントです。CICS ではこのタスクは引き続き端末内として扱われます。



# 17 CICS 環境での Natural セッションのログ

---

■ ログイン機能 .....	100
■ Natural ログファイルの定義 .....	100
■ Natural のログレコード .....	101

この章では、Naturalセッションに関する情報を、バッチモードで処理および評価できるファイルにログする方法について説明します。

## ロギング機能

---

オプションとして、Naturalセッションに関する情報を、バッチモードで処理および評価できるファイルにログすることができます。

現在のシステムの使用状況に関するスナップショットを提供するだけのオンライン SYSTP ユーティリティとは違い、このロギング機能を使用すると、より長期間にわたって Natural CICS システムの使用状況を追跡することができます。

### 特別な考慮事項

- 複数の Natural CICS 環境で（つまり一意のスレッド、ロール機能、スワッププールおよびバッファプールを持つ複数のシステムディレクトリ）、同じ Natural ログの保存先を共有することができます。SCP 環境を初期化すると、「システム ID」がシステムディレクトリに書き込まれます。このシステム ID は、Natural CICS システム環境によるログレコードを「ソート」する評価プログラムの一部です。
- Natural ログファイルは Natural アプリケーション CICS に定義することをお勧めします。「リモート」ログファイルにログを記録するとパフォーマンスが低下するためです。
- ログファイル評価プログラムを実行する場合は（Natural の『ユーティリティ』ドキュメントの「バッチモードでの SYSTP」を参照）、ログファイルは CICS で閉じられている必要があります。閉じられていないと、最後のバッファがストレージに残っていたり、ファイルに EOF レコードが見つからないことが原因となり、予測できない結果が発生する場合があります。
- 十分なディスクスペースを Natural ログファイル用に予約する必要があります。可能であれば、ログファイルは増分割り当てを使用して定義してください（z/VSE でファイルがいっぱいになると、z/VSE メッセージ NO MORE AVAILABLE EXTENTS が発行され、オペレータは新しいエクステントを入力するか CICS をキャンセルするように指示されます）。

## Natural ログファイルの定義

---

Natural ログファイルはシーケンシャルディスクファイル、つまり、CICS における「追加パーティションの宛先」です。デフォルトでは、ログファイルの内部（論理）名は NLOG です。この名前は、NCMPRM マクロで LOGDEST パラメータを指定することによって変更できます。

ログファイルは、関連するデータセット制御情報（DCT の TYPE=SDSCI エントリ）を使って TYPE=EXTRA として CICS DCT で定義する必要があります。このファイルは CICS 起動 JCL にも定義する必要があります（z/OS の DD ステートメント、z/VSE の DLBL ステートメント）。

## Natural のログレコード

---

Natural のログファイルには、次のレコードをログすることができます。

- Natural CICS システムの再起動レコード
- Natural セッション終了レコード

### Natural CICS システムの再起動レコード

長さ =96

SCP システムの初期化が正常に終了した後で、初期化日時、共通のシステム接頭辞などの他のシステムデータ、RCB の数またはスレッドグループ数を保持するレコードがログファイルに書き込まれます。

この最初のログ要求に失敗すると、Natural ログファイルはシステムディレクトリで利用不可としてフラグが付けられ、ログは記録されなくなります。

SYSTP ユーティリティの対応するシステム管理機能によってシステムハイウォーターマークがリセットされると、そのたびにシステム再起動レコードが書き込まれます。システムの起動情報に加えて、これらのレコードには端末 ID および SYSTP ユーザーのユーザー ID が含まれています。

### Natural セッション終了レコード

長さ =216

Natural セッションの正常終了時または異常終了時に、セッションログレコードがログファイルに書き込まれます。このレコードは、内部的に 6 つに分割されます。

1. 実際のセッション統計を保持するレコード制御部：
  - 現在の日時（つまりセッションが終了した日時）
  - セッションがアクティブ化された Natural CICS 環境を示すシステム ID
  - レコードタイプ = セッションレコード

レコード制御部は、異なるレコードタイプを識別するためにすべての Natural ログレコードで共通しています。NCMLOG マクロはレコードのレイアウトを保持しています。

2. 実際のセッション統計を保持するユーザーセッション部：
  - 端末 ID
  - (最後の) ユーザー ID
  - セッション開始日時
  - セッションによって割り当てられた最大ストレージ

- セッションの再開／スワップイン／ロールインの数
  - セッションによってロールされたレコードの最大数（存在する場合）
3. セッションに関連するスレッドグループの現在のデータを保持するスレッドグループ部
- スレッドグループ番号
  - グループ内の TCB スロット数（存在する場合）
  - グループの共通スレッドサイズ
  - セッションによってグループ内で割り当てられた最大ストレージ
  - グループ内でアクティブなセッションの最大数
  - グループの待機キューの最大サイズ（TYPE=SHR のスレッドグループ）、およびグループ内で同時にアクティブなセッションの最大数（TYPE=GETM のスレッドグループ）
  - この待機キューの最大サイズに達した回数
4. セッションによって最後のスレッドとして使用された TYPE=SHR スレッドのデータを保持するスレッド部（使用された場合）：
- スレッド名
  - スレッド使用数
  - セッションによって使用されたスレッドの最大ストレージ
  - このスレッドに対するセッション再開／ロールインの数
  - このスレッドの待機キューの最大サイズ
  - この待機キューの最大サイズに達した回数
5. セッションが割り当てられたロール機能（割り当てられた場合）に関する情報を保持するロール機能部：
- ロール機能名
  - このロール機能に割り当てられたセッションの最大数
  - ロール機能のレコードサイズ
  - ロール機能のスロットサイズ
  - このロール機能のスロット数
  - このロール機能に対するロールアウト／ロールインの最大数
6. グローバルなシステム使用状況に関する統計を保持するシステムディレクトリ部：
- UCB ブロックエクステンションの最大数
  - システム内でアクティブなセッションの最大数
  - SCP において同時にアクティブなセッションの最大数
  - SCP システムリカバリの回数



設計により、セッション終了レコードはセッションの日時別に格納されます。つまり、後のセッションレコードの3~6の部分は常に、前のセッションレコードの同じ部分より新しい情報を保持しています。レコードの3~6の部分は、ログファイル評価プログラムによって、提供された対応する情報、つまり、スレッドグループ、スレッド、ロール機能、および SCB に関する情報を更新するために使用されます。

この方法により、CICS が制御不能になって終了した場合に、Natural CICS システムリソースに関する最新の情報が保存されます。

セッション終了ログレコードは、対応するセッションによって使用されたリソースのみを反映します。このため、これらのレコードは SCP 環境を完全に反映していない場合があります。完全な SCP 環境のレポートを取得するには、SYSTP システム管理機能を使用するか、または CICS 環境の Natural を CICS PLTSD に置くことで（「[Natural CICS の特殊な機能](#)」を参照）、環境全体のスナップショットを作成します。

Natural ログファイルのシステムのスナップショットレコードは、2の部分に記載されるようなセッション固有の情報のないセッション終了レコードを示します。



# 18 Natural CICS のパフォーマンスに関する考慮事項

---

■ 環境固有の考慮事項 .....	106
■ ロール機能の選択 .....	106
■ 共有ストレージのスレッドと GETMAIN を実行したスレッドの違い .....	109
■ CICS のパラメータ設定 .....	112
■ 行圧縮システム .....	113
■ 擬似会話型トランザクションと会話型トランザクションの違い .....	113
■ Natural と Adabas .....	113
■ CICS モニタリング製品 .....	113

この章では、CICS 環境の Natural を設定するためのガイドラインを説明します。

### 環境固有の考慮事項

---

環境固有の考慮事項は以下のとおりです。

- CICSplex 環境（z/OS のみ）で Natural を実行する場合は、Natural ロールサーバーを使用する必要があります。
- しかし、単一の CICS リージョンで Natural をローカルに実行する場合は、複数の選択肢があります。

選択肢の 1 つは（z/OS のみ）、Natural ロールサーバーを使用することです。CICS ロール機能およびスワッププールを使用することに対するロールサーバーの利点は、Natural ロールサーバーは、CICS リージョンに非同期に実行されるため、スワッププールよりデータスペースにおいて多くのロールバッファを提供できることです。

### ロール機能の選択

---

このセクションでは、次のセクションについて説明します。

- [コントロールインターバル](#)
- [VSAM ロールファイルと CICS 一時ストレージの違い](#)
- [CICS の補助の一時ストレージの使用](#)
- [CICS のメインの一時ストレージの使用](#)
- [VSAM RRDS ロールファイルの使用](#)
- [CICS 環境での Natural スワッププールの使用](#)

#### コントロールインターバル

両方のロール機能（VSAM および補助の一時ストレージ）をコントロールインターバルの最大サイズである 32 KB に定義することを強くお勧めします。これにより、ロールを実行するのに必要な I/O の数および CPU オーバーヘッドを最小限にすることができます。

コントロールインターバルのサイズを 32 KB 未満にする理由には、ディスクトラックを有効利用できる、または VSAM バッファ用仮想ストレージを使用するということもあります。

## VSAM ロールファイルと CICS 一時ストレージの違い

CISIZE/レコードサイズが同じ場合は、一時ストレージは VSAM ロールファイルより CPU オーバーヘッドが少なくなります。

一時ストレージに  $n$  個のレコードを書き込むには、 $n+1$  個の CICS 要求を発行する必要があります (DELETEQ に 1 個、PUTQ に  $n$  個)。これに対して VSAM ロールファイルには、VSAM トランザクションロジック  $n$  倍 (UPDATE の READ に加えて REWRITE) のために  $2n$  個の要求を発行する必要があります。

VSAM 更新要求に対しては物理的な I/O が常に実行されます。これに対して、一時ストレージ (AUX) レコードに対してはバッファが実行されるため、多くの場合、読み込まれるレコードはバッファに残っています。

しかし CICS 一時ストレージは、他のアプリケーションにも使用されている場合にはボトルネックになる場合があります。

特に I/O の競合を回避できる場合は、Natural で VSAM ロールファイルを使用する方がこの状況を克服できます (ただし追加の VSAM バッファスペースが必要です)。他の要件のために CICS 一時ストレージファイルにこのレコードサイズを指定できない場合は、最適/最大の CISIZE/レコードサイズを持つ VSAM ロールファイルは特に効果的です。

CICS 一時ストレージを Natural 専用に行える場合は、常に CICS 一時ストレージを使用してください。CICS 一時ストレージが他のアプリケーションにも使用される場合は、VSAM ロールファイルを使用するときの方が Natural のパフォーマンスが優れているかどうかを評価する必要があります。

CICS 一時ストレージを使用しても Natural のパフォーマンスが低下しない場合は、CICS 一時ストレージをロール機能に選択し、「節約できた」VSAM ロールファイルのバッファスペースは、TS バッファやスレッドの追加に使用してください。

## CICS の補助の一時ストレージの使用

主なロール機能は VSAM RRDS です。一時ストレージのデフォルトタイプは AUXILIARY です。

VSAM ロールファイルを使用している場合は、CICS セッション中にすべてのロールファイルがいっぱいになるか使用不可能になったときに、Natural CICS インターフェイスは一時ストレージ (AUX) を使用します。

ただし、ロールファイルを使用したくない場合、またはロールファイルが正しくインストールされていない場合、CICS 環境の Natural はすべてのロールに対して一時ストレージ (AUX) を使用します。一時ストレージ (AUX) をロールファイルとして使用する場合、このファイルのコントロールインターバルのサイズは最低でも 4KB が必要です。補助の一時ストレージを利用できない場合は、代わりにメインの一時ストレージが使用されます。

CICS SIT パラメータ TS によって定義された VSAM バッファの数は、物理 I/O の数を減らすために適切な値に増やす必要があります。CICS 統計で、このエリアのボトルネックをチェックしてください。

### CICS のメインの一時ストレージの使用

CICS のメインの一時ストレージをロール機能として使用すると、ロールにおいて I/O は実行されませんが、使用されるメインストレージが大きくページングが増加するため、調整を検討する必要があります。

### VSAM RRDS ロールファイルの使用

FCT の BUFNO および STRNO パラメータなど、通常の CICS VSAM ファイルの調整については、VSAM ロールファイルを検討する必要があります。CICS シャットダウン統計で、このエリアのボトルネックをチェックしてください。

ロールファイルは Natural に対して一種のページデータセットとして機能するため、ジャーナリングとロギングが発生します。このため、Natural のロール処理の速度を低下させるものはすべて回避する必要があります。ロールファイルに対するダイナミックトランザクションバックアウト (DTB) およびフォワードリカバリは機能せず、オーバーヘッドの原因となるだけです。

### MRO 環境

パフォーマンスの理由から、VSAM ロールファイルは Natural アプリケーションが実行されているのと同じ CICS システムで定義する必要があります。MRO 機能シップは起動しないでください。利用できるバッファが十分にある場合は、CICS のローカル共有リソース (LSR) を使用することができます。

### Natural に対する個別の LSR プール

スレッド数より大きな文字列数 (STRNO) で、Natural ロールファイルに対して個別の LSR プールを定義することをお勧めします。バッファ数もスレッド数より大きい必要があります。バッファ数が大きいほど索引のヒット率が上昇します。

### CICS 環境での Natural スワッププールの使用

大量の VSAM 一時ストレージ (AUX) バッファまたは一時ストレージ (MAIN) よりも、スワッププールを使用することを強くお勧めします。

Natural スワップマネージャは、圧縮されたセッションストレージを非常に効率的に処理し、CPU および I/O のオーバーヘッドを削減します。スワッププールのサイズは、できるだけ大きくしてください。例えば、50 KB スロットに適合する 50 のセッションを保持するには、2.5 MB のスワッププールが必要です。

パフォーマンスの観点からは、スワッププールのバックアップ機能としてメインの一時ストレージを使用することは意味がありません。これらの機能はいずれも CICS メインストレージを使用

するためです。しかし一般的には、CICSのサービスオーバーヘッドがなくなるため、スワッププールを使用の方が効率的です。メインの一時ストレージにオーバーフローするよりも、スワッププールを拡大してディスクストレージ（つまりVSAMロールファイルまたは補助ストレージ）をバックアップ機能として使用する方が有利です。

仮想ストレージがボトルネックになる場合は、ロール機能バッファの数およびスレッド数などを、スワッププールのために最小限にする必要があります。

Natural スワッププールキャッシュを使用する場合は、Natural の最大スレッドサイズのロールバッファが、スワッププールとその（データスペース）キャッシュ間での Natural セッションデータの転送に必要です。このロールバッファはスワッププールの GETMAIN から得られます。つまり、スワッププールに実際に利用できるストレージのサイズは、指定されたサイズから Natural の最大スレッドサイズを引いたものです。

したがって、スワッププールのサイズおよびそのキャッシュのサイズの両方が、Natural の最大スレッドサイズの少なくとも 2 倍である場合にのみ、Natural スワッププールキャッシュが割り当てられます。

## 共有ストレージのスレッドと GETMAIN を実行したスレッドの違い

このセクションでは、次のセクションについて説明します。

- ストレージの使用
- ストレージ使用の制御
- スワッピング／ローリング
- CICS/TS に関する考慮事項
- 結論

### ストレージの使用

共有ストレージのスレッドは、Natural CICS システムの初期化時にあらかじめ割り当てられます。つまり、アクティブなセッションがあるかどうかに関係なく、これらのスレッドの対象となるストレージは Natural CICS システム専用で使用されます。これに対して、GETMAIN を実行したスレッドは、CICS タスクがアクティブな間のみ存在します。

## ストレージ使用の制御

共有ストレージのスレッド (TYPE=SHR) については、CICS 環境の Natural は、Natural の初期化時にあらかじめ割り当てられたストレージを常に使用します。このため、Natural スレッドによって使用されるストレージのサイズは容易に予測することができます。しかし、GETMAIN を実行したスレッド (TYPE=GETM) については、実際に使用されるストレージは、現在アクティブな Natural セッションの数によって異なります。

Natural 自体には GETMAIN を実行したスレッドの最大数を設定するメカニズムはありませんが、Natural トランザクションコードを TRANCLASS (CICS バージョン 4.1 より前の TCLASS) にグループ化することによって制御することができます。トランザクションコードがこのようなクラスに属する場合は、並行タスクの最大数は、TRANCLASS 定義の MAXACTIVE パラメータによって (または CICS バージョン 4.1 より前の CICS システム初期化テーブル (SIT) の CMXT パラメータを使用して) 調整することができます。

## スワッピング／ローリング

Natural セッションが共有ストレージのスレッドを解放するとき、別のセッションが特にそのスレッドを使用する必要がなければ、セッションデータは非圧縮形式でスレッドに保存されます。この場合、新しいセッションが古いセッションのデータを保存する役割を果たします。

このようなアクティビティを「据え置きロール」と呼びます。これにより、利用できるスレッドの数が、同時にアクティブな Natural セッションの数以上であれば、ローリングまたはスワッピングを完全になくすことができます。

これに対して GETMAIN を実行したスレッドを使用するセッションは、常に CICS タスクの終了時に FREEMAIN 操作に先立ってデータを保存します。このため、同時にアクティブな Natural セッションの数に関係なくローリング／スワッピングによるオーバーヘッドが発生します。

Natural トランザクションが大量に発生する環境においては、「即時」ロールによる方法でも「据え置き」ロールによる方法でも、セッションデータの保存に実質的な違いはありません。

処理量が多く、プログラムストレージスレッドに対する Natural セッションの比率が高い Natural 環境においては、これらのスレッドは複数の Natural セッションによって共有されるため、ロールイン／ロールアウトのオーバーヘッドがさらに高くなります。この状況で発生する可能性がある問題として、長期間にわたる Adabas 要求、つまり多くの Adabas 呼び出しを含む Natural タスクによって引き起こされるスレッドの競合があります。

このようなタスクによってスレッドが長期間「ロック」されるのを防ぐために、Natural プロファイルパラメータ DBROLL を適切に使用することによって、これらのタスクのスレッドを強制的に解放することができます。

しかし、GETMAIN を実行したスレッドの場合は、複数の Natural セッション間の競合は発生しません。これは、TYPE=GETM スレッドは以前割り当てられていた Natural セッションにのみ属するためです。



このように、TYPE=GETM スレッドは共有されることのない「使い捨ての」リソースと考えることができます。これに対して、TYPE=SHR スレッドは共有できる「複数回使用可能な」リソースと考えることができます。

## CICS/TS に関する考慮事項

z/OS における CICS/TS の最も重要な機能は、トランザクションアイソレーションです。つまり、タスクのストレージを他のタスクから保護する機能です。

Natural でこの機能を利用するには、TYPE=GETM スレッドを使用する必要があります。これらのスレッドはトランザクションアイソレーションの対象となりますが、「共有される」TYPE=SHR スレッドは対象外になるためです。また、CICS/TS では TYPE=SHR スレッドに対する CICS のオーバーヘッドが増加します。

TYPE=GETM スレッドのスレッド選択アルゴリズムは単純ですが（Natural タスクが開始されると、スレッドは CICS GETMAIN を介して割り当てられる）、TYPE=SHR スレッドではもっと複雑になります。この場合は、Natural スレッド環境は NCISTART（キューイングおよびバランス）によって管理されますが、CICS は Natural スレッドに関して何も認識しません。CICS が遅くともタスクの終了時にスレッドを解放する TYPE=GETM スレッドとは違い、TYPE=SHR スレッドの場合、Natural セッションに対するスレッドの割り当て／解放を Natural で行う必要があります。このために、Natural はスレッドコントロールブロック（TCB）のリストを保持しています。

Natural は、タスクの異常終了時に CICS が認識していないセッションリソース（例えば TYPE=SHR スレッド）を解放できるように出口を常にアクティブにしていますが、Natural タスクが、関連する TCB においてスレッドが解放済みとしてマークされることなく終了する状況が発生する場合があります（EXEC CICS ABEND CANCEL 要求が Natural によって呼び出された Natural 以外のプログラムで発行された場合、または Natural セッションがパフォーマンスモニタの KILL トランザクションによってフラッシュされた場合など）。

意図せずにスレッドが使用中のままになっている問題を防ぐために、CICS 環境の Natural では、常にスレッド選択アルゴリズムで、使用中のスレッドに関連する CICS タスクが存在するかどうかをチェックして、存在しない場合はそのスレッドを解放します。

CICS/TS より前の CICS バージョンでは、アクティブな CICS タスクに対するこのチェックは、コントロールブロックのジャンプによって実行されました。つまり、Natural は、タスクの EISTG、TCA、および TQE コントロールブロックの一貫性をテストすることによって、アクティブなタスクをチェックしていました。CICS/TS では、トランザクションアイソレーションのために、別のタスクのストレージにアクセスすることはできません。

CICS/TS でこの機能を実現するために、NCISTART は、スレッド選択ルーチンにおいて使用中と認識されたスレッドに対して EXEC CICS INQUIRE STORAGE TASK() 要求を発行します。つまり、スレッドリソースに対してタスクが最後に ENQueue される前に、CICS 要求があった可能性があります。同じ CICS コマンドが、Natural セッションのシリアライゼーションにも使用されます（TYPE=SHR スレッドの据え置きロールなど）。

## 結論

TYPE=SHR スレッドと TYPE=GETM スレッドのどちらも、長所と短所があります。しかし CICS/TS では、次の理由から TYPE=GETM スレッドが有利です。

- トランザクションアイソレーションのサポート (z/OS のみ)
- より CICS に近い調整の可能性
- TYPE=SHR スレッドのパフォーマンスの低さ

## CICS のパラメータ設定

---

CICS SIT パラメータ AMXT および CMXT は、同時に実行する Natural タスクの数を制御するために使用します。

指定する数は、スレッド数より大きい必要があります。また、非同期 Natural タスクおよび Natural Advanced Facilities スプールタスクに対して、適切な CMXT パラメータで個別のトランザクションクラスを指定することも検討する必要があります。これは、スレッドを占有するこのような「バックグラウンド」タスクが多すぎるために、「通常の」Natural 端末のタスクがログアウトするのを防ぐためです。このようなトランザクションに特別なスレッドグループを定義することもできます。

Software AG の担当者からデバッグ目的のために依頼されない限り、Natural トランザクションに対する CICS ダンプは実施しないでください。Natural は、プログラムチェック以外の異常終了に対して、および Natural セッションパラメータ DU が ON に設定されている場合のプログラムチェックに対して、(EXEC CICS DUMP を介して) ダンプを生成します。Natural ダンプが生成されない場合は、CICS ダンプは冗長でオーバーヘッドを発生させるだけです (特にシステム/リージョンのダンプを作成する場合。これは、スナップダンプが完成するまで CICS システム全体が停止されるためです)。

問題を分析するときに CICS のトレースは不可欠ですが、システムにオーバーヘッドが発生します。また、CICS パフォーマンスモニタリングツールおよびアカウントパッケージは、30 パーセント以上のシステムオーバーヘッドを発生させます。一部のパッケージでは、内部的に CICS トレースを有効にしてからインターセプトします。この潜在的なシステムオーバーヘッドに注意してください。Natural CICS インターフェイスは、CICS コマンドレベルのアプリケーションプログラミングインターフェイスを使用することも注意する必要があります。CICS コマンドレベルの要求は、CICS マクロレベル要求よりかなり多くのトレースエントリを (他のオーバーヘッドとは別に) 生成します。

## 行圧縮システム

Natural は、RA（アドレス反復）および画面イメージなどその他の技術によってデータストリームを最適化します。他の行圧縮システムがインストールされた場合は、Natural トランザクションは、これらのシステムによって処理されないようにする必要があります。この理由は、オーバーヘッドが発生するだけで何の利点もないからです。

## 擬似会話型トランザクションと会話型トランザクションの違い

セッションを再開する場合、会話型の Natural タスクは最初のスレッドにロックされます。つまり、そのとき最初のスレッドを利用できない場合は、会話型のタスクはこのスレッドが空くのを待つ必要があります。しかし、擬似会話型 Natural タスクは柔軟性があるため、空いている任意のスレッドを利用することができます。


言い換えれば、会話型のタスクに見られる「古典的な」利点（画面 I/O 操作によるデータの保存/リストアにおいて I/O が少ない）は、Natural で使用されるスレッド技術のために Natural では活用されません。

## Natural と Adabas

CICS の Natural タスクは Adabas 呼び出しが完了するのを待つため、サービスを提供する Adabas リージョン/パーティションは、待ち時間を最小限にするために CICS リージョン/パーティションより常に優先度が高くなる必要があります。

## CICS モニタリング製品

CICS モニタリング製品は、CICS タスクを除去する機能を備えているため、アプリケーションによって設定された異常終了出口をバイパスできます。

-  **注意:** このような機能を Natural タスクをキャンセルするのに使用しないでください。Natural はリソースをクリーンアップできない場合があります。さらに悪い場合、タスクの処理内容によっては Natural CICS システムが一貫性のない状態になる可能性があります。

Natural セッションをキャンセルするには、Natural SYSTP ユーティリティのセッションのキャンセル/フラッシュ機能を代わりに使用してください。詳細については、Natural の『ユーティリティ』ドキュメントで関連するセクションを参照してください。



# 19 CICS 環境での Natural の出力ファイルおよびワー クファイル

---

- 出力ファイルおよびワークファイルの使用方法のカスタマイズ ..... 116
- CICS 一時ストレージの出力ファイルおよびワークファイル ..... 116
- CICS 一時データの出力ファイルおよびワークファイル ..... 117

この章では、CICS での Natural の出力ファイルおよびワークファイルの使用について説明します。

## 出力ファイルおよびワークファイルの使用方法のカスタマイズ

---

Natural CICS インターフェイスは、補助およびメインの CICS 一時データキューまたは CICS 一時ストレージキューとして、CICS での Natural の出力ファイルおよびワークファイルをサポートします。

使用方法をカスタマイズするには、PRINT および WORK プロファイルパラメータで、次のサブパラメータを設定します。

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

詳細については、以下に示すリンクを参照してください。

- WORK プロファイルパラメータの説明、および上記のサブパラメータ値を設定する方法については、NETWORK パラメータマクロを参照してください。
- PRINT プロファイルパラメータの説明、および上記のサブパラメータ値を設定する方法については、NTPRINT パラメータマクロを参照してください。

Natural CICS インターフェイスの出力ファイルサポートは、トレースおよびログを目的として提供されています。レポートの処理は目的としていません。特に、DEFINE PRINTERのキーワードパラメータ PRTY、CLASS、COPIESなどは、まったく効果がありません。

## CICS 一時ストレージの出力ファイルおよびワークファイル

---

CICS 出力ファイルおよびワークファイルの CICS 一時ストレージキュー（補助およびメイン）は、設計により RECFM=V ファイルで、入出力に利用できます。

CICS 環境の Natural では Natural セッションによる特定の TS キューに対する排他制御はありませんが、プロファイルパラメータ PRINTのサブパラメータ DEST、またはプロファイルパラメータ WORKのサブパラメータ DESTで NCIPARM パラメータ **TERMVAR** に定義されている文字列を指定することによって（デフォルトは&TID）、セッションまたは端末依存の出力ファイルまたはワークファイルを自動的に作成することができます。このような文字列が 8 文字の DEST サブパラメータ内に見つかり、実際の端末 ID と置き換えられます。

## CICSplex 環境

CICSplex 環境で実行する場合、CICS 一時ストレージの Natural 出力ファイルおよびワークファイルは、CICS TST で TYPE=SHARED または TYPE=REMOTE として定義する必要があります。

## NCI システムキュー

CICS 環境の Natural では、NCI システムキューにアクセスすることはできません（NCI システムキューは、NCMDIR マクロの **TSKEY** パラメータに定義された接頭辞を持つ TS キューです）。

## CICS 一時データの出力ファイルおよびワークファイル

Natural CICS 出力ファイルおよびワークファイルの CICS 一時データキューは、CICS DCT で定義する必要があります。間接的な宛先については、基本的な宛先の属性が伝播されます。特に、RECFM または TYPEFLE のような追加パーティションの宛先の属性により、Natural ワークファイルの属性が決定されます。

内部パーティションの宛先では、設計により RECFM=V に設定されており、入出力の両方に利用できます。

CICS 一時データの出力ファイルおよびワークファイルは、複数のセッションがこのようなファイルに対して I/O を発行できるという意味で「共有ファイル」であると言えます。





# 20

## Com-plete/SMARTS 環境の Natural

---

▪ NFMPRM マクロのパラメータ .....	120
▪ 異常終了出口の使用 .....	125
▪ ストレージの使用 .....	126
▪ バックエンドプログラムのサポート .....	126
▪ Natural バッチ実行における Com-plete のサポート .....	127
▪ Com-plete/SMARTS 環境での Natural の非同期処理 .....	127
▪ ユーザープログラムからの Natural の起動 .....	128
▪ ストレージスレッドキーの処理 .....	128
▪ セッション初期化中のユーザー出口処理のサポート .....	128
▪ SMARTS サーバー環境の使用 .....	129
▪ Com-plete のリカバリ可能なセッション処理のサポート .....	132

このドキュメントでは、Natural Com-plete/SMARTS インターフェイス（製品コード NCF）の機能、および Com-plete 環境での Natural の操作および個別のコンポーネントについて説明します。

以下の項目も参照してください。

- Com-plete のドキュメント（Com-plete 製品の詳細について）
- オンライン処理（『Natural システムアーキテクチャ』ドキュメント）
- 次のトピックについては、Natural の『インストール』ドキュメントを参照してください。
- Natural Com-plete/SMARTS インターフェイスの構造と機能
- 必要条件
- Natural Com-plete/SMARTS インターフェイスのインストールテープ
- Natural Com-plete/SMARTS インターフェイスのインストール手順
- Com-plete 環境での Natural サーバーのインストール
- Com-plete \*ULIB 機能の使用
- インストール確認
- SYSTP ユーティリティ（Natural の『ユーティリティ』ドキュメント）
- Com-plete/SMARTS 環境下の Natural のユーザーアベンドコード（Natural の『メッセージおよびコード』ドキュメント）

## NFMPRM マクロのパラメータ

---

Natural Com-plete 環境をカスタマイズするために、NFMPRM マクロの次のパラメータを変更することができます。

EXIT | HCDTID | INITID | LC | LE370 | MSGHDR | NTHSIZE | SERVER | SPIEA | THABOVE | TTYxx |  
UCTRAN | U2PRINT |

## EXIT - ユーザー出口のモジュール名

このパラメータは、Naturalが初期化される前に、セッションの初期化時に呼び出すことができるユーザー出口のモジュール名を定義します。可能な値は次のとおりです。

値	説明：
<i>name</i>	ユーザー出口の名前。

デフォルト値はありません。

## HCDTID - ハードコピー出力先の初期化

このパラメータは、ハードコピーの出力先の初期化を制御します。

可能な値は次のとおりです。

値	説明：
YES	ハードコピーの出力先は端末 ID で初期化されます。
NO	ハードコピーの出力先は論理端末名に対応しています。 これはデフォルト値です。

## INITID - \*INIT-ID の内容

このパラメータは、システム変数 \*INIT-ID の内容を制御します。

可能な値は次のとおりです。

値	説明：
TIBNAM	*INIT-ID には、ユーザー端末の論理ユニット名が含まれています。
TID	*INIT-ID には文字列 <i>1bnnnnnn</i> が含まれています。 <i>1</i> はセッションが実行されているスタックレベル、 <i>b</i> は空白、 <i>nnnnnn</i> は先頭にゼロなしで右詰めされた TID 番号を示します。 これはデフォルト値です (Natural 端末 ID)。
CPATCH	*INIT-ID には INITID=TID と同じ文字列が含まれていますが、 <i>b</i> は空白ではなく Com-plete のパッチ文字です。

## LC - 小文字モードの有効化

このパラメータは端末を小文字モードに設定します。

可能な値は次のとおりです。

値	説明：
YES	小文字モード。 これはデフォルト値です。
NO	大文字モード。

## LE 370 - LE/370 環境の使用

このパラメータは、Complete/SMARTS 環境において、あらかじめ初期化されている環境（CEEPIPI インターフェイス）として LE/370 を使用する方法を指定します。

可能な値は次のとおりです。

値	説明：
YES	すべての 3GL 呼び出しは、あらかじめ初期化された LE/370 エンクレーブで処理されます。
NO	これはデフォルト値です。

## MSGHDR - メッセージヘッダーのアクティブ化

このパラメータは、Com-plete のメッセージ交換機能を非同期 Natural トランザクションに使用して、Natural のエラーおよび終了メッセージのメッセージヘッダーをアクティブまたは非アクティブにします。

可能な値は次のとおりです。

値	説明：
YES	メッセージヘッダーがアクティブになります。 これはデフォルト値です。
NO	メッセージヘッダーが非アクティブになります。

## NTHSIZE - Natural のスレッドサイズ

このパラメータは、Naturalのバッファ、データエリア、およびスレッドに使用されるストレージエリアのサイズを指定します。

可能な値は次のとおりです。

値	説明：
<i>nnnn</i>	サイズ (KB 単位)。
1024	これはデフォルト値です。

このストレージエリアは、Com-plete の物理的なスレッド内に割り当てられます。残りのエリア (Natural トランザクションの Com-plete リージョンサイズ RG から NTHSIZE を引いたサイズ) は、例えば非 Natural サブルーチンのダイナミックなロード、変更可能な Natural スレッドバッファの増加、または Natural ワークプールなどに利用できます。

## SERVER - Natural サーバーの名前

このパラメータは、Com-plete 起動時に初期化される Natural サーバーの名前を定義します。このパラメータは、ローカルバッファプールなど Natural セッションで共通のストレージおよびテーブルを保持するために使用します。サーバーは Com-plete 起動時に定義する必要があります。

可能な値は次のとおりです。

値	説明：
<i>name</i>	Natural サーバーの名前。
NCFNAT42	これはデフォルト値です。

提供されたサーバーモジュール NCFNAT42 を異なる名前でコピーして、異なる Com-plete インターフェイスを異なるサーバーで、つまり同じ Com-plete で異なるローカルバッファプールを使用してリンクおよび実行することができます。

## SPIEA - ABEXIT 出口のアクティブ化

このパラメータは、ABEXIT 出口をアクティブまたは非アクティブにします。

可能な値は次のとおりです。

値	説明：
YES	ABEXIT 出口をアクティブにします。 これはデフォルト値です。
NO	ABEXIT 出口を非アクティブにします。テスト目的にのみ使用してください。

### THABOVE - Natural スレッドの場所

このパラメータは、Natural スレッドの場所を決定します（NTHSIZE パラメータを参照）。

可能な値は次のとおりです。

値	説明：
YES	Natural スレッドは 16MB 境界より上の Com-plete スレッドエクステンションに割り当てられます。 これはデフォルト値です（Com-plete スレッドエクステンションを使用）。
NO	Natural スレッドは、16 MB 境界より下の Com-plete の物理的なスレッドに割り当てられます。

### TTY xx - TTY デバイス制御文字

このパラメータは、テレタイプライタ（TTY）デバイス制御文字を設定します。次の 16 進値を設定することができます。

値	説明：
TTYCR=0D	TTY 改行
TTYLF=15	TTY 改行
TTYIC=00	TTY アイドル文字
TTYNIC=00	TTY アイドル文字数
TTYBS=16	TTY バックスペース
TTYAL=07	TTY アラーム

## UCTRAN - Com-plete/SMARTS エラーメッセージの小文字から大文字への変換

このパラメータは、Com-plete/SMARTS エラーメッセージの小文字から大文字への変換を制御します。

可能な値は次のとおりです。

値	説明：
YES	大文字への変換が有効になります。
NO	大文字への変換が無効になります。 これはデフォルト値です。

## U2PRINT - プリンタのダイナミックアロケーション

このパラメータは、ハードコピー要求に対する Com-plete によるプリンタのダイナミックアロケーション機能を制御します。

可能な値は次のとおりです。

値	説明：
YES	Natural は、プリンタの出力先を指定するためにハードコピー要求である Com-plete の U2PRINT ルーチン呼び出します。
NO	ハードコピープリンタのダイナミックアロケーションを無効にします。Natural は、Natural プロファイルパラメータ HCDEST からデフォルト値を使用します。 これはデフォルト値です。

## 異常終了出口の使用

ABEXIT 出口は、一般的に NCFPARM で SPIEA=NO を設定することによって非アクティブにすることができます。

ABEXIT 出口は、プログラムの異常終了処理に対する Com-plete の EOJ 処理時に呼び出されます。

デフォルトでは、OCX 異常終了は ABEXIT 出口ルーチンによって解釈されます。

- DU=ON/SNAP/ABEND で実行すると、Natural セッションはダンプされて正しく終了し、エラーメッセージ NAT9974 が出力されます。
- DU=FORCE で実行すると、ABEXIT 出口ルーチンは無効になり、Com-plete 時の即時のダンプが生成されます。

- NFMPRM マクロで LE370=YES を指定して、LE プログラムに制御があるときに異常終了が発生すると、ユーザーが記述した条件ハンドラまたは言語固有の条件ハンドラは無視されます。異常終了は ABEXIT 出口ルーチンによって処理され、Natural エラーメッセージ NAT0950 または NAT9967 が発生します。

DU=OFF の場合は、Natural はエラーメッセージ NAT0950、NAT0954、NAT0955、または NAT0956 を出力し、異常終了 PSW 全体およびレジスタ 0~15 はオフセット x'290' の IOCB に含まれません。

## ストレージの使用

---

セッションの初期化時に、NCFPARM のパラメータ NTHSIZE で定義されたスペースの合計は、パラメータ THABOVE に基づいて 16 MB 境界の上または下にスレッド GETMAIN として割り当てられ、Natural に使用されます。

Natural プロファイルパラメータ WPSIZE は、ワークプール上下のサイズを決定します。デフォルトでは、サブプール下のサイズは 32 KB に設定されています。

このため、Natural Com-plete フロント部を Com-plete ユーティリティ ULIB、RG size = 36 KB 以上でカタログする必要があります。

上下の Com-plete スレッド部内の残りのエリア (Com-plete ULIB RG=*specification* および/または THABOVESIZE=*specification*) は、Com-plete によって以下の目的で使用されます。

- ユーザーサブルーチン
- Natural スレッド内の可変バッファの増加
- 「物理的な」 GETMAIN 要求を行うサブプロダクト。これにより Natural ワークプールアロケーションを実行します。

ULIB RG および THABOVESIZE パラメータの詳細については、『*Com-plete Utilities*』ドキュメントを参照してください。

## バックエンドプログラムのサポート

---

Natural は、次の文字列をバックエンドプログラムに渡します。

- Natural リターンコード (フルワード)
- Natural 終了メッセージ (A72)
- 終了エリアの長さ (フルワード)
- 終了データ



この文字列は NAMBCKP マクロによってマップされます。

XNCFBACK ソースモジュールは、Com-plete 環境における Natural バックエンドプログラムの一例です。これはリエントラントコードとして記述され、RESIDENTPAGE プログラムとしてロードするか、各ユーザーが一度だけロードすることができます。

## Natural バッチ実行における Com-plete のサポート

Natural バッチ実行において Com-plete サービスを使用すると、バッチ実行の終了時にバッチユーザー ID はログオンされたままになります。

この状況を回避するには、Com-plete 配布ライブラリの COMPBTCH モジュールをバッチ Natural ニュークリアスに含めます。これにより、モジュール EOJ のエントリポイントが解決されます。このモジュールは、終了時のクリーンアップのために Natural バッチジョブの終了時に呼び出されます。

NCFAM モジュールは Com-plete 出力/ワークファイルにアクセスするために使用されます。このモジュールは、Com-plete 配布ライブラリの COMPBTCH モジュールとともに Natural ニュークリアスのリンクに含める必要があります。

## Com-plete/SMARTS 環境での Natural の非同期処理

Natural の非同期処理については、Natural の『オペレーション』ドキュメントの「非同期処理」で説明しています。しかし、Natural を Com-plete 環境で実行するときには、いくつか追加の考慮事項があります。

適切な SENDER および OUTDEST の宛先が非同期 Natural セッションに対して指定されていることを確認してください。指定されていないと、出力のために異常終了します。

SENDER および OUTDEST の Com-plete 端末 ID に加えて、Natural Com-plete/SMARTS インターフェイスでは、次のキーワードがサポートされます。

キーワード：	説明：
DUMMY	出力は無視されます。
CONSOLE	出力はオペレータコンソール/Com-plete ログファイルにルーティングされます。

デフォルトでは、3270 データストリームプロトコルが、Com-plete 環境で実行されている非同期 Natural セッションの出力に使用されます。

Com-plete 環境で非同期 Natural トランザクションを開始する例は、SYSEXTP ライブラリの ASYNCOMP プログラムにあります。

## ユーザープログラムからの Natural の起動

---

Com-plete `FETCH` 機能を使用して、Com-plete 環境でユーザーのフロントエンドプログラムから Natural を起動することができます。詳細については、『*Com-plete Application Programmer's Manual*』を参照してください。

## ストレージスレッドキーの処理

---

Com-plete とアプリケーションプログラムの間で保護モードを使用する場合は、Natural パラメータモジュール `NATPARM` でプロファイルパラメータ `SKEY=OFF` を設定する必要があります。アプリケーションプログラムは、対応するスレッドキーで実行されます。Natural または Editor のバッファプール呼び出しに対しては、フロントエンドドライバは適切なキーに切り替えて、呼び出し後に元のスレッドキーに切り替えます。

使用しているマシンで `Storage-Protection Override` 機能をアクティブにすることによって、Com-plete 環境でアプリケーションプログラムのパフォーマンスを大幅に向上させることができます。

Natural サブグループに対して、Com-plete 起動パラメータ `THREAD-GROUP` でスレッドキー = 9 に設定します。

スレッドキーが 9 の場合、フロントエンドドライバは Natural アプリケーションを自動的に特権モードに設定し、ファンクションコード `THRD/TCS` で Com-plete 関数 `MODIFY` を使用する代わりに、キースイッチ処理に `SPKA` 命令を使用します。

## セッション初期化中のユーザー出口処理のサポート

---

セッション初期化時に、ユーザー出口のアクティブ化に関するユーザー固有のセッション情報を Natural に渡すことができます。出口は、ドライバ/IOCB の初期化が完了した後で、Natural が初期化される前に呼び出されます。

ドライバは、パラメータとしてレジスタ 1 の IOCB のアドレスを渡しますが、出口は Com-plete 関数 `COLOAD/CODEL` によってアクティブ/非アクティブにされます。詳細については、『*Com-plete Application Programmer's Manual*』を参照してください。

`NCFUEXIT` ソースモジュールは、ユーザー出口の一例です。ユーザー出口は、パラメータモジュール `NCFPARM` で定義することができます。

## SMARTS サーバー環境の使用

SMARTS サーバー環境では、ネイティブファイルシステムのデータセットに加えて、SMARTS ポータブルファイルシステムを入出力ファイルのコンテナとして使用することができます。環境変数がポータブルファイルシステムを参照するか、ネイティブファイルシステムを参照するかは、SMARTS パラメータ `CDI_DRIVER` および `MOUNT_FS` の設定に依存しています。詳細については、『*SMARTS Installation and Operations Manual*』を参照してください。

環境変数が定義されていない場合は、Natural の『オペレーション』ドキュメントの「z/OS バッチモードの *Natural* で使用されるデータセット」に説明されているように、通常のデータセットにアクセスします。

以下では次のトピックについて説明します。

- 入力／出力
- 出力ファイル／ワークファイル

### 入力／出力

SMARTS サーバー環境での入出力は、DLL `NCF42I0` によって実行されます。

`NCF42I0` は常駐域にロードする必要があります。`NCF42I0` がアプリケーションプログラムスレッドにロードされると、Natural セッションは終了して `NAT9980` エラーメッセージが表示されます。

サポートされている環境変数：

- `CMPRINT` - 主なレポートの出力
- `CMSYNIN` - 主なコマンド入力
- `CMOBJIN` - **Natural INPUT** ステートメントの入力

これらの環境変数について、以下に説明します。

## CMPRINT - 主なレポートの出力

構文：

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

上記の意味は次に示すとおりです。

<i>pathname</i>	出力ファイルの場所を指定します。  <i>pathname</i> がポータブルファイルシステムを参照している場合は、パスが作成されます。ネイティブデータセットを参照している場合は、ネイティブデータセットを利用する必要があります。
<i>filename</i>	出力ファイルの名前を指定します。  ファイル名にアスタリスク (*) を指定すると、ファイル名は実際のユーザー ID から生成されます。  <i>pathname</i> がネイティブファイルシステムを参照し、 <i>filename</i> がスラッシュ文字 (/) で終わっている場合は、シーケンシャルデータセット <i>pathname/filename</i> にアクセスします。「/」で終わっていない場合は、データセット <i>pathname</i> のメンバ <i>filename</i> にアクセスします。
<i>mode</i>	fopen() 関数について C ライブラリに文書化されているように、ファイルモードを指定します。デフォルトは w (書き込み) です。

例えば、/fs/ はネイティブファイルシステムにマップされ、/pfs/ はポータブルファイルシステムにマップされていると想定します。

CMPRINT=/fs/natural/test/print	データセット <i>natural.test</i> のメンバ <b>print</b> にアクセスします。
CMPRINT=/fs/natural/test/print/	シーケンシャルデータセット <i>natural.test.print</i> にアクセスします。
CMPRINT=/pfs/natural/test/print	ポータブルファイルシステムの / <i>natural/test</i> のメンバ <b>print</b> にアクセスします。

## CMSYNIN - 主なコマンド入力

構文：

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

適切なコマンド入力ファイルの *pathname* および *filename* を指定します。

パス名がネイティブファイルシステムを参照し、ファイル名が「/」文字で終わっている場合は、シーケンシャルデータセット *pathname/filename* にアクセスします。スラッシュ (/) で終わっていない場合は、データセット *pathname* のメンバ *filename* にアクセスします。

## CMOBJIN - Natural INPUT ステートメントの入力

構文：

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

適切なデータ入力ファイルの *pathname* および *filename* を指定します。

*pathname* がネイティブファイルシステムを参照し、*filename* がスラッシュ文字 (/) で終わっている場合は、シーケンシャルデータセット *pathname/filename* にアクセスします。スラッシュ (/) で終わっていない場合は、データセット *pathname* のメンバ *filename* にアクセスします。

## 出力ファイル／ワークファイル

SMARTS サーバー環境での出力ファイルおよびワークファイルへのアクセスは、DLL NCF42APS によって実行されます。

NCF42APS は常駐域にロードする必要があります。NCF42APS がアプリケーションプログラムスレッドにロードされると、Natural セッションは終了して NAT9980 エラーメッセージが表示されます。

サポートされている環境変数：

- NAT\_PRINT\_ROOT - PFS またはネイティブファイルシステムのプリンタファイルへのパス。
- NAT\_WORK\_ROOT - PFS またはネイティブファイルシステムのワークファイルへのパス。

構文例：

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

上記の意味は次に示すとおりです。

<i>qualifier</i>	SMARTS ポータブルファイルシステムにアクセスするか、ネイティブの OS で管理されているファイルシステムにアクセスするかを決定します。
<i>path1/path2</i>	適切なファイルシステムのファイルの場所へのパスです。

## Com-plete のリカバリ可能なセッション処理のサポート

z/OS 環境で利用できる Com-plete のリカバリ可能なセッション処理を活用するには、モジュール NCFROLLS をフロントエンドモジュールにリンクする必要があります。NCFROLLS は Natural ロールサーバーへのインターフェイスとして機能します。リカバリ可能セッションをサポートするには、Natural ロールサーバーを開始する必要があります。さらに、RRS インターフェイスモジュールは Com-plete ルーチン TLOPUSER の一部であるため、モジュール ATRRCSS をフロントエンドモジュールにリンクする必要はありません。会話型の端末 I/O を実行する場合は、Com-plete の再起動後に Natural セッションを再開できるようにするため、Natural スレッドは Natural ロールファイルに圧縮形式で書き込まれます。非会話型の端末 I/O およびスレッドがロックされたアプリケーションについては、Natural スレッドは Natural ロールファイルに書き込まれません。そのため、これらのセッションはリカバリすることができません。

# 21 IMS/TM 環境の Natural - 概要

---

このドキュメントでは、Natural IMS/TM インターフェイス（製品コード NII）の機能、および IMS/TM 環境での Natural の操作および個別のコンポーネントについて説明します。

- 環境
- コンポーネント
- コンフィグレーションマクロ
- サービスプログラム
- サービスモジュール
- ユーザー出口
- 特殊な関数
- リカバリ処理

以下の項目も参照してください。

- インストール - Natural の『インストール』ドキュメントの「*Natural* IMS インターフェイスのインストール」を参照してください。
- エラーコード - Natural IMS インターフェイス（NII）によって発行されるエラーコードおよびメッセージのリストについては、Natural の『メッセージおよびコード』ドキュメントの「IMS/TM 環境下の *Natural* のエラーコード」を参照してください。





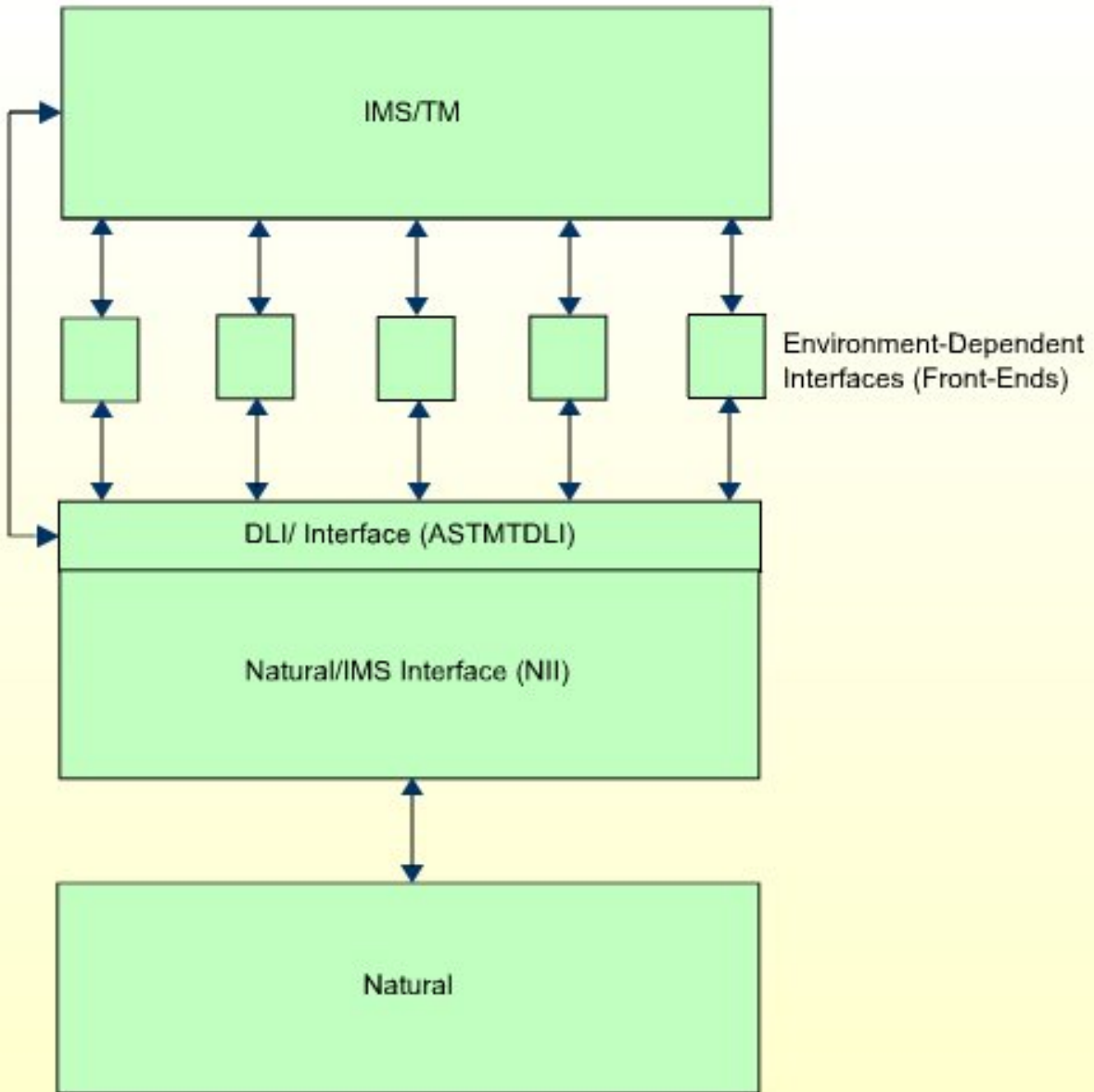
## 22 IMS/TM 環境の Natural - 環境

---

■ IMS/TM インターフェイスの概要 .....	136
■ IMS/TM 環境 .....	137
■ ダイアログ指向の環境 .....	138
■ メッセージ指向の環境 .....	140
■ バッチメッセージ処理環境 .....	142
■ Natural WRITE (n) ステートメントのサポート .....	143
■ SET CONTROL 'N' (端末コマンド %N) .....	146
■ IMS/TM における Natural メッセージでの TS=ON のサポート .....	146
■ SENDER の宛先 .....	146
■ Natural プロファイルパラメータ PROGRAM のサポート .....	147

この章では、Natural をさまざまな IMS/TM 環境で実行する方法について説明します。

## IMS/TM インターフェ이스の概要



## IMS/TM 環境

---

IMS/TM では異なる 3 種類の環境を提供します。

- **メッセージ処理リージョンの Natural (MPP 環境)**
- **バッチメッセージ処理リージョンの Natural (BMP 環境)**
- **オフラインの DL/I バッチリージョンの Natural**

これらの各環境で Natural を使用するために、Natural IMS インターフェイスにはそれぞれの環境固有のインターフェイスが提供されています。このようなインターフェイスの役割は、環境から入力（通常は端末入力メッセージ）を受信すること、受信した入力を処理するために Natural に渡すこと、および得られた出力を正しい宛先に（通常は端末出力メッセージ）返すことです。このようにして、利用可能なすべての IMS 環境で Natural の機能を使用することができます。

異なるさまざまな環境で利用できるだけでなく、各環境では操作方法が異なります。

### メッセージ処理リージョンの Natural (MPP 環境)

メッセージ処理リージョンでは、Natural オンライントランザクションは、以下のいずれかになります。

- **ダイアログ指向の Natural**
- **メッセージ指向の Natural**

#### ダイアログ指向の Natural

ダイアログ指向の Natural セッションは、IMS 画面との進行中の対話を確立します。Natural との入出力メッセージは論理的に関連付けられ、ダイアログ手順全体を通じて、Natural は次の入力メッセージを正しく処理できるように情報を保存します。ダイアログ指向の方法では、Natural は会話型トランザクションまたは非会話型トランザクションのいずれかとして実行することができます。

ダイアログ指向の環境では、Natural は、入力待ち (Wait-for-Input、WFI) トランザクションとして、および並列スケジュールオプションを指定して、複数のメッセージ処理リージョンで実行することができます。

ダイアログ指向環境で Natural を実行するには、ロールサーバーまたはロールファイルを使用する必要があります（「[ロールファイルおよびロールサーバー](#)」を参照）。

Natural IMS インターフェイスがエラー状態を検出すると、このエラー状態に関する情報を含むレコードが IMS ログファイルに書き込まれます（「[リカバリ処理](#)」を参照）。したがって、Natural が実行されるすべての端末およびすべての Natural トランザクションコードは、Automated Operator Interface (AOI) を使用する /LOG コマンドの発行を認可されている必要があります。

## メッセージ指向の Natural

メッセージ指向の Natural セッションは、IMS メッセージキューからの 3270 形式以外のメッセージを処理します。入力メッセージは互いに無関係であると見なされます。またダイアログの一部ではありません。メッセージ指向の方法では、Natural は非会話型トランザクションとして実行する必要があります。

## バッチメッセージ処理リージョンの Natural (BMP 環境)

バッチメッセージ処理リージョンでは、Natural は入力トランザクションコードを使用することによって IMS メッセージキューにアクセスすることができます。バッチ指向の BMP リージョンでは、Natural はシンボリックチェックポイントおよび拡張再スタートをサポートしています。入力メッセージの形式は 3270 以外の形式です。

## オフラインの DL/I バッチリージョンの Natural

BMP Natural はオフラインの DL/I バッチジョブとしても実行することができます。

IOPCB を利用できない場合は、すべての END TRANSACTION および BACKOUT TRANSACTION ステートメントが無視されます。

診断の目的で次の機能を利用することができます。プロファイルパラメータ TPF=(1) を指定して Natural が起動された場合、上記の事実を示す非公式の WTO メッセージが発行されます。

## ダイアログ指向の環境

このセクションでは、ダイアログ指向の会話型環境にのみ有効な特別な点について説明します。

- [会話型環境の特別な考慮事項](#)
- [非会話型環境の特別な考慮事項](#)
- [MSC 環境の特別な考慮事項](#)

### 会話型環境の特別な考慮事項

ダイアログ指向の会話型環境は、会話型 MPP インターフェイスによって実装されます。このインターフェイスは、Natural パラメータモジュールによって会話型 MPP フロントエンドにリンクされています。このフロントエンドは IMS/TM アプリケーションプログラムで、割り当てられたトランザクションコードの入力メッセージが IMS/TM メッセージキューで利用できる場合に、IMS/TM によってスケジュールされます。

ダイアログ指向の会話型環境では、少なくとも 157 バイトに NRSTART 値を加えたスクラッチパッドエリア (SPA) が必要です。

## 非会話型環境の特別な考慮事項

ダイアログ指向の非会話型環境は、非会話型の MPP インターフェイスによって実装されます。このインターフェイスは、Natural パラメータモジュールによって非会話型 MPP フロントエンドにリンクされています。このフロントエンドは IMS/TM アプリケーションプログラムで、割り当てられたトランザクションコードの入力メッセージが IMS/TM メッセージキューで利用できる場合に、IMS/TM によってスケジュールされます。

ダイアログ指向の非会話型環境を使用する場合は、SIP 機能を有効にした *Natural* 認可サービスマネージャおよび物理的な入力編集ルーチンが前提条件となります。

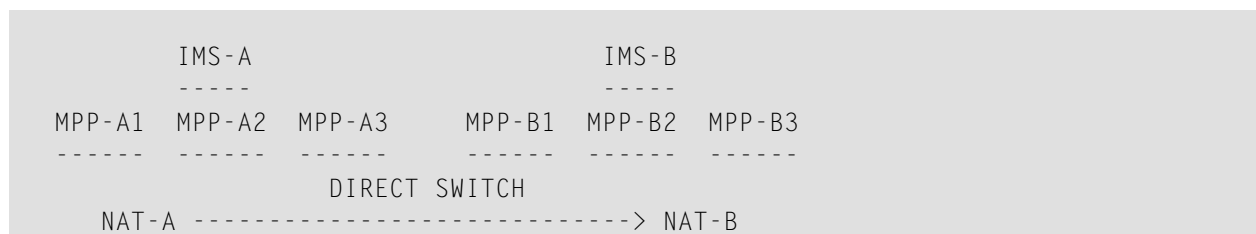
- *Natural* 認可サービスマネージャは、IMS/TM SPA をシミュレートするために使用します。
- 物理的な入力編集ルーチンは、入力メッセージの前にトランザクションコードを挿入するために使用します。

以下では、同じ Natural サブシステム ID を指定する必要があります。

- NIMPARM マクロの SPATID
- NIMPIXT マクロの SPATID
- 認可サービスマネージャの起動パラメータ

## MSC 環境の特別な考慮事項

次の環境を想定して、Natural IMS インターフェイスは NAT-B に対してメッセージ X'000500006D' を用意します。これは端末ユーザーが CLEAR キーを押したことを意味します。



トランザクションコードテーブルに、2つのエントリを作成する必要があります。1つ目のエントリは NAT-A、2つ目のエントリは NAT-B です。

これらの2つのエントリは Natural Reserved Area (NRA) に対して異なるオフセットを指定する必要があります、またこれらのエリアがオーバーラップしないようにする必要があります。

NAT-B は、Natural セッションが通常の方法で IMS-B で開始されることを検出し、セッション開始出口ルーチンに制御を与えます。セッション開始出口ルーチンは入力メッセージで文字列 X'000500006D' をチェックして、Natural から見た入力メッセージの長さを 0 に設定します。

追加のロジックが出口 NIIXSTAR または出口 NIIXSSTA のいずれかで提供されない場合は、Natural は IMS-B で新しいユーザーセッションを開始します。

IMS-A および IMS-B は、Natural に対して異なる専用ルールファイルが割り当てられていると想定します。

両方（またはそれ以上）の Natural セッションは、直接プログラム間の切り替えを実行するときに SPA でデータを転送することによって、互いに通信することができます。

しばらくの間、このような環境に複数の Natural セッションが存在する場合は、「アクティブ」なセッションのみが正しく終了します。

## メッセージ指向の環境

---

このセクションでは、IMS/TM 環境の Natural で使用するメッセージ指向のインターフェイスについて説明します。

- [メッセージ指向の環境について](#)
- [メッセージ指向環境の操作](#)

### メッセージ指向の環境について

このインターフェイスは、純粋なデータ入力メッセージを処理するように設計されています。つまり、メッセージは 3270 データストリームを示していません。メッセージ指向のインターフェイスは、ユーザーが記述した Natural プログラムによって駆動されます。このプログラムが、入力メッセージの取得を目的として IMS メッセージキューにアクセスするようにインターフェイスに指示します。

メッセージ指向のインターフェイスが作成された目的は、非会話型で非端末駆動のトランザクションを、非会話型の MPP トランザクションとして実行できるようにすることです。

### メッセージ指向環境の操作

メッセージ指向のインターフェイスには、MPP および BMP インターフェイスの両方から機能が組み込まれています。要求される処理の多くは BMP タイプのトランザクションをエミュレートするため、BMP インターフェイスが基盤として使用されます。

メッセージ指向のインターフェイスは端末指向ではないため、メッセージまたは画面イメージが自動的に生成されて端末に送信されることはありません。Natural ニュークリアスにはバッチ環境で実行されていることが通知されるため、出力はプリンタ出力と解釈され、入力は CMSYNIN ファイルからの入力であると想定されます。通常 CPRINT に書き込まれるすべての出力は、Natural プロファイルパラメータ SENDER で指定された IMS/TM の宛先に送信されます。詳細については、以下の「[SENDER の宛先](#)」を参照してください。

Natural が入力データの取得を試みたが、入力データが STACK コマンドを介してアプリケーションから提供されていない場合、EOF は入力が存在しないことを示し、Natural は終了します。

サービスモジュール CMSNFPRT を使用することによって、ランタイム時に SENDER を新しい値に設定することができます。

チェックポイント処理を除いて、DL/I 用の Natural および DB2 用の Natural での処理は、BMP モードと同様になります。これが必要な理由は、物理的な 1 つのスケジューリングによって複数の無関係な入力メッセージが処理される可能性があるからです（ほとんどの場合、こうなります）。会話型の MPP インターフェイス環境では、1 つの Natural セッション中に処理されるすべてのトランザクション、およびこの Natural セッション内のすべての DL/I 要求は関連していると見なされ、データベースの位置および PCB 使用のメンテナンスが必要となります。非会話型のインターフェイスでは、この DL/I 用の Natural のロジックは適用されません。

1 つのスケジューリング（また 1 つの Natural セッション）中に処理されるトランザクションは互いに関連していないため、ロールファイルの Natural セッション情報を保持する必要はありません。このため、このインターフェイスにはロールデータセットを割り当てる必要はありません。ロールスロットエリアは GETMAIN を介して割り当てられ、すべての Natural コントロールブロックおよびワークエリアの格納に使用されます。

処理はメッセージごとに実行されるので、再配置ロジックは必要ありません。

メッセージ指向のインターフェイスでは、メッセージキューからのすべてのメッセージの取得はフロントエンドの Natural プログラムによって開始されます。ユーザー固有の処理要件を満たすために、このプログラムはユーザーが記述する必要があります。しかし、次に示すような特定の構造も必要です。

```
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
```

サービスモジュール CMGETMSG は、最初のメッセージセグメントを読み込みます。サービスモジュール CMGETSEG は、他のすべてのメッセージセグメントを読み込みます。

Natural は CMSYNIN からの入力を読み込むことができないため、Natural スタックを入力に使用する必要があります。この処理は、Natural プロファイルパラメータ STACK を使用して行われます。

Naturalを終了する前にアプリケーションがIMSメッセージキューにアクセスする処理は、ユーザーが行います。これを行わないと、Naturalトランザクションは異常終了し、メッセージキューに対するGUが実行されていないことを示すIMSアベンドコード462が表示されます。

異常終了の場合でもこれらのNaturalメッセージを取得するには、最初の代替PCBをEXPRESS PCBとして定義することをお勧めします。

メッセージ指向の環境は、NTRD インターフェイスによって実装されます。このインターフェイスは、NaturalパラメータモジュールによってNTRDフロントエンドにリンクされています。このフロントエンドは、直接IMS/TMによって、または**NIMBOOT** マクロで生成されたブートストラップモジュールを介して呼び出すことができます。

IMS/TMによって直接呼び出される場合、このフロントエンドはIMS/TMによってスケジューラされるIMS/TMアプリケーションプログラムです（割り当てられたトランザクションコードに対する入力メッセージをIMS/TMメッセージキューで利用できる場合）。必要なSTACKパラメータを含むNaturalプロファイルを使用することをお勧めします。NaturalパラメータモジュールでPROFILE=PROGRAMを指定し、インターフェイスが起動されたトランザクションコードと同じ名前で作成します。こうすることで、使用されるトランザクションコードごとに異なるSTACKを持つ異なるプロファイルを使用する柔軟性を得ることができます。

ブートストラップモジュールを介して呼び出す場合、このブートストラップモジュールはIMS/TMによってスケジューラされるIMS/TMアプリケーションプログラムです（割り当てられたトランザクションコードに対する入力メッセージをIMS/TMメッセージキューで利用できる場合）。このブートストラップモジュールは、一連のダイナミックプロファイルパラメータを提供します。その1つはSTACKプロファイルパラメータで、このブートストラップモジュールの生成中に名前が指定されるNTRDフロントエンドを呼び出します。多様なダイナミックプロファイルパラメータ設定でNaturalを呼び出す場合は、さまざまなブートストラップモジュールを生成し、それぞれに独自のダイナミックプロファイルパラメータ文字列を使用する必要があります。これらのブートストラップモジュールは、それぞれ一意の名前でリンクされている必要があります。また、得られる各ロードモジュールに一意のIMS/TMトランザクションコードを割り当てる必要があります。

## バッチメッセージ処理環境

---

バッチメッセージ処理（BMP）環境は、BMP インターフェイスによって実装されます。このインターフェイスは、Naturalパラメータモジュールおよびワークファイル/出力ファイルアクセスルーチンNATWKFOによって、BMPフロントエンドにリンクされています。このフロントエンドは、BMP JCLで指定されたIMS/TMアプリケーションプログラムです。

標準のNaturalバッチは、バッチメッセージ処理リージョンで実行されます。標準のNaturalバッチ実行と比較して、オプションの入力データセットCONTROLを使用することができます。

オプションのBMP CONTROL ファイルには、最大2つの入力カードが含まれています。



- 1つ目の入力カードには、次のキーワードパラメータが含まれています。

キーワード	意味
ENV-TAB=	使用する環境テーブルの名前。
TRNCODE=	使用するトランザクションコードの名前。TRNCODE パラメータの説明を参照してください。

例：

```
ENV-TAB=ENVBMP0 TRNCODE=NATIMS
```

- CONTROL ファイルの2つ目の入力カードには、ダイナミックな Natural パラメータが含まれています。

ダイナミックな Natural パラメータを渡すための **CMPRMIN** データセットと **CONTROL** ファイルの使用

ダイナミックな Natural パラメータを渡すために CMPRMIN データセットも使用する場合は、CONTROL の入力に CMPRMIN の入力に追加されます。これは、CONTROL で指定されたパラメータが CMPRMIN で指定されたパラメータを上書きすることを意味します。

**CONTROL** ファイルを使わない作業

CONTROL ファイルを使用しない場合、環境テーブルの名前は使用されるトランザクションコード（トランザクション指向の BMP）、または使用される PSB 名（バッチ指向の BMP）に対応するトランザクションコードテーブルのエントリによって決定されます。

## Natural WRITE (n) ステートメントのサポート

WRITE (n) ステートメントを使用すると、異なるプリンタで最大 31 の異なるレポートを、同じ Natural プログラム内に作成することができます。レポートは、Natural パラメータモジュールで、または Natural DEFINE PRINTER (n) ステートメントを使用して指定された IMS 端末に送信されます。レポートを制御する NTPRINT マクロで AM=IMS を指定する必要があります。

このステートメントを使用するには、同じ Natural プログラム内に作成する並列レポートの数と同数の追加の代替 TP-PCB を PSB に定義し、WRKPCB パラメータを使用してトランザクションコードテーブルで追加の代替 TP-PCB の数を指定します。

- ❗ **注意:** 1つ目の代替 TP-PCB は Natural IMS インターフェイスによって使用されることに注意してください。

ダイアログ指向の環境で WRITE (n) ステートメントを使用するときは、次の制限が適用されません。

レポートの生成は、1つ以上の画面 I/O にまたがって実行することはできません。画面 I/O 後に同じプリンタを使用する場合は、CLOSE PRINTER (n) ステートメントを使用して、画面 I/O の前にプリンタを明示的に閉じる必要があります。

レポートの作成には、NTPRINT マクロの次のキーワードサブパラメータが関連しています。

パラメータ	意味
AM	「IMS」 に設定する必要があります。
DEST	IMS/TM の宛先を指定します。
BLKSIZE	宛先に送信されるバッファのサイズを指定します。レポート行がバッファされます。
DRIVER	レポートの作成に使用するドライバを指定します。利用できる値のリストについては、Natural/IMS インターフェイスパラメータモジュールの <b>PRTDRIV</b> パラメータを参照してください。ドライバは、フォームフィードの場所（レポートの開始位置、終了位置、開始位置と終了位置の両方、またはフォームフィードなし）、ページの開始位置（1行目、または NII 2.2 との互換性のために 2 行目）、およびレポートの出力場所（SCS プリンタまたは非 SCS プリンタ）を決定します。さらに、JES API を使用することを指定できます。
NAME FORMS DISP COPIES CLASS PRTY	これらのパラメータは、JES API を使用する場合のみ評価されます。

## NTPRINT および CLOSE PRINTER に関するヒント

### NTPRINT の設定

IMS/TM プリンタ（つまり、AM=STD で定義されたプリンタ）の NTPRINT マクロまたは PRINT プロファイルパラメータ定義の OPEN および CLOSE のサブパラメータには常にデフォルトを使用することを強くお勧めします。つまり、OPEN および CLOSE に対して値を指定しないか、またはデフォルトの OPEN=ACC および CLOSE=CMD を使用してください。

これは、OPEN および CLOSE の他のオプションを使用して、異なるアクセス方式に対して NATPARM に静的にプリンタを定義した場合、および AM=IMS でアクセス方式をダイナミックに上書きする場合に特に重要です。この場合、常に AM=IMS、OPEN=ACC、CLOSE=CMD を同時に指定してください。



**注意:** アクセス方式が上書きされても、NTPRINT オプションはダイナミックに指定された PRINT オプションとマージされます。

非デフォルト値を使用すると発生する可能性がある問題

1. OPEN=OBJ を使用すると、OUTPUT オプションを DEFINE PRINTER ステートメントで指定している場合に、間違った出力先に出力されたり、NAT8211 エラーが発生する場合があります。  
OPEN=OBJ を指定すると、OUTPUT の上書きが評価される前にプリンタが開かれ、使用されるプ

リンタの出力先は OUTPUT オプションで指定された場所ではなく、PRINT パラメータで指定された場所になります。

2. CLOSE=FIN を指定すると、プリンタは CLOSE PRINTER 時ではなく FIN 時に閉じられます。これは、GU がメッセージキューに発行され、宛先が TP PCB でリセットされた後に、CLOSE が来る場合があることを意味します。このために、IMS/TM ステータスコード QF (MPP) または A3 (BMP および OBMP/NTRD) に対して NII エラー NII3641 が発生します。CLOSE=CMD を指定すると、プリンタは CLOSE PRINTER ステートメントで実際に閉じられます。

## CLOSE PRINTER または DEFINE PRINTER の使用

IMS/TM プリンタに書き込まれたレポートは、次の GU 呼び出しで（つまり端末 I/O 時または CMGETMSG を介して）、IMS/TM によって暗黙的に閉じられます。つまり、IMS/TM はプログラムの CLOSE PRINTER または DEFINE PRINTER ステートメントに関係なくレポートを出力します。

Natural に対してはプリンタは開いた状態で、同じレポート番号を持つ次の WRITE ステートメントがすでに出力されたレポートを続行するため、NAT1518 エラーが発生します。


シナリオ：

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

INPUT/CMGETMSG は「物理的」にプリンタを閉じ、IMS/TM は行 'line 1' を含むレポートを出力します。

プリンタは Natural に対して「論理的には」開いているために、行 'line 2' は新しいレポートを開始せず、宛先が GU 呼び出しによって削除されるためにエラー NAT1518 が発生します。

このため、次のルールを守ることを強くお勧めします。

-  **注意:** GU の後に同じ番号のレポートが継続された場合は CLOSE PRINTER ステートメントが必要です。

DEFINE PRINTER ステートメントは暗黙の終了を実行します。この場合、CLOSE PRINTER ステートメントは使用されません。例えば次のようになります。

正しいステートメント	正しいステートメント	誤ったステートメント (NAT1518)
<pre>REPEAT   DEFINE PRINTER (1)   WRITE (1)    INPUT LOOP</pre>	<pre>DEFINE PRINTER (1) REPEAT   WRITE(1)   CLOSE PRINTER (1)   INPUT LOOP</pre>	<pre>DEFINE PRINTER (1) REPEAT   WRITE(1)    INPUT LOOP</pre>

## SET CONTROL 'N' (端末コマンド %N)

ステートメント SET CONTROL 'N' (端末コマンド %N) は IMS/TM 環境では適用されません。IMS/TM 環境で使用すると、次の論理出力画面が妨げられます。

## IMS/TM における Natural メッセージでの TS=ON のサポート

TS=ON が Natural セッションで指定されると、IMS/TM における Natural メッセージはすべて大文字に変換されます。

## SENDER の宛先

メッセージ指向 (NTRD) およびサーバー (SRVD) 環境において、通常は CPRINT に書き込まれるすべての出力は、Natural プロファイルパラメータ SENDER で指定された宛先に送信されます。SENDER を指定する場合は、出力が IMS メッセージキューを介して送信される有効な IMS/TM の宛先 (通常は LTERM) を指定するか、または次の予約値のいずれかを指定します。

値	意味
*WTO	Natural 出力は宛先コード 11 (プログラマ情報) で WTO を使用して、ジョブログに送信されます。
*MTO	Natural 出力は /BROADCAST MASTER コマンドを使用して、IMS マスタコンソールに送信されます。
*PRINTnn	Natural 出力は Natural プリンタ nn に書き込まれます。つまり、出力は内部の WRITE(nn) を使用して書き込まれます。  出力ファイルは各出力行後に閉じられ、各出力行の改行制御文字は空白になります。



**注意:**

1. /BROADCAST MASTER コマンドが失敗すると（例えば認証問題のため）、WTO を使用してエラーメッセージが発行され、現在のメッセージを含むすべての Natural メッセージがジョブログに送信されます。つまり、SENDER の宛先は内部的に \*WTO に設定されます。
2. \*PRINT *nn* の場合には、AM=STD で定義されたプリンタのみを使用することを強くお勧めします。
3. Natural プロファイルパラメータモジュールで SENDER の宛先をコード化することを強くお勧めします。これにより、Natural の初期化が失敗した場合でも（Adabas エラー NAT3048 または NAT3148 などのため）、宛先は見つかります。
4. SENDER の宛先をダイナミックに指定する必要がある場合は、予約値を一重引用符で囲む必要があります（例えば SENDER='\*WTO'）。

## Natural プロファイルパラメータ PROGRAM のサポート

Natural プロファイルパラメータ PROGRAM は、ダイアログ指向の環境および BMP 環境でサポートされています。

BMP 環境では、パラメータ PROGRAM は、標準の z/OS バッチ環境と同じように動作します。XNIIBACK という名前を使った例が NII *vrs.* SRCE データセットで提供されています。起動されたバックエンドプログラムは Natural IMS インターフェイスに戻るようになっています。

ダイアログ指向の環境では、Natural プロファイルパラメータ PROGRAM は、BMP 環境を含む他の Natural 環境とは動作が少し異なります。

1. プロファイルパラメータ PROGRAM または Natural サブプログラム CMPGMSET で指定される名前は、IMS トランザクションコードの名前です。
2. Natural セッションがエラーなく終了した場合のみ、指定されたトランザクションコードが起動します。
3. TERMINATE ステートメントで提供されるデータは、起動された IMS トランザクションに入力メッセージとして渡されます。



# 23

## IMS/TM 環境での Natural - コンポーネント

---

■ フロントエンドモジュール .....	150
■ Natural IMS インターフェイスモジュール NIIINTFM .....	151
■ 物理的な入力編集ルーチン .....	152
■ ユーザーメッセージテーブル DFSCMTU0 .....	152
■ ロールファイルおよびロールサーバー .....	153
■ 認可サービスマネージャ .....	154
■ Natural 共有ニュークリアス .....	155
■ Natural バッファプール .....	155
■ Adabas インターフェイス .....	155
■ プリロードリスト .....	156

Natural IMS インターフェイスドキュメントのこの章では、Natural IMS インターフェイスのコンポーネントについて説明します。

## フロントエンドモジュール

---

フロントエンドモジュールは、CALL インターフェイス **NIIBOOTS** によって呼び出されるサーバー環境を除き、IMS/TM プログラムコントローラ **DFSPPC20** から制御を受け取ります。

フロントエンドモジュールはインストールプロセス時に作成する必要があり、以下から構成されています。

- **環境従属インターフェイス (ドライバ)**
- **Natural パラメータモジュール NATPARM**
- **Natural ワークファイルおよび出力ファイルのアクセスメソッドモジュール NATWKFO (AM=STD)**

### 環境従属インターフェイス (ドライバ)

NIMDRIV マクロを使用して、Natural IMS インターフェイスでサポートされている各 IMS 環境に対して環境従属インターフェイスを生成する必要があります。

NIMDRIV マクロの詳細については、「[NIMDRIV マクロパラメータ](#)」を参照してください。

### Natural パラメータモジュール NATPARM

Natural パラメータモジュールの詳細については、Natural の『オペレーション』ドキュメントで「[Natural パラメータモジュールのアセンブル](#)」を参照してください。

### Natural ワークファイルおよび出力ファイルのアクセスメソッドモジュール NATWKFO (AM=STD)

NATWKFO モジュールは基本 Natural の一部として提供されます。このモジュールは、AM=STD に定義されたワークファイルおよび出力ファイルの処理に使用されます。これは、オフラインの DL/I バッチリジョンを含む BMP 環境、メッセージ指向の環境、およびサーバー環境に適用できます。ダイアログ指向の環境には適用できません。

Natural for DB2 および Natural for DL/I などの一部の Natural 製品では、モジュールが Natural IMS フロントエンドモジュールにリンクされている必要があります。詳細については、該当する製品ドキュメントを参照してください。



## Natural IMS インターフェイスモジュール NIINTFM

Natural IMS インターフェイスモジュールはインストール処理時に作成する必要があり、すべての環境に共通しています。

このインターフェイスモジュールは、次のコンポーネントで構成されています。

- Natural IMS ニュークリアス
- Natural IMS パラメータモジュール NIIPARM
- トランザクションコードテーブル NIITRTAB
- メッセージテキストモジュール NIIMSGT
- DL/I 言語インターフェイス ASMTDLI

このインターフェイスモジュールは完全にリエントラントで、16 MB 境界より上で実行することができます。このため、ECSA は、すべての IMS 環境に対してこのインターフェイスモジュールのコピーを 1 つだけ持つことができます。

### Natural IMS ニュークリアス

Natural IMS ニュークリアスはロードモジュールとして提供され、Natural IMS インターフェイスに必要なランタイムルーチンがすべて含まれています。

### Natural IMS パラメータモジュール NIIPARM

Natural IMS パラメータモジュール NIIPARM は、複数 (1~n) のパラメータテーブル (「環境テーブル」とも呼ぶ) を含み、それぞれがマクロ NIMPARM によって定義され、パラメータ ENTRYNM によって識別されます。NIIPARM 内の各パラメータテーブルは、特定の環境に対して Natural IMS 固有のパラメータを定義します。このため、1 つのパラメータモジュールですべての Natural IMS 環境に対してパラメータを設定することができます。現在使用されている環境は、トランザクションコードテーブル NIITRTAB で設定されます。

NIMPARM マクロの詳細については、「[NIMPARM マクロパラメータ](#)」を参照してください。

### トランザクションコードテーブル NIITRTAB

トランザクションコードテーブルの NIITRTAB テーブルは可変長で、各エントリは NIMTRNTG マクロによって生成されます。

トランザクションコードテーブルの各エントリは、Natural IMS パラメータモジュールのエントリを参照し、NIMLPCB マクロの 1 つ以上のオカレンスが後ろに続く場合があります。

NIMLPCB マクロの詳細については、「[NIMLPCB マクロパラメータ](#)」を参照してください。

このテーブル内の適切なエントリは、現在のトランザクションコードによって検出されます。メッセージ駆動型以外の BMP で、トランザクションコードが定義されていない場合は、現在の PSB 名が代わりに取られます。

NIMTRNTG マクロの詳細については、「[NIMTRNTG マクロパラメータ](#)」を参照してください。

### メッセージテキストモジュール NIIMSGT

メッセージテキストモジュール NIIMSGT は Natural IMS インターフェイスモジュールの一部で、ロードモジュールおよびソースモジュールとして提供されます。ここでは、発生する可能性のある各 Natural IMS ランタイムエラーに対して、対応するメッセージテキストが含まれています。各エントリは、NIIMSGT マクロによって生成されます。

NIIMSGT マクロの詳細については、「[NIIMSGT マクロパラメータ](#)」を参照してください。

### DL/I 言語インターフェイス ASMTDLI

DL/I 言語インターフェイスの ASMTDLI は IMS/TM の一部です。

## 物理的な入力編集ルーチン

---

物理的な入力編集ルーチンは、ダイアログ指向の非会話型環境でのみ必要です。このルーチンは、端末に送信されるメッセージの前にトランザクションコードを挿入するために使用します。これは MFS バイパスモードでの Natural の実行に必要で、端末に送信されるメッセージにはトランザクションコードは含まれません。

物理的な入力編集ルーチンは NIMPIXT マクロを使用して生成されます。NIMPIXT マクロの詳細については、「[NIMPIXT マクロパラメータ](#)」を参照してください。

物理的な入力編集ルーチンが生成されると、このルーチンの名前を IMS/TM システム定義の TYPE または LINEGRP マクロで指定する必要があります。非会話型環境の実行がサポートされている端末の場合はいずれも、TERMINAL マクロで EDIT パラメータを使用して物理的な編集を有効にする必要があります。

## ユーザーメッセージテーブル DFSCMTU0

---

提供されているユーザーメッセージテーブル DFSCMTU0 は、ダイアログ指向の非会話型環境でのみ必要です。このテーブルには、物理的な入力編集ルーチンによって検出されるエラーのエラーメッセージが含まれています。

ユーザーメッセージテーブル DFSCMTU0 は、IMS/TM インストールの既存のユーザーメッセージテーブルに統合する必要があります。IMS/TM インストールにすでに存在するユーザーメッセー

ジ番号と競合する場合は、EQUATES PIXTE および SIPSE を変更して新しいメッセージ番号の範囲を作成することにより、提供された DFSCMTUO のメッセージ番号を変更することができます。メッセージ番号の範囲の新しい開始値は、NIMPXT マクロで指定する必要があります。

## ロールファイルおよびロールサーバー

これらのコンポーネントは、ダイアログ指向の環境でのみ使用されます。

Natural セッション関連の情報は、Natural スレッドに保持されます。端末出力ごとに、Natural スレッドの内容はロールファイルに、またはロールサーバーを使用して保存されます。媒体は NIMPARM パラメータ ROLLSRV によって定義されます。

### ロールファイルの使用

ロールファイルを使用するには、ROLLSRV パラメータを NO に設定します。


ロールファイルのロールスロットは Natural セッション初期化時に、Natural ユーザーごとに予約されます。スロットの識別子は、Natural セッションが開始される IMS/TM LTERM です。このため、ロールファイルの同じセットを使用するすべての端末が、異なる LTERM 名を持っているようにする必要があります。このことは、ロールファイルが単一の IMS/TM によって使用される場合は常に該当します。Natural セッションが正常に終了すると、スロットが解放されます。異常終了した場合にはロールスロットが割り当てられたままになりますが、同じユーザー（ユーザーの LTERM によって識別）が新しい Natural セッションを開始すると再利用されます。

ロールファイルは、DD ステートメント ROLLF1~ROLLF5 でアクセスされます。使用されるロールファイルの数は、NIMPARM パラメータ ROLLFN によって定義されます。

Natural トランザクションコードが複数の MPP リージョンでスケジュールされている場合、または異なる MPP リージョンで実行するトランザクションコード間で切り替える場合は、すべての MPP リージョンで同じロールファイルを使用する必要があります。

1つまたは複数のロールファイルを再フォーマットする場合は、アクティブな状態の Natural トランザクションが存在しないことを確認してください。ロールファイルが再初期化された後にトランザクションがスケジュールされている場合、ロールファイルにロールスロットを配置することができず、異常終了します。この問題を回避するために、ロールファイルが再フォーマットされた後に IMS/TM をコールドスタートすることをお勧めします。

IMS 環境の Natural によって使用されるロールファイルは、ロールサーバーによって使用されるロールファイルとレイアウトが同じであり、同じユーティリティプログラムによってフォーマットされます。

 **注意:** IMS 環境の Natural によって使用されるロールファイルを、ロールサーバーと共有することはできません。ロールファイルを IMS 環境の Natural とロールサーバーで同時

に使用する場合は、独自のロールファイルのセットをロールサーバーに割り当てる必要があります。

### ロールサーバーの使用

ロールサーバーを使用するには、`ROLLSRV` パラメータを YES に設定します。

各 MPP リージョンに割り当てる必要のあるロールファイルを使用する代わりに、Natural ロールサーバーを使用することができます。ロールサーバーには、次の長所があります。

- 各 MPP リージョンに DD ステートメントがありません。
- 中心的な 1 つのアドレススペースでロールファイルへのアクセスを処理します。
- メインストレージのバッファをサポートし、ロールファイルへのディスク I/O を削減します。

ロールサーバーのスロットは、Natural セッション初期化時に Natural ユーザーごとに予約されます。スロットの識別子（ロールサーバーのユーザー ID）はセッションが開始される IMS/TM LTERM で、z/OS ホスト ID、および対応する Natural トランザクションがスケジュールされる IMS/TM 依存のリージョンの IMS/TM サブシステム ID と連結されます。Natural セッションが正常に終了すると、スロットが解放されます。異常終了した場合にはスロットが割り当てられたままになりますが、同じユーザー（ユーザーの LTERM によって識別）が新しい Natural セッションを開始すると再利用されます。

SYSPLEX 環境では、ロールサーバーを使用する必要があります。

ロールファイルおよびロールサーバーの詳細については、Natural の『オペレーション』ドキュメントの「ロールサーバー」を参照してください。

## 認可サービスマネージャ

---

Natural 認可サービスマネージャは、次の場合に必要です。

- ダイアログ指向の非会話型環境。「[非会話型環境の特別な考慮事項](#)」を参照してください。
- モニタリングまたはブロードキャストが使用される場合。「[モニタリング](#)」または「[ブロードキャスト](#)」を参照してください。
- アカウンティングが使用され、アカウンティング情報が SMF に書き込まれる場合。「[アカウンティング](#)」を参照してください。
- バッファプールが反映される場合。プロファイルパラメータ `BPPROP` を参照してください。

最初の 2 つのケースでは、オプションの SIP 機能を、認可サービスマネージャの起動中に利用できるようにしておく必要があります。

SYSPLEX 環境では、SIP をカップリング機能に配置する必要があります。

## Natural 共有ニュークリアス

---

IMS/TM 環境では、Natural ニュークリアスは、環境依存のインターフェイス（ドライバ）から常に分離されます。つまり、共有の Natural ニュークリアスをインストールする必要があります。同じ Natural ニュークリアスを、すべての Natural IMS 環境で共有することができます。

詳細については、Natural の『オペレーション』ドキュメントの「Natural 共有ニュークリアス」を参照してください。

## Natural バッファプール

---

IMS 環境の Natural は複数の MPP リージョンで実行可能なので、Natural バッファプールをグローバルバッファプールにすることをお勧めします。

ローカルバッファプールを使用することもできますが、パフォーマンスの理由から、端末駆動の環境ではお勧めしません。

詳細については、Natural の『オペレーション』ドキュメントの「Natural グローバルバッファプール」を参照してください。

## Adabas インターフェイス


---

Natural システムファイルおよび Adabas ユーザーファイルにアクセスするには、Adabas インターフェイスが必要です。

デフォルトでは、適切な Adabas インターフェイスは、ランタイムにダイナミックにロードされます。

- ダイアログ指向で端末駆動の環境では、Adabas/IMS インターフェイスモジュール ADALNI が使用されます。
- 他のすべての環境では、Adabas バッチインターフェイスモジュール ADALNK が使用されます。

Natural プロファイルパラメータ ADANAME を指定することによって、使用する Adabas インターフェイスの名前を上書きすることができます。

 **注意:** これらのインターフェイスモジュールのリエントラントバージョンは使用しないでください。

## プリロードリスト

---

Natural IMS インターフェイスではプリロードリストを使用する必要はなくなりましたが、パフォーマンスの理由から、Natural リージョンのプリロードリストに次のモジュールの名前を追加することをお勧めします。

- Natural IMS フロントエンド
- Natural IMS インターフェイスモジュール
- Natural 共有ニュークリアス
- Adabas インターフェイス

# 24 IMS/TM 環境での Natural - コンフィグレーションマ

## クロ

---

▪ NIMDRIV マクロパラメータ .....	158
▪ NIMPARM マクロパラメータ .....	160
▪ A .....	160
▪ B~C .....	161
▪ E~H .....	162
▪ L~N .....	163
▪ P .....	164
▪ R~S .....	166
▪ T~U .....	167
▪ NIMTRNTG マクロパラメータ .....	168
▪ NIMLPCB マクロパラメータ .....	171
▪ NIMMSGT マクロパラメータ .....	171
▪ NIMPIXT マクロパラメータ .....	172
▪ NIMBOOT マクロパラメータ .....	173

Natural IMS インターフェイスドキュメントのこの章では、Natural IMS インターフェイスのコンフィグレーションマクロについて説明します。

## NIMDRIV マクロパラメータ

NIMDRIV マクロは環境従属インターフェイス（ドライバ）を生成します。NIMDRIV マクロで指定可能なパラメータについて、以下に説明します。

TYPE | LE370 | NIINAME | SUBPOOL | TRNCODE | THRELO

パラメータ	設定可能値	説明	デフォルト	コメント
TYPE	CON	TYPE は、生成されるドライバのタイプを指定します。  「CONV」 の場合には、ダイアログ指向の会話型の環境が生成されます。	なし	なし
	NONC	ダイアログ指向の非会話型の環境が生成されます。		
	NTRD	メッセージ指向の（非端末駆動の）環境が生成されます。		
	BMP	バッチ指向でトランザクション指向の BMP 環境が生成されます。  DLIBATCH 環境でもこのインターフェイスを使用することができます。		
	SRVD	サーバー環境が生成されます。		
LE370	YES	LE370 は、IMS 環境の Natural が IBM 言語環境を初期化するかどうかを指定します。  YES の場合には、IBM 言語環境は初期化され、Natural IMS フロントエンドが IMS プログラムコントローラに戻るまで、この状態が維持されます。  LE 準拠の 3GL プログラムが、LE 呼び出し規則に従って呼び出されます。LE プログラムに制御がある間は、ユーザーが記述した条件ハンドラまたは言語固有の条件ハンドラが適用されます。	NO	なし
	AMODE24	YES の場合と同様に、IBM 言語環境が初期化されます。ただし YES の場合とは違い、LE ランタイムオプション ALL31=(OFF)、STACK=(,BELOW) は、AMODE(24) 3GL プログラムをサポートするように設定されています。このオプションを使用すると、AMODE(24) 3GL プログラムを使用していて、LE ランタイムオプションのシステムデフォルトでは AMODE(24) 3GL プログラムの実行が許可されていない場合でも、IBM 言語環境で Natural を実行することができます（例えば REQUEST DOCUMENT ステートメントを使用）。		



パラメータ	設定可能値	説明	デフォルト	コメント
	NO	IBM 言語環境は初期化されません。IBM 言語環境は LE プログラムコールごとに初期化されて終了します。		
NIINAME	XXXXXXXX	現在のドライバに使用される Natural IMS インターフェイスモジュールの名前を指定します。8文字までの任意の有効なモジュール名を使用することができます。	NIIINTFM	なし
SUBPOOL	<i>n</i>	STORAGE OBTAIN 要求に対して z/OS サブプール番号を指定します。問題状況のプログラムに対して任意の有効な z/OS タスク関連のサブプール番号 (0~127) を使用することができます。  <b>注意:</b> 同じ IMS/TM 依存リージョンで Natural IMS/TM インターフェイスの旧バージョンを操作する場合は、サブプール番号 126 および 127 を使用しないでください。	125	
TRNCODE	YES	オプションの TRNCODE パラメータが適用されます。適用先は以下のとおりです。  ■ BPM CONTROL ファイルの 1 つ目のカードの TRNCODE パラメータ  ■ NIMBOOT マクロの TRNCODE パラメータ  ■ サーバー環境の TRNCODE 起動パラメータ	NO	
	NO	TRNCODE パラメータは無視されます。		
THRELO	YES	スレッドの再配置が有効になります。つまり、Natural スレッドは端末 I/O 後に異なる仮想アドレスに割り当てられる場合があります。	YES	
	NO	スレッドの再配置が無効になります。また、Natural スレッドは同じ仮想アドレスを維持します。Natural スレッドのサイズは、スレッドを割り当てる最初の Natural セッションによって決定されます。結果は次のようになります。  ■ Natural/IMS THSIZE パラメータは、MPP の最初の Natural セッションを除くすべてのセッションで無視されます。  ■ RELO=FORCE が無効になります。  ■ MPP が停止されるまで、Natural スレッドのストレージは割り当てられた状態を維持します。  <b>注意:</b> このパラメータは、Con-form を使用している顧客以外は使用しないでください。		

## NIMPARM マクロパラメータ

NIMPARM マクロは、NIIPARM パラメータモジュールに含まれるパラメータテーブルを生成します。

NIMPARM マクロで指定可能なパラメータについて、以下に説明します。ENTRYNM パラメータは、現在のパラメータテーブルを識別します。

パラメータを、アルファベット順で以下に記載します。

ACTACTV | ACTAHDR | ACTARID | ACTLOG | BMPABER | BROACTV | CMBSIZE | COLPSCR | ENDMODN |  
 ENTRYNM | ERRLHDR | HCBSIZE | HDENSU | LINPSCR | MISIZE | MONACTV | MOSIZE | MSACTV | MSCMPTB  
 | MSCRKEY | MSDBD | MSMAX | MSRSKEY | NSBNAME | PRTRDRIV | ROLLSRV | ROLLFN | SPASIZE | SPATID  
 | SUPNONC | TERMDB | TERMIPL | THBELOW | THSIZE | USERID

### A

パラメータ	設定可能値	説明	デフォルト	コメント
ACTACTV	YES	ACTACTV は、アカウントिंग機能をアクティブにするかどうかを指定します。  YES の場合には、アカウントングレコードが端末 I/O ごとに書き込まれます。	NO	ダイアログ指向の環境でのみ使用します。
	NO	アカウントングレコードは書き込まれません。		
ACTAHDR	xxxxxxxx	IMS ログファイルに書き込まれた場合にアカウントングレコードのヘッダーを定義します。8 バイトまでの任意の文字列を使用することができます。	SAG\$\$\$\$	このパラメータは、ACTLOG パラメータが CMD に設定されている場合のみ評価されます。アカウントング機能にのみ使用します。
ACTARID	log code	ACTLOG パラメータの LOG または SMF 設定を使用してアカウントングレコードが書き込まれる場合に、アカウントングレコード ID を指定します。	なし	アカウントング機能にのみ使用します。
	または			
	SMF record type			
		ログコード (A0~FF)	ACTLOG パラメータ (以下を参照) が LOG に設定されている場合。	
		SMF レコードタイプ (128~255)	ACTLOG パラメータが SMF に設定されている場合。	

パラメータ	設定可能値	説明	デフォルト	コメント
ACTLOG	CMD	ACTLOGは、アカウントレコードの書き込み方法を指定します。CMD の場合には、アカウントレコードは CMD 呼び出しを使用して IMS ログファイルに書き込まれます。	CMD	アカウントレコード機能にのみ使用します。
	LOG	アカウントレコードは、LOG 呼び出しを使用して IMS ログファイルに書き込まれます。		
	SMF	アカウントレコードは認可サービスマネージャを使用して SMF に書き込まれます。		

## B～C

パラメータ	設定可能値	説明	デフォルト	コメント
BMPABER	YES	BMPABER は、Natural ランタイムエラーまたは Natural IMS インターフェイスに回復不可能なエラーが発生した場合に、BMPの実行をどのように終了するかを指定します。  YES の場合には、BMP の実行はユーザーアベンドコード U3521 で異常終了します。	NO	なし
	NO	BMP の実行は正常に終了し、コンディションコードとして Natural 終了エラーが出力されます。BMP の実行が回復不可能な Natural IMS エラーのために終了した場合は、コンディションコード 1024 が設定されます。		
BROACTV	YES	ブロードキャスト機能を利用するかどうかを指定します。	NO	ダイアログ指向の環境でのみ使用します。
	NO			
CMBSIZE	xxxxx	コマンドバッファのサイズをバイト単位で指定します。16 MB までの任意の数値を指定することができます。	1024	コマンドバッファは、サービス API NIICMD および NIIGCMD、サービスモジュール CMCMND およびアカウントレコード機能によって使用されます。  コマンドバッファのサイズは、処理される IMS コマンドの最大長、およびユーザー拡張を含むアカウントレコードの最大長に適合している必要があります。

パラメータ	設定可能値	説明	デフォルト	コメント
COLPSCR	xx	画面ごとの列数を指定します。任意の有効な画面幅（数値）に設定することができます。	80	なし

## E～H

パラメータ	設定可能値	説明	デフォルト	コメント
ENDMODN	xxxxxxxx	Naturalセッションが正しく終了した後に表示される画面を設定するためのMOD名を指定します。8文字までの任意の有効なMOD名を指定することができます。	DFSM02	Naturalを顧客固有のメニューに含めることができます。  ENDMODNパラメータの値は、サービスAPI <b>NIIEMOD</b> およびサービスモジュール <b>CMEMOD</b> によって無効にすることができます。  Naturalセッションがエラーで終了する場合は、適切なNaturalエラーメッセージを発行するためにDFSM02が常に使用されます。
ENTRYNM	xxxxxxxx	現在のパラメータテーブルを識別します。8文字までの任意の文字列を使用することができます。	ENV00000	なし
ERRLHDR	xxxxxxxx	Natural IMS インターフェイスでエラーが発生したときに書き込まれるIMS ログレコードのヘッダーを指定します。8文字までの任意の文字列を使用することができます。	NIIERR\$\$	回復不可能なNatural IMSエラーが発生した場合にメッセージがIMSログに書き込まれないようにするには、ERRLHDRパラメータを明示的に空値に設定します。つまり、ERRLHDR=, を指定します。  詳細については、「 <a href="#">リカバリ処理</a> 」を参照してください。
HCBSIZE	xxxxx	ハードコピープリントバッファのサイズをバイト単位で指定します。16 MB までの任意の数値を指定することができます。	1024	Natural ハードコピー機能を使用してプリンタの出力先に送信されるレコードはバッファされます。
HDENS DU	YES NO	Natural IMSエラーによって引き起こされたスナップダンプを高密度ダンプとして3800の印刷サブシステムに書き込むかどうかを指定します。	NO	なし

## L~N

パラメータ	設定可能値	説明	デフォルト	コメント
LINPSCR	xx	画面ごとの行数を定義します。任意の有効な画面サイズ（数値）を設定することができます。	24	なし
MISIZE	xxxxx	入力メッセージを含むバッファのサイズ（バイト単位）。16MBまでの任意の数値を指定することができます。	4096	このエリアは、IMS/TMから受信する最大の入力メッセージと同じ大きさである必要があります。
MONACTV	YES	MONACTVは、モニタリング機能をアクティブにするかどうかを指定します。  YESの場合には、セッションステータスが端末 I/O ごとに SIP サーバーに書き込まれます。	NO	ダイアログ指向の環境でのみ使用します。
	NO	セッションステータスは保持されません。		
MOSIZE	xxxxx	出力メッセージを含むバッファのサイズ（バイト単位）。16MBまでの任意の数値を指定することができます。	4096	このエリアは、IMS/TMに送信する最大出力メッセージと同じ大きさである必要があります。
MSACTV	YES	マルチセッション機能を利用するかどうかを指定します。	NO	ダイアログ指向の環境でのみ使用します。
	NO			
MSCMPTB	YES	セッションを、Natural IMS インターフェイスバージョン2.2またはIMS/TM環境のNatural Advanced Interfaceバージョン2.2と互換性を持つモードに切り替えるかどうかを指定します。	NO	マルチセッション機能にのみ使用されます。
	NO			
MSCRKEY	NONE	新しいセッションを開始できる PF キーを指定します。	NONE	マルチセッション機能にのみ使用されます。  MSCMPTB=YES の場合は、MSCRKEY を NONE に設定する必要があります。
	PF1~PF24			
MSDBD	DBD-name	マルチセッションデータベースの名前を指定します。任意の有効な DBD 名を指定することができます。	なし	マルチセッション機能にのみ使用されます。
MSMAX	2 - 9	端末ごとに可能な Natural 並列セッションの最大数を指定します。	9	マルチセッション機能にのみ使用されます。
MSRSKEY	NONE	古いセッションを再スタートできる PF キーを指定します。	NONE	マルチセッション機能にのみ使用されます。 <b>MSCMPTB=YES</b> の場合は、
	PF1~PF24			

パラメータ	設定可能値	説明	デフォルト	コメント
				MSRSKEY を NONE に設定する必要があります。
NSBNAME	PSBNAME	NSB 名を IMS/TM によって使用される PSB 名に設定します。	PSBNAME	Natural for DL/I インターフェイスによって使用されます。
	NIITRTAB	NSB 名を、トランザクションコードテーブル NIITRTAB で、使用中のトランザクションコードに割り当てられた PSB 名に設定します。		

## P

パラメータ	設定可能値	説明	デフォルト	コメント
PRTDRIV	以下に示すドライバの表を参照してください。  SCS プリンタのドライバ 非 SCS プリンタのドライバ JES API のドライバ	IMS/TM プリンタに直接書き込まれるレポートに使用されるプリントドライバを指定します。	SCS_S2	詳細については、「 <i>Natural WRITE (n) ステートメントのサポート</i> 」を参照してください。

### SCS プリンタのドライバ

ドライバ	目的
SCS_B1	レポートの開始および終了でフォームフィード。1 行目からページを開始します。
SCS_B2	レポートの開始および終了でフォームフィード。2 行目からページを開始します。
SCS_E1	レポートの終了でフォームフィード。1 行目からページを開始します。
SCS_E2	レポートの終了でフォームフィード。2 行目からページを開始します。
SCS_N1	レポートの開始または終了でフォームフィードしません。1 行目からページを開始します。
SCS_N2	レポートの開始または終了でフォームフィードしません。2 行目からページを開始します。
SCS_S1	レポートの開始でフォームフィード。1 行目からページを開始します。
SCS_S2	レポートの開始でフォームフィード。2 行目からページを開始します。

## 非 SCS プリンタのドライバ

ドライバ	目的
NSCS_B1	レポートの開始および終了でフォームフィード。1行目からページを開始します。
NSCS_B2	レポートの開始および終了でフォームフィード。2行目からページを開始します。
NSCS_E1	レポートの終了でフォームフィード。1行目からページを開始します。
NSCS_E2	レポートの終了でフォームフィード。2行目からページを開始します。
NSCS_N1	レポートの開始または終了でフォームフィードしません。1行目からページを開始します。
NSCS_N2	レポートの開始または終了でフォームフィードしません。2行目からページを開始します。
NSCS_S1	レポートの開始でフォームフィード。1行目からページを開始します。
NSCS_S2	レポートの開始でフォームフィード。2行目からページを開始します。

## JES API のドライバ

ドライバ	目的
JES	この場合、JES に対して次のデータセット処理オプションが、対応する NTPRINT マクロパラメータまたは DEFINE PRINTER ステートメントオプションから取られます。
JES	NTPRINT                      DEFINE PRINTER
CLASS	CLASS                              CLASS
COPIES	COPIES                            COPIES
DEST	DEST                                OUTPUT
FORMS	FORMS                              FORMS
NAME	NAME                                NAME
OUTDISP	DISP                                DISP
PRTY	PRTY                                PRTY
	生成される JES API パラメータ文字列は次のとおりです。
	<pre>IAFP=AOM,PRT0=..OUTDI(<i>disp</i>),DES(<i>dest</i>), CLA(<i>class</i>,COP(<i>copies</i>), FORMS(<i>forms</i>),NAME(<i>name</i>), PRTY(<i>prty</i>)</pre>
	<b>注意:</b> NTPRINT/DEFINE PRINTER パラメータ/オプションは、指定しなければ無視されます。

ドライバ	目的
JESxxxxx	<p>この場合、JESに対するデータセット処理オプションは、名前が JESxxxxx の OUTPUT JCL ステートメントから取られます。</p> <p>生成される JES API パラメータ文字列は次のとおりです。</p> <pre>IAFP=AOM,OUTN=JESxxxxx</pre> <p>OUTPUT JCL ステートメントは、次のようになります。</p> <pre>JESxxxxx OUTPUT OUTDISP=WRITE,DEST=dest, CLASS=A,COPIES=1,FORMS=form,...</pre> <p><b>注意:</b> OUTPUT JCL ステートメントがジョブストリームに見つからない場合は、エラーが報告されます。</p>

## R~S

パラメータ	設定可能値	説明	デフォルト	コメント
ROLLSRV	YES	ROLLSRV は端末の出力と入力間の Natural スレッドを保存する媒体を指定します。ROLLSRV=YES の場合は、Natural ロールサーバーが使用されます。	YES	ダイアログ指向の環境でのみ使用します。  「IMS/TM 環境での Natural - コンポーネント」の「ロールファイルおよびロールサーバー」も参照してください。
	NO	ロールファイルが使用されます。以下の ROLLFN を参照してください。		
ROLLFN	1 - 5	ROLLSRV=NO の場合は、使用するロールファイルの数を指定します。	1	ダイアログ指向の環境でのみ使用します。
SPASIZE	xxxxx	スクラッチパッドエリアを含むバッファのサイズをバイト単位で指定します。  16 MB までの任意の数値を指定することができます。	1024	非会話型の環境では、これは SIP サーバーに書き込まれるシミュレートされた SPA のサイズでもあります。
SPATID	xxxx	非会話型のドライバの SPA を保存するために使用する認可サービスマネージャの Natural サブシステム ID を指定します。4 文字までの任意の文字列を使用することができます。		この値はすべてのパラメータテーブルで同じであり、NIMPIXT マクロの SPATID に指定した値と同じである必要があります。



パラメータ	設定可能値	説明	デフォルト	コメント
SUPNONC	YES	端末指向の非会話型環境から会話型環境に切り替え可能にするかどうかを指定します。	NO	ダイアログ指向の会話型環境でのみ使用します。
	NO			

## T～U

パラメータ	設定可能値	説明	デフォルト	コメント
TERMDB	YES	PSB で指定された DL/I データベースのいずれかを利用できない場合に、Natural セッションを終了する必要があるかどうかを指定します。	NO	ダイアログ指向の環境でのみ使用します。  TERMDB を NO に設定し、データベースの1つにアクセスしたときにそれを利用できない場合は、Natural トランザクションコードが IMS/TM によって中断されます。
	NO			
TERMIPL	YES	IPL が現在のトランザクション手順とセッション開始時の間に発生したときに、Natural セッションを終了してエラーメッセージを出力するかどうかを指定します。	NO	ダイアログ指向の環境でのみ使用します。
	NO			
THBELOW	YES	Natural スレッドの割り当て場所を指定します。YES の場合、Natural スレッドは 16 MB 境界より下に割り当てられます。	YES	バッチメッセージの処理については、スレッドは常に 16 MB 境界より下に割り当てられます。
	NO	Natural スレッドは、16 MB 境界より上に割り当てられます。		
THSIZE		Natural スレッドのサイズを指定します。100000 以上の任意の 8 の倍数を指定することができます。	300000	これはユーザーセッションに関連する Natural バッファがすべて含まれるエリアです。
USERID	YES	システム変数 *INIT-USER の値を決定する方法を指定します。YES の場合、*INIT-USER で指定される Natural ユーザー ID は、セキュリティパッケージがアクティブな場合はセキュリティアクセスコントロールブロックから、またはジョブカードの USER パラメータから取られます。	NO	BMP ドライバにのみ使用されます。
	NO	*INIT-USER で指定される Natural ユーザー ID は、ジョブ名から取られます。		

## NIMTRNTG マクロパラメータ

NIMTRNTG マクロは、関連するパラメータで指定されたトランザクションコードを含むトランザクションコードテーブル NIITRTAB にエントリを生成します。Natural トランザクションコードごとに、トランザクションコードテーブルにエントリを含める必要があります。NIITRTAB の詳細については、「[トランザクションコードテーブル NIITRTAB](#)」を参照してください。

NIMTRNTG マクロで指定できるパラメータを、以下にアルファベット順に記載します。

[ALTPCB](#) | [HCPCB](#) | [MSGPCB](#) | [MSPCB](#) | [NIIPENT](#) | [NRASRT](#) | [PSBNAME](#) | [TRANCODE](#) | [TYPE](#) | [WRKPCBS](#)

パラメータ	設定可能値	説明	デフォルト	コメント
ALTPCB	1 - 255	ALTPCB は、サービスモジュール <a href="#">CMQUEUE</a> 、 <a href="#">CMQUEUEEX</a> 、 <a href="#">NIIDQUMS</a> 、および <a href="#">NIIDPURG</a> に対して使用する代替 TP PCB を指定します。	1	NIMTRNTG マクロで指定された代替 TP PCB の数は、サービスモジュールによって書き換えることができます。
HCPCB	SYSPCB	HCPCB は、ハードコピー機能に使用する PCB を指定します。  SYSPCB の場合は、1 つ目の代替 TP PCB が使用されます。	SYSPCB	なし
	WRKPCB	追加の代替 TP PCB のいずれかが使用されます。これによりエクスプレス TP PCB をハードコピー機能に使用することができます。		
MSGPCB	SYSPCB	MSGPCB は、非端末指向の環境において、およびサーバードライバに対して、エラーメッセージおよび標準出力を出力するときに使用する PCB を指定します。  SYSPCB の場合は、1 つ目の代替 TP PCB が使用されます。	SYSPCB	非端末指向の環境およびサーバードライバに対してのみ関係します。
	OWNPCB	2 つ目の代替 TP PCB が予約および使用されます。これによりエクスプレス TP PCB をメッセージの送信に使用することができます。		
MSPCB	NO	MSPCB は、マルチセッションデータベース PCB の数を指定します。  NO を指定すると、マルチセッション機能は使用されません。	NO	マルチセッション機能に対してのみ関係します。

パラメータ	設定可能値	説明	デフォルト	コメント
	1 - 255	マルチセッションデータベースの PCB。		
NIIPENT	xxxxxxxx	トランザクションコードテーブルでこのエントリに対して使用される Natural IMS パラメータテーブルの名前を指定します。空白文字以外で8文字までの文字列を使用することができます。	ENV00000	なし
NRASTART	<i>offset value</i>	スクラッチパッドエリア内に Natural Reserved Area (NRA) のオフセットを定義します。14 より大きい任意の数値を指定することができます。	16	現在の NRA の長は 157 バイトです。 NRA の長さは Natural IMS インターフェイスのバージョンによって異なる場合があります。  非 Natural トランザクションに渡すために SPA に独自の情報を保存する場合は、バージョンの互換性を維持するために NRA の前にデータを保存することをお勧めします。
PSBNAME	<i>PSB-name</i>	現在のトランザクションコードに対応する PSB 名を指定します。任意の有効な PSB 名を指定することができます。	なし	非メッセージ駆動型バッチメッセージ処理およびバッチ処理環境に対してトランザクションコードテーブルのエントリを識別するために使用します。
TRANCODE	<i>transaction-code-name</i>	トランザクションコードテーブル内の各エントリの識別子を指定します。任意の有効なトランザクションコード名を指定することができます。	なし	非メッセージ駆動型 BMP およびバッチ処理環境の両方に影響がありません。
TYPE	CONV	TYPE は、Natural トランザクションコードのタイプを指定します。CONV の場合には、トランザクションコードは会話型の Natural セッション用になります。	CONV	なし

パラメータ	設定可能値	説明	デフォルト	コメント
	NONC	トランザクションコードは、非会話型の Natural セッション用になります。		
WRKPCBS	0	WRKPCBS は、1 つ目の TP PCB に、また該当する場合は MSGPCB に追加を出力するために利用できる代替 TP PCB の数を指定します。  0 の場合には、IMS プリンタは利用できません。	0	以下の例を参照してください。
	1 - 32	1 つ目の TP PCB に、また該当する場合は MSGPCB に追加を出力するために使用される代替 TP PCB の数。		

## 例

### 例 1

次のように指定します。

```
MSGPCB=SYSPCB
WRKPCBS=2
```

PSB は 3 つの代替 TP PCB を含んでいる必要があります。

### 例 2

次のように指定します。

```
MSGPCB=OWNPCB
WRKPCBS=2
```

PSB は 4 つの代替 TP PCB を含んでいる必要があります。2 つ目の代替 TP PCB は、非端末指向の環境のエラーメッセージおよび標準出力のために予約されています。

## NIMLPCB マクロパラメータ

NIMLPCB マクロは、オプションとしてトランザクションコードテーブルの NIMTRNTG エントリの後続けることができます。

NIMLPCB マクロで指定できるパラメータを、以下にアルファベット順に記載します。

NAME | NUM

パラメータ	設定可能値	説明	デフォルト	コメント
NAME	xxxxxxxx	PCB の論理名を指定します。空白文字以外で8文字までの文字列を使用することができます。	なし	なし
NUM	<i>PCB-positional-number</i>	PSB での PCB の位置番号を指定します。任意の整数を指定することができます。	なし	NUM を指定しないと、NIMLPCB マクロの位置番号が使用されます。

## NIMMSGT マクロパラメータ

NIMMSGT マクロは、Natural IMS インターフェイスモジュールの一部であるメッセージテキストモジュール NIIMSGT の各エントリを生成します。生成された各エントリは、発生する可能性のある各 Natural IMS エラー番号に対してメッセージテキストを提供します。

NIMMSGT マクロは、次の2つの方法のいずれかを使って指定します。

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

この場合、IMS/TM 環境の Natural は定義されたメッセージテキストを表示します。メッセージテキストの最大長は 72 文字です。

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

この場合、IMS 環境の Natural は現在のメッセージテキストにエラー固有の理由コードを追加します。メッセージテキストの最大長は 64 文字です。

エラー番号の後にアスタリスク (\*) を付けると、エラーが発生したときにスナップダンプが生成されます。メッセージテキストは、独自の要件に適合させることができます。また、特定の

エラー番号の DUMP オプションを追加または削除することができます。エラー番号、およびエラー番号の前にある N または R の文字は変更しないでください。

## NIMPIXT マクロパラメータ

NIMPIXT マクロは物理的な入力編集ルーチンを生成します。

NIMPIXT マクロで指定できるパラメータを、以下にアルファベット順に記載します。

PIXTE | SIPSE | SPATID | WTO | USER

パラメータ	有効な値	説明	デフォルト	コメント
PIXTE	1 - 999	物理的な入力編集ルーチンによってエラーが検出された場合のエラー番号の開始値を指定します。	400	この値は、物理的な入力編集ルーチンによって設定されたリターンコードに追加されます。この結果、ユーザーメッセージテーブル DFSCMTUO の IMS/TM エラーメッセージ番号が得られます。
SIPSE	1 - 999	認可サービスマネージャによってエラーが検出された場合のエラー番号の開始値を指定します。	500	この値は、認可サービスマネージャによって設定されたリターンコードに追加されます。この結果、ユーザーメッセージテーブル DFSCMTUO の IMS/TM エラーメッセージ番号が得られます。
SPATID	xxxx	非会話型のドライバで SPA の保存に使用される認可サービスマネージャの Natural サブシステム ID を指定します。4 文字までの任意の文字列を使用することができます。	なし	このパラメータの値は、NIMPARM マクロの SPATID パラメータで指定された値と同じである必要があります。
WTO	YES	認可サービスマネージャが失敗した場合に WTO メッセージを発行するかどうかを指定します。	NO	なし
	NO			
USER	xxxxxxxx	NIMPIXT マクロで SPA が見つからない場合に、ユーザー固有の物理的な入力編集ルーチンを呼び出すかどうかを指定します。  ユーザー固有の入力編集ルーチンを呼び出す場合は、ルーチンの名前を指定します。	NO	なし
	NO			

## NIMBOOT マクロパラメータ

NIMBOOT マクロは、メッセージ指向の環境によって使用されるブートストラップモジュール、またはサーバー環境によって使用されるサーバーコールインターフェイスを生成します。

NIMBOOT は、次のパラメータを含みます。

TYPE | DRIVERN | ENVTNAM | TRNCODE | DYNPARM | SERVERN

パラメータ	設定可能値	デフォルト	コメント
TYPE	SERVER	Empty	TYPE は、生成されるインターフェイスモジュールのタイプを指定します。  TYPE=SERVER を指定すると、サーバーコールインターフェイス NIIBOOTS が生成されます。
	Empty		何も指定しないと、メッセージ指向の環境によって使用されるブートストラップモジュールが生成されます。
DRIVERN	任意の有効な z/OS モジュール名	なし	このパラメータは、フロントエンドモジュールの名前を指定します。  TYPE=SERVER を指定する場合は、フロントエンドモジュールがサーバー環境に対して生成されている必要があります (NIMDRIV パラメータ TYPE=SRVD)。TYPE を指定しない場合は、フロントエンドモジュールがメッセージ指向の環境に対して生成されている必要があります (NIMDRIV パラメータ TYPE=NTRD)。
ENVTNAM	任意の有効な z/OS モジュール名	なし	このパラメータは、メッセージ指向の環境に対するブートストラップモジュールによってのみ使用されます (TYPE は空)。  このパラメータは、環境テーブルの名前を指定します。このパラメータはオプションです。指定しないと、環境テーブルは、使用されるトランザクションコードに対応するトランザクションコードテーブルのエントリによって決定されます。
TRNCODE			このパラメータは、メッセージ指向の環境に対するブートストラップモジュールによってのみ使用されます (TYPE は空)。  このパラメータは、Natural IMS インターフェイスによって内部的に使用されるトランザクションコードの名前を指定します。このパラメータはオプションで、TRNCODE=YES が NIMDRIV マクロでコード化されている場合のみ適用されます。このパラメータを指定しない場合、または TRNCODE=YES が NIMDRIV マクロで指定されていない場合は、IMS/TM INQY 呼び出しによって返されたトランザクションコードが使用されます。  トランザクションコードは、トランザクションコードテーブル NIIITRAB のエントリを決定するために使用されます。

パラメータ	設定可能値	デフォルト	コメント
DYNPARM	80文字までの任意の文字列。	なし	<p>このパラメータは、メッセージ指向の環境に対するブートストラップモジュールによってのみ使用されます (TYPE は空)。</p> <p>このパラメータは、Natural ダイナミックパラメータに 80 文字までの有効な文字列を定義するために使用されます。</p>
SERVERN	任意の有効な z/OS モジュール名	NIIBOOTS	<p>このパラメータは、サーバーコールインターフェイスによってのみ使用されます (TYPE=SERVER)。</p> <p>このパラメータは、サーバー環境の名前を指定します。同じリージョンで複数の Natural サーバーを使用する場合のみ関係します。この場合、複数のサーバーコールインターフェイスを生成し、それぞれに対して SERVERN を使って一意の名前を指定する必要があります。「特別な機能」の「サーバー環境」を参照してください。</p>



# 25

## IMS/TM 環境の Natural - サービスプログラム

---

■ Natural IMS/TM インターフェイスサービスプログラムについて .....	176
■ Natural IMS/TM インターフェイスサービスプログラムの説明 .....	177
■ NIIBRCST - 渡されたメッセージを端末に送信 .....	177
■ NIICMD - IMS コマンドを IMS に送信 .....	177
■ NIIDEFT - Natural トランザクションコードへの遅延切り替えを用意 .....	178
■ NIIDEFTX - 非 Natural のトランザクションコードへの遅延切り替えを用意 .....	178
■ NIIDIRT - Natural トランザクションコードへの直接切り替えを用意 .....	179
■ NIIDIRTX - トランザクションコードへの直接切り替えを用意 .....	180
■ NIIMOD - モジュール出力ディスクリプタの設定変更 .....	181
■ NIIGCMD - 以前の IMS/TM コマンドの次の応答セグメントを取得 .....	181
■ NIIGMSG - 次メッセージの最初のセグメントを取得 .....	182
■ NIIGSEG - 入力メッセージの次のセグメントを取得 .....	182
■ NIIGSPA - SPA からデータを取得 .....	183
■ NIIMSIN - IMS 環境情報を取得 .....	183
■ NIISRTF - 複数セグメントのメッセージを作成 .....	184
■ NIISRTM - メッセージキューにメッセージセグメントを挿入 .....	185
■ NIIPCBAD - PSB 名および PCB アドレスの返却 .....	185
■ NIIPCOM - 応答エリアにデータを移動 .....	186
■ NIIPMSG - メッセージを送信 .....	186
■ NIIPSBAD - PSB アドレスの返却 .....	187
■ NIIPSPA - SPA でのデータ置換 .....	187
■ NIIPURG - PURG 呼び出しの発行 .....	188
■ NIIRETRM - メッセージエリアにデータを移動 .....	188
■ NIISASD - SENDER と OUTDEST の設定を変更 .....	189
■ NIU3962 - セッションの終了 .....	189

Natural IMS インターフェイスドキュメントのこの部分では、Natural IMS/TM インターフェイスのサービスプログラムについて説明します。

## Natural IMS/TM インターフェイスサービスプログラムについて

---

### Natural IMS/TM インターフェイスサービスプログラムの目的

サービスプログラムとは、IMS 環境の Natural に追加機能を提供する Natural のサブプログラムです。これらは、標準の CALLNAT ステートメントを使用して Natural プログラム内部から呼び出すことができます。

### サービスプログラムの場所

サービスプログラムは SYSEXTP ライブラリで提供されています。これらは SYSTEM または steplib ライブラリにコピーする必要があります。サービスプログラムを起動するサンプルの Natural プログラムも SYSEXTP ライブラリで提供されています。

### 共通のリターンコード

各サービスプログラムの最後のパラメータは、形式が (I4) のリターンコードです。

次のリターンコード値は、すべてのサービスプログラムで共通しています。

0	OK
-1	サポートされていない関数。これは内部エラーです。Software AG サポートに連絡してください。

特定のリターンコード値については、以下の各サービスプログラムの説明を参照してください。

### エラー処理

エラーが発生した場合は、Natural エラーメッセージが発行されるか、またはセッションが終了して Natural IMS エラーメッセージが表示されます。Natural の『メッセージおよびコード』ドキュメントの「IMS/TM 環境下の Natural のエラーコード」を参照してください。

## Natural IMS/TM インターフェイスサービスプログラムの説明

次のサービスプログラムについて、以下に説明します。

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

### NIIBRCST - 渡されたメッセージを端末に送信

MOD\_name パラメータで指定されたメッセージ出力ディスクリプタを使用して、渡されたメッセージを指定された端末に送信します。

次のパラメータを使用できます。

パラメータ	フォーマット/長さ
Terminal_name	(A8)
Message	(A1/1:V)
Message_length	(I4)
MOD_name	(A8)
Return_code	(I4)

特定のリターンコード値：なし。

サンプルプログラム：NIPGMSG

### NIICMD - IMS コマンドを IMS に送信

指定された IMS コマンドを IMS に渡します。応答がある場合は、提供された応答エリアに移動されます。応答が応答エリアに適合しない場合は、切り捨てられてリターンコードが 4 に設定されます。

次のパラメータを使用できます。

パラメータ	フォーマット/長さ	タイプ
Command	(A1/I:V)	入力
Command_length	(I4)	入力
Reply_area	(A1/I:V)	入力/出力
Reply_area_length	(I4)	入力
Reply_length	(I4)	出力
Status_code	(A2)	出力
Return_code	(I4)	出力

特定のリターンコード値：4（応答は切り捨てられる）

サンプルプログラム：NIPCMD

## NIIDEFT - Natural トランザクションコードへの遅延切り替えを用意

指定された Natural トランザクションコードへの遅延切り替えを用意します。次の端末 I/O で出力は端末に送信され、この端末からの次の入力はパラメータ Transaction\_code で指定されたトランザクションコードによって処理されます。

次のパラメータを使用できます。

パラメータ	フォーマット/長さ	タイプ
Transaction_code	(A8)	入力
Return_code	(I4)	出力

特定のリターンコード値：なし。

サンプルプログラム：NIPDEFT

## NIIDEFTX - 非 Natural のトランザクションコードへの遅延切り替えを用意

非 Natural のトランザクションコードへの遅延切り替えを用意します。次の端末 I/O で、出力は指定された MOD\_name を使用して端末に送信され、この端末からの次の入力はパラメータ Transaction code で指定されたトランザクションコードによって処理されます。

中断フラグを Y に設定すると Natural セッションは中断され、後で再開できます。Natural セッションが再開されると、最後の Natural 画面が最初に発行されます。

中断フラグをYに設定すると、会話型のNaturalセッションから非会話型トランザクションコードに切り替えることはできません。切り替えようとすると、Naturalエラーメッセージが発行されます。

次のパラメータを使用できます。

パラメータ	フォーマット/長さ	タイプ
Transaction_code	(A8)	入力
Transaction_type	(A4)	入力 設定可能値： 会話型に対して CONV 非会話型に対して NONC
Suspend_flag	(A1)	入力 設定可能値： Yに設定すると Natural セッションは中断されます 他の値に設定すると Natural セッションは終了します
MOD_name	(A8)	入力
Message	(A1/1:V)	入力
Message_length	(I4)	入力
Return_code	(I4)	出力

特定のリターンコード値：なし。

サンプルプログラム：NIPDEFTX

## NIIDIRT - Natural トランザクションコードへの直接切り替えを用意

指定された Natural トランザクションコードへの直接切り替えを用意します。次の端末の書き込みにおいて、指定されたトランザクションコードへの CHNG コマンドが発行され、Natural 画面が代替 TP PCB を使用して挿入されます。

会話型の Natural セッションから非会話型のセッションに切り替えると、その会話は終了し、MOD\_nameNIIMODNCを使用するダミーメッセージが挿入されます。このメッセージによって一時的に画面の保護が解除され、非会話型の Natural セッションの最初の画面によって上書きされます。

パラメータ	フォーマット/長さ	タイプ
Transaction_code	(A8)	入力
Return_code	(I4)	出力

特定のリターンコード値：なし。

サンプルプログラム：NIPDIRT

## NIIDIRTX - トランザクションコードへの直接切り替えを用意

指定されたトランザクションコードへの直接切り替えを用意します。次の端末の書き込みにおいて、新しいトランザクションコードのCHNG呼び出しが発行され、メッセージまたはSPAが代替TP PCBを使用して挿入されます。このトランザクションタイプは、新しいトランザクションコードのタイプを定義します。

- 会話型トランザクションコードから非会話型に切り替えると、その会話は MOD\_nameNIIMODN を使用するダミーメッセージの発行によって終了されます。これにより画面の保護が一時的に解除され、非会話型トランザクションコードから発行された画面によって上書きされます。
- 中断フラグを Y に設定すると、Natural セッションは中断され、後で再開できます。Natural セッションが再開されると、最後の Natural 画面が発行されます。
- 中断フラグを Y に設定すると、会話型の Natural から非会話型トランザクションコードに切り替えることはできません。切り替えようとすると、Natural エラーメッセージが発行されます。
- メッセージの長さを 0 に設定すると、メッセージは挿入されません。しかし、この操作は会話型トランザクションコードに切り替える場合のみ可能です。

次のパラメータを使用できます。

パラメータ	フォーマット/長さ	タイプ
Transaction_code	(A8)	入力
Transaction_type	(A4)	入力 設定可能値： 会話型トランザクションコードに対して CONV 非会話型トランザクションコードに対して NONC

パラメータ	フォーマット/長さ	タイプ
Suspend_flag	(A1)	入力  設定可能値：  Y に設定すると Natural セッションは中断されます 他の値に設定すると Natural セッションは終了します
Message	(A1/1:V)	入力
Message_length	(I4)	入力
Return_code	(I4)	出力

特定のリターンコード値：なし。

サンプルプログラム：NIPDIRTX

## NIEMOD - モジュール出力ディスクリプタの設定変更

Naturalセッションの最後のメッセージの挿入に使用されるモジュール出力ディスクリプタの現在の設定を変更し、パラメータ MOD\_name で指定された値に設定します。

次のパラメータを使用できます。

パラメータ	フォーマット/長さ	タイプ
MOD_name	(A8)	入力
Return_code	(I4)	出力

サンプルプログラム：NIPEMOD

## NIIGCMD - 以前の IMS/TM コマンドの次の応答セグメントを取得

以前に発行された IMS/TM コマンドの次の応答セグメントを取得します。応答の長さは応答の長さパラメータで返されます。応答が応答エリアに適合しない場合は、応答が切り捨てられ、リターンコード 4 が発行されます。

次のパラメータを使用できます。

パラメータ	フォーマット/長さ	タイプ
Reply_area	(A1/1:V)	入力/出力
Reply_area_length	(I4)	入力
Reply_length	(I4)	出力
Status_code	(A2)	出力
Return_code	(I4)	出力

特定のリターンコード値：4（応答は切り捨てられる）

サンプルプログラム：NIPCMD

## NIIGMSG - 次メッセージの最初のセグメントを取得

GUを発行することによって、メッセージキューから次のメッセージの最初のセグメントを取得します。メッセージエリアには、先頭の LLZZ バイトを含む取得されたメッセージが含まれています。メッセージキューにメッセージがない場合は、LLZZ は 0 に設定されます。

次のパラメータを使用できます。

パラメータ	フォーマット/長さ	タイプ
Message_area	(A1/1:V)	出力
Message_area_length	(I4)	入力
Return_code	(I4)	出力

特定のリターンコード値：なし。

サンプルプログラム：NIPGMSG、NIPGSEG

## NIIGSEG - 入力メッセージの次のセグメントを取得

GN呼び出しを発行することによって、入力メッセージの次のセグメントを取得します。メッセージエリアには、先頭の LLZZ バイトを含む取得されたメッセージが含まれています。現在のメッセージにそれ以上メッセージセグメントがない場合は、LLZZ は 0 に設定されます。



次のパラメータを使用できます。

パラメータ	フォーマット/長さ	タイプ
Message_area	(A1/1:V)	出力
Message_area_length	(I4)	入力
Return_code	(I4)	出力

特定のリターンコード値：なし。

サンプルプログラム：NIPGSEG

## NIIGSPA - SPA からデータを取得

指定されたオフセットから開始する SPA から指定された長さのデータを取得します。

次のパラメータを使用できます。

パラメータ	フォーマット/長さ	タイプ
Offset	(I4)	入力
Length	(I4)	入力
Area	(A1/1:V)	入力/出力
Return_code	(I4)	出力

特定のリターンコード値：4

取得されたデータは、Natural に予約されている SPA に部分的または完全に格納されます。

サンプルプログラム：NIPGSPA

## NIIMSIN - IMS 環境情報を取得

INQY ENVIRON 呼び出しを使用して IMS 環境情報を取得します。Reply\_area\_length を 102 未満に指定すると、応答が切り捨てられて、リターンコード X'0100' および理由コード X'000C' が出力されます。

次のパラメータを使用できます。

パラメータ	フォーマット/長さ	タイプ
Reply_area	(A1/1:V)	出力
Reply_area_length	(I4)	入力
Return_code	(I4)	出力

特定のリターンコード値：*nnxx*

*nn*：最初の2バイトは AIB リターンコードを含みます。*xx*：次の2バイトは AIB 理由コードを含みます。AIB は「アプリケーションインターフェイスブロック」を表し、AIBTDLI インターフェイスを介して IMS を呼び出すときに使用されます。

サンプルプログラム：NIPIMSIN

## NIISRTF - 複数セグメントのメッセージを作成

複数セグメントのメッセージを作成します。NIISRTF は、指定された宛先に対して CHNG 呼び出しを実行し、PURG 呼び出しを実行することなく、最初のメッセージセグメントを挿入します。追加のメッセージセグメントは、NIISRTM を使用して挿入することができます。メッセージは NIIPURG を使用して終了する必要があります。LLZZ バイトはサービスモジュールによって作成されます。

次のパラメータを使用できます。

パラメータ	フォーマット/長さ	タイプ
Destination	(A8)	入力
Message	(A1/1:V)	入力
Message_length	(I4)	入力
Return_code	(I4)	出力

特定のリターンコード値：なし。

サンプルプログラム：NIPISRTM

## NIISRTM - メッセージキューにメッセージセグメントを挿入

CHNG または PURG 呼び出しを実行せずに、メッセージキューに次のメッセージセグメントを挿入します。LLZZ バイトはサービスモジュールによって作成されます。

次のパラメータを使用できます。

パラメータ	フォーマット/長さ	タイプ
Message	(A1/1:V)	入力
Message_length	(I4)	入力
Return_code	(I4)	出力

特定のリターンコード値：なし。

サンプルプログラム：NIPISRTM

## NIPCBAD - PSB 名および PCB アドレスの返却

現在スケジュールされている PSB 名、および論理名によって識別された PCB のアドレスを返します。論理的な PCB 名がトランザクションコードテーブルで定義されていない場合は、Natural エラーメッセージが発行されます。

次のパラメータを使用できます。

パラメータ	フォーマット/長さ	タイプ
PSB_name	(A8)	出力
Logical_PCB_name	(A8)	入力
PCB_address	(B4)	出力
Return_code	(I4)	出力

特定のリターンコード値：なし。

サンプルプログラム：NIPPCBAD

## NIIPCOM - 応答エリアにデータを移動

データエリアで提供されるデータを、NIIBOOTS 呼び出しで指定された、指定のオフセットの応答エリアに、指定された長さで移動します。NIIPCOM は、サーバー環境からのみ呼び出すことができます。

次のパラメータを使用できます。

パラメータ	フォーマット/長さ	タイプ
Offset	(I4)	入力
Data_area	(A1/1:V)	入力
Length	(I4)	入力
Return_code	(I4)	出力

特定のリターンコード値：4（サーバー環境ではなく呼び出し環境）

サンプルプログラム：NIPPCOM

## NIIPMSG - メッセージを送信

指定された MOD\_name を使用して、I/O PCB によって示される宛先にメッセージを送信します。メッセージは指定されたメッセージエリア長のメッセージエリアから取得されます。メッセージエリアには、最初に LLZZ バイトが含まれてはいけません。こうすることで、MFS 形式の出力メッセージを入力メッセージの発行元に送り返すことができます。

次のパラメータを使用できます。

パラメータ	フォーマット/長さ	タイプ
Message	(A1/1:V)	入力
Message_length	(I4)	入力
MOD_name	(A8)	入力
Return_code	(I4)	出力

特定のリターンコード値：なし。

サンプルプログラム：NIPPMMSG

## NIIPSBAD - PSB アドレスの返却

PCB アドレスリストのアドレスである PSB のアドレスを返します。

次のパラメータを使用できます。

パラメータ	フォーマット/長さ	タイプ
PSB_address	(B4)	出力
Return_code	(I4)	出力

特定のリターンコード値：なし。

サンプルプログラム：NIPBOOTS

## NIIPSPA - SPA でのデータ置換

データエリアで提供されるデータによって、指定されたオフセットにある SPA に配置された指定の長さのデータを置換します。

次のパラメータを使用できます。

パラメータ	フォーマット/長さ	タイプ
Offset	(I4)	入力
Length	(I4)	入力
Data_area	(A1/1:V)	入力
Return_code	(I4)	出力

特定のリターンコード値：なし。

SPA のヘッダー（最初の 14 バイト）および/または Natural Reserved Area に存在するデータを上書きする試みは拒否されて、Natural エラーメッセージが発行されます。

サンプルプログラム：NIPPSPA

## NIIPURG - PURG 呼び出しの発行

PURG 呼び出しを発行します。

次のパラメータを使用できます。

パラメータ	フォーマット/長さ	タイプ
Return_code	(I4)	出力

特定のリターンコード値：なし。

サンプルプログラム：NIPISRTM

## NIIRETRM - メッセージエリアにデータを移動

指定のメッセージエリアに、指定されたオフセットで開始する指定された長さの入力メッセージからデータを移動します。

オフセットは LLZZ バイトから計算されます。

次のパラメータを使用できます。

パラメータ	フォーマット/長さ	タイプ
Offset	(I4)	入力
Length	(I4)	入力
Message_area	(A1/1:V)	入力/出力
Return_code	(I4)	出力

特定のリターンコード値：なし。

サンプルプログラム：NIPRETRM

## NIISASD - SENDER と OUTDEST の設定を変更

Natural プロファイルパラメータ SENDER および OUTDEST の現在の設定を変更します。

次のパラメータを使用できます。

パラメータ	フォーマット/長さ	タイプ
Sender	(A8)	入力
Outdest	(A8)	入力
Return_code	(I4)	出力

特定のリターンコード値：なし。

サンプルプログラム：NIPNTRD

## NIIU3962 - セッションの終了

ユーザーアバンドコード U3962 でセッションを終了し、ダンプを生成します。

次のパラメータを使用できます。

パラメータ	フォーマット/長さ	タイプ
Return_code	(I4)	出力

特定のリターンコード値：なし。

サンプルプログラム：NIPU3962

---



# 26

## IMS/TM 環境の Natural - サービスモジュール

---

■ サービスモジュールの目的 .....	192
■ サービスモジュールの説明 .....	192
■ CMCMMND - IMS オペレータコマンドの発行 .....	192
■ CMDEFSW - Natural トランザクションコードへの遅延トランザクション切り替え .....	193
■ CMDEFSWX - Natural 以外のトランザクションコードへの遅延トランザクション切り替え .....	193
■ CMDIRNMX - メッセージのない別の会話型トランザクションへの切り替え .....	194
■ CMDIRNMZ - メッセージのない別の会話型トランザクションへの切り替え .....	194
■ CMDIRSWX - メッセージのある別の会話型トランザクションへの切り替え .....	195
■ CMDIRSWZ - メッセージのある別の会話型トランザクションへの切り替え .....	196
■ CMDISPCB - PCB の内容の取得 .....	197
■ CMEMOD - MOD 名のダイナミックな変更 .....	197
■ CMGETMSG - 次のメッセージの読み取り .....	198
■ CMGETSEG - 次のセグメントの読み取り .....	198
■ CMGETSPA - SPA からのデータ転送 .....	199
■ CMIMSID - MVS サブシステム ID の取得 .....	199
■ CMIMSINF - システム環境情報 .....	199
■ CMPCBADR - PCB アドレスを返す .....	200
■ CMPRNTR - デフォルトのハードコピー出力先の変更 .....	201
■ CMPUTMSG - IO-PCB への出力メッセージの挿入 .....	201
■ CMPUTSPA - SPA へのデータの移動 .....	202
■ CMQTRAN - 現在のトランザクションコードテーブルエントリの内容 .....	202
■ CMQUEUE - 代替 PCB へのメッセージの挿入 .....	203
■ CMQUEUEX - メッセージの内容に対する完全な制御 .....	203
■ CMSNFPRT - 論理デバイス名の設定 .....	204
■ CMSVC13D - Natural セッションの終了 .....	205
■ CMTRNSET - 代替 PCB による SPA の挿入 .....	205
■ NIIDDFS - 外部トランザクションへの遅延切り替え .....	205
■ NIIDPURG - 複数セグメントメッセージの送信 .....	206
■ NIIDQUMS - 複数セグメントメッセージの作成 .....	206
■ NIIDSETT - 外部トランザクションコードの取得 .....	207

この章では、Natural IMS インターフェイスのサービスモジュールについて説明します。

## サービスモジュールの目的

サービスモジュールは IMS/TM 固有の機能を実行します。これらのモジュールは、標準的な Natural CALL インターフェイスを使用して、Natural プログラム内から呼び出すことができます。サンプルプログラムは、Natural INPL によってライブラリ SYSEXTP にロードされます。

## サービスモジュールの説明

このセクションでは、すべてのサービスモジュールについてアルファベット順に詳しく説明します。この説明には、利用可能なパラメータのリストやモジュール関連のサンプルプログラムの名前も含まれています。

## CMCMMND - IMS オペレータコマンドの発行

モジュール CMCMMND は、IMS オペレータコマンドを発行し、応答セグメントを Natural ユーザープログラムに返します。

次のパラメータを使用できます。

名前	フォーマット/長さ	タイプ	コメント
Command		入力	
Command length	(B4)	入力	
Reply		出力	
Length of reply area	(B4)	入力	

コマンドエリアに含まれているオペレータコマンドは、指定した長さで IMS に発行されます。

ユーザーが応答の長さを 0 以外に設定した場合、IMS からの応答セグメントは、利用可能な最大の長さで応答エリアに移動されます。応答エリアに最低 2 バイトの長さがある場合、最初の 2 バイトには、コマンドコールを発行した後の IMS ステータスコードが含まれます。REPLGTH フィールドの右端の 2 バイトには、REPLY フィールドに移動された応答全体の有効な長さが格納されます。

IMS からの応答を切り捨てる必要がある場合、REPLGTH フィールドの左端のバイトを X'80' と設定することでこの処理を指定します。

サンプルプログラム：NIPSCMND

## CMDEFSW-Natural トランザクションコードへの遅延トランザクション切り替え

モジュール CMDEFSW は、Natural トランザクションコードへの遅延トランザクション切り替えを実行します。

次のパラメータを使用できます。

名前	フォーマット/長さ	タイプ	コメント
Trancode		入力	

次の端末 I/O で出力は端末に送信され、この端末からの次の入力は、パラメータメッセージとして渡されたトランザクションコードで処理されます。

## CMDEFSWX-Natural 以外のトランザクションコードへの遅延トランザクション切り替え

モジュール CMDEFSWX は、Natural 以外のトランザクションコードへの遅延切り替えを実行します。

次のパラメータを使用できます。

名前	フォーマット/長さ	タイプ	コメント
Trancode		入力	
Message		入力	
Message length		入力	
MOD name		入力	

次の端末 I/O で、指定した MOD 名で指定したメッセージが挿入され、Natural セッションは終了されます。

新しいトランザクションコードが Natural トランザクションコードである場合、パラメータとして渡されたメッセージと MOD 名は無視され、CMDEFSWX は CMDEFSW と同じように動作します。

サンプルプログラム：NIPSDEFX

## CMDIRNMX - メッセージのない別の会話型トランザクションへの切り替え

代替 PCB にメッセージを挿入しないことを除けば、モジュール CMDIRNMX は [CMDIRSWX](#) と同じ機能です。したがって、ユーザーが指定する必要があるパラメータは Trancode のみです。

次のパラメータを使用できます。

名前	フォーマット/長さ	タイプ	コメント
Trancode		入力	

また CMDIRNMX を使用すると、別の Natural トランザクションコードに直接切り替えることができます。この処理が可能である理由は、この場合、入力メッセージとして CLEAR キーがデフォルトで Natural に渡されるからです。

Natural 以外のトランザクションコードに切り替える場合は、代わりに TERMINATE ステートメントとサービスモジュール [CMTRNSET](#) を使用することを強くお勧めします。

```
CALL 'CMTRNSET' TRANCODE /* set transaction code */
TERMINATE                /* terminate Natural and call TRANCODE */
```

## CMDIRNMZ - メッセージのない別の会話型トランザクションへの切り替え

代替 PCB にメッセージを挿入しないことを除けば、モジュール CMDIRNMZ は [CMDIRSWZ](#) と同じ機能です。したがって、ユーザーが指定する必要があるパラメータは Trancode のみです。

次のパラメータを使用できます。

名前	フォーマット/長さ	タイプ	コメント
Trancode		入力	

## CMDIRSWX-メッセージのある別の会話型トランザクションへの切り替え

モジュール CMDIRSWX は、別の会話型トランザクションへの直接切り替えを実行し、この新しいトランザクションに渡すメッセージを指定します。

次のパラメータを使用できます。

名前	フォーマット/長さ	タイプ	コメント
Trancode		入力	
Message		入力	
Message length	(B4)	入力	

次の端末 I/O で、代替 PCB に対して変更コールが実行され、宛先が Trancode フィールドの値に設定されます。次に SPA とメッセージが代替 PCB に挿入されます。

新しいトランザクションコードは、Natural のトランザクションコードであるか、Natural 以外のトランザクションコードであるかがチェックされます。

Natural 以外のトランザクションコードの場合、Natural セッションが終了します。

Natural トランザクションコードの場合、CLEAR キーが入力メッセージとして Natural に渡されます。つまり、Natural は端末ユーザーが CLEAR キーを押した場合と同様に動作します。新しいトランザクションコードのタイプは自動的に保持されます。

Natural 以外のトランザクションコードに切り替える場合は、代わりに TERMINATE ステートメントとサービスモジュール **CMTRNSET** を使用することを強くお勧めします。

```
CALL 'CMTRNSET' TRANCODE /* set transaction code */
TERMINATE 0 MESSAGE      /* terminate Natural and call TRANCODE with MESSAGE */
```

メッセージ MESSAGE は、Natural 変数 Message の長さでトランザクションコード Trancode に渡されます。TERMINATE ステートメントのリターンコードはゼロになる必要があります。それ以外の場合、Natural セッションは終了エラーメッセージ NAT9987 で終了し、トランザクションコードの切り替えは実行されません。

サンプルプログラム：NIPSDIRX

## CMDIRSWZ - メッセージのある別の会話型トランザクションへの切り替え

モジュール CMDIRSWZ は、CMDIRSWX と同じ機能です。

次のパラメータを使用できます。

名前	フォーマット/長さ	タイプ	コメント
Trancode		入力	
Message		入力	
Message length	(B4)	入力	

CMDIRSWX との違いは、Natural 以外のトランザクションコードに切り替える場合に現在の Natural セッションが終了されないことです。この動作の目的は、次のとおりです。

- 特定の Natural セッションが Natural 以外のトランザクションコードに制御を渡しますが、このセッションは終了しません。
- Natural 以外のトランザクションは端末 I/O を実行し、次に元の Natural トランザクションに再び切り替えてデータを SPA に渡します。
- Natural トランザクションは、新しいセッションを開始しないで、一時的に放置状態になっても、以前のセッションをそのまま維持します。つまり、スワッププールからロールスロットが取得され、既存のセッションで処理を続行できるように制御が Natural に渡されます。

Natural 以外のトランザクションコードは、メッセージ LLZZD を渡す必要があります。この場合、LL=H'0005'、ZZ=X'0000'、および D=X'6D' は、Natural に対して CLEAR キーが押された状態をシミュレートします。CLEAR キーに反応するように Natural プログラムを設定すると、呼び出された Natural 以外のトランザクションが制御を取り戻したことがプログラムで認識され、Natural 以外のトランザクションが準備したデータをプログラムが取得して、それ以降の処理で使用することができます。

切り替えるトランザクションコードが Natural トランザクションコードである場合、CMDIRSWZ は使用することができません。

サンプルプログラム：NIPSDIFS

## CMDISPCB - PCB の内容の取得

モジュール CMDISPCB は、PCB の内容を取得するために使用します。

次のパラメータを使用できます。

名前	フォーマット/長さ	タイプ	コメント
PCB number	(B4)	入力	
Receiving area		出力	
Area length	(B4)	入力	

コールの実行後、受信エリアには、要求した長さで PCB の内容が格納されます。要求した PCB が現在の PCB リストに含まれていることを確認するチェックが実行されます。最初の PCB は PCB 番号 1 であり、第 2 の PCB は PCB 番号 2 となります。無効な番号を指定すると、フィールド PCB number が X'FFFFFFFF' に設定され、アプリケーションプログラムにはそれ以上の情報が渡されません。

サンプルプログラム：NIPSPCBD

## CMEMOD - MOD 名のダイナミックな変更

モジュール CMEMOD は、Natural セッションの正常な終了時に、特定の LTERM に対して MOD name をダイナミックに変更することができます。

次のパラメータを使用できます。

名前	フォーマット/長さ	タイプ	コメント
MOD name	(A8)	入力	

セッションの正常終了時に、環境依存のインターフェイスによって、MOD name パラメータに値が格納された MOD 名で、メッセージ X'00060000403F' が IOPCB に挿入されます。この処理の目的は、端末ユーザーが端末で作業を続けられるように、意味を理解できる画面（例えば、一般的なメニューなど）を表示することです。

## CMGETMSG - 次のメッセージの読み取り

モジュール CMGETMSG は、メッセージキューから次のメッセージを読み取ります。

次のパラメータを使用できます。

名前	フォーマット/長さ	タイプ	コメント
Message area		出力	
Message area length	(B4)	入力	

受信したメッセージがメッセージエリアに収まるかどうかを確認するために、長さがチェックされます。LLZZ バイトを含めて、メッセージはメッセージエリアに移動されます。メッセージがそれ以上ない場合、LL=0 がメッセージエリアに移動されます。

メッセージがメッセージエリアに収まらない場合、対応するエラーメッセージが返されます。

サンプルプログラム：NIPSGETM および NIPSOBMP

## CMGETSEG - 次のセグメントの読み取り

モジュール CMGETSEG は、メッセージキューから現在のメッセージの次のセグメントを読み取ります。

次のパラメータを使用できます。

名前	フォーマット/長さ	タイプ	コメント
Message area		入力	
Message area length	(B4)	入力	

受信したメッセージがメッセージエリアに収まるかどうかを確認するために、長さがチェックされます。LLZZ バイトを含めて、メッセージセグメントはメッセージエリアに移動されます。メッセージセグメントがそれ以上ない場合、LL=0 がメッセージエリアに移動されます。

メッセージがメッセージエリアに収まらない場合、対応するエラーメッセージが返されます。

サンプルプログラム：NIPSOBMP



## CMGETSPA - SPA からのデータ転送

モジュール CMGETSPA は、指定されたオフセットから開始して、要求された長さで SPA から受信エリアにデータを転送します。

次のパラメータを使用できます。

名前	フォーマット/長さ	タイプ	コメント
Offset	(B4)	入力	
Length	(B4)	入力	
Area	(B4)	出力	

サンプルプログラム：NIPSGSPA および NIPSPSPA

## CMIMSID - MVS サブシステム ID の取得

モジュール CMIMSID を使用すると、Natural プログラムは、現在スケジュールが設定されている IMS システムの MVS サブシステム ID を取得できます。

次のパラメータを使用できます。

名前	フォーマット/長さ	タイプ	コメント
IMSID	(A4)	出力	

コールの実行後に、フィールド IMSID には、現在スケジュールが設定されている IMS システムの MVS サブシステム ID が格納されます。

モジュール CMIMSID は、内部の IMS コントロールブロックに依存しています。したがって、これは IMS のリリースによって異なる機能であり、可能な場合には常に更新されます。

## CMIMSINF - システム環境情報

モジュール CMIMSINF は、システム環境情報を提供します。

次のパラメータを使用できます。

名前	フォーマット/長さ	タイプ	コメント
IMSID	(A4)	出力	IMS ID。
SUFFIX	(A2)	出力	プリロードの接尾辞。
APPLGNAM	(A8)	出力	アプリケーショングループ名。
APPLNAM	(A8)	出力	アプリケーション名。
NRENT	(B4)	出力	プリロードされるリエントラントモジュールの数。
NNONR	(B4)	出力	プリロードされる非リエントラントモジュールの数。

CMIMSINF も、IMS のリリースに応じて異なるモジュールです。

サンプルプログラム：NIPSINF

## CMPCBADR - PCB アドレスを返す

モジュール CMPCBADR は、論理名で識別された PCB のアドレスを返します。PSB 名も Natural プログラムに返されます。

次のパラメータを使用できます。

名前	フォーマット/長さ	タイプ	コメント
PSB name	(A8)	入力	
PCB name	(A8)	入力	
PCB address	(B4)	入力	

コールの実行後、フィールド PCBADR には PCB のアドレスが格納されます。このアドレスは、テーブルエントリにある論理名 PCBNAME によってテーブルモジュール内で識別された PCB のアドレスであり、現在スケジュールが設定されているトランザクションコードに対応しています。そのトランザクションコードに対して論理名が存在しない場合、PCBADR フィールドで X'FFFFFFFF' が返されます。いずれの場合でも、フィールド PSBNAME には、現在スケジュールが設定されている PSB の名前が格納されます。

サンプルプログラム：NIPSPCBA

## CMPRNTR - デフォルトのハードコピー出力先の変更

モジュール CMPRNTR は、モジュール NIIMSHC で設定されたデフォルトの印刷先を、パラメータとして渡された値に変更します。

次のパラメータを使用できます。

名前	フォーマット/長さ	タイプ	コメント
Destination	(A8)	入力	

モジュール CMPRNTR は、互換性保持の目的でのみ提供されています。Natural ステートメントの SET CONTROL *hdest-id* を使用することをお勧めします。

## CMPUTMSG - IO-PCB への出力メッセージの挿入

モジュール CMPUTMSG を使用すると、長さを指定した任意の出力メッセージを IO-PCB に挿入することができます。このとき、MFSMOD name を指定します。こうすると、入力メッセージの発行元に、MFS フォーマットの出力メッセージを送り返すことができます。

CMPUTMSG は、メッセージ長に指定されたバイト数のデータをメッセージエリアから取得し、指定された MOD name でメッセージキューにこれらのデータを挿入します。メッセージ長の制限はありませんが、データが環境依存インターフェイスの入力メッセージエリアに収まる必要があります。

次のパラメータを使用できます。

名前	フォーマット/長さ	タイプ	コメント
Message area		入力	
Message length	(B4)	入力	
MOD name		入力	

空白以外のステータスコードが IO-PCB で返された場合、Natural エラーメッセージ NAT8272 が発行されます。このメッセージではステータスコードが変数として格納されています。

## CMPUTSPA - SPA へのデータの移動

モジュール CMPUTSPA は、指定された長さのデータを、指定されたオフセットで SPA に移動します。

次のパラメータを使用できます。

名前	フォーマット/長さ	タイプ	コメント
Offset	(B4)	入力	
Length	(B4)	入力	
Data		入力	

指定したオフセットが SPA 内の Natural Reserved Area (NRA) に含まれていないかどうか、チェックが実行されます。含まれている場合は、リターンコード 4 が返されます。

サンプルプログラム：NIPSPSPA

## CMQTRAN - 現在のトランザクションコードテーブルエントリの内容

モジュール CMQTRAN は、トランザクションコードテーブルにある現在のエントリの内容を返します。

次のパラメータを使用できます。

名前	フォーマット/長さ	タイプ	コメント
Transaction code		出力	実行しているトランザクションコード。
Offset	(B2)	出力	SPA での NRA のオフセット。
Length	(B2)	出力	NRA の長さ。
Uoffset	(B2)	出力	未使用。
PSB name		出力	スケジュールが設定されている PSB の名前。
Number of PCBs		出力	モジュール CMPCBADR を使用してアドレスを取得できる PCB の数。

セキュリティ上の懸念があるため、モジュール CMPCBADR にユーザーが指定できる PCB の論理名は返されません。ユーザーが指定できる論理名は、システムによって通知されます。

サンプルプログラム：NIPSQTRA

## CMQUEUE - 代替 PCB へのメッセージの挿入

モジュール CMQUEUE は、指定された代替 PCB にメッセージを挿入します。

次のパラメータを使用できます。

名前	フォーマット/長さ	タイプ	コメント
Destination		入力	
Message		入力	
Message length	(B4)	入力	
TP PCB number	(B4)	入力	任意

このコールによって、即時変更コールが、指定された代替 PCB の宛先を、フィールド Destination に指定された値に設定します。これ以降、メッセージは指定されたメッセージ長で代替 PCB に挿入されます。

トランザクションコードは、データ長 8 で LLZZ バイトの後に挿入されます。

PURGE 呼び出しの発行後、制御は Natural プログラム内の次の命令に戻されます。

メッセージの最大長は、入力メッセージエリアのサイズです（通常、8000 バイト引く 12 バイト）。

使用する代替 PCB は、最後のオプションパラメータで指定します。呼び出しに TP PCB number を指定していない場合、NIMTRNTG マクロの **ALTPCB** パラメータで指定した代替 TP PCB が使用されます。

サンプルプログラム：NIPSQL0A

## CMQUEUEX - メッセージの内容に対する完全な制御

モジュール CMQUEUEX を使用すると、IMS/TM 入力キューでキューイングするメッセージの内容を完全に制御できます。

次のパラメータを使用できます。

名前	フォーマット/長さ	タイプ	コメント
Destination		入力	
Message		入力	
Message length	(B4)	入力	
TP PCB number	(B4)	入力	任意

このコールによって、即時変更コールが、指定された代替PCBの宛先を、フィールド Destination に指定された値に設定します。これ以降、メッセージは LLZZ バイトの後、指定されたメッセージ長で代替 PCB に挿入されます。CMQUEUE との違いは、LLZZ バイトの後にトランザクションコードが挿入されないことです。

PURGE 呼び出しの発行後、制御は Natural プログラム内の次の命令に戻されます。メッセージの最大長は、入力メッセージエリアのサイズです（通常、8000 バイト引く 12 バイト）。

使用する代替 PCB は、最後のオプションパラメータで指定します。呼び出しに TP PCB 番号を指定していない場合、NIMTRNTG マクロの **ALTPCB** パラメータで指定した代替 TP PCB が使用されます。

サンプルプログラム：NIPSQUEX

## CMSNFPRT - 論理デバイス名の設定

モジュール CMSNFPRT は、オンライン BMP の実行時に Natural メッセージの送信先とするデバイスの論理名を設定します。

次のパラメータを使用できます。

名前	フォーマット/長さ	タイプ	コメント
Printer name		入力	

CMSNFPRT の呼び出し前に、Natural プロファイルパラメータ SENDER を使用してデフォルトの出力先を定義します。

サンプルプログラム：NIPSOBMP

## CMSVC13D - Natural セッションの終了

モジュール CMSVC13D は、ユーザーアベンドコード U3962 で Natural セッションを終了し、ダンプを生成します。

パラメータ：なし

サンプルプログラム：なし

## CMTRNSET - 代替 PCB による SPA の挿入

Natural セッションが正常に終了された場合、Natural IMS インターフェイスは、指定されたトランザクションコードへのプログラム間の直接的な切り替えを実行し、代替 PCB によって SPA を挿入します。

次のパラメータを使用できます。

名前	フォーマット/長さ	タイプ	コメント
Trancode		入力	

サンプルプログラム：NIPSEOSS

## NIIDDEFS - 外部トランザクションへの遅延切り替え

モジュール NIIDDEFS は、モジュール CMDEFSWX によく似ています。NIIDDEFS を使用して外部トランザクションへの遅延切り替えを実行する場合、モジュール CMDIRSWZ の使用時と同様に、現在の Natural セッションは中断されます。中断された Natural セッションは、CLEAR キーを含むメッセージを Natural に返信することでいつでも再開できます。

次のパラメータを使用できます。

名前	フォーマット/長さ	タイプ	コメント
Transaction code		入力	切り替え先のトランザクションコード。
Message		入力	外部トランザクションコードに送信されるメッセージ。
Message length	(B4)	入力	
MOD name	(A8)	入力	

名前	フォーマット/長さ	タイプ	コメント
Transaction type	(A4)	入力	外部トランザクションが会話型の場合は A4 変数に文字列 CONV が格納され、外部トランザクションが非会話型の場合は A4 変数に文字列 NONC が格納されます。

リターンコード：

0	OK
4	メッセージ長が、環境テーブルに定義されたメッセージエリアのサイズよりも長くなっています。
8	会話型の Natural から非会話型の外部トランザクションへ、中断を伴う遅延切り替えを実行しようとして失敗しました。これは正常に実行できません。
12	第 5 のパラメータが無効です。値が CONV でも NONC でもありません。

サンプルプログラム：NIPSDEFS

## NIIDPURG - 複数セグメントメッセージの送信

モジュール NIIDPURG にはパラメータがありません。このモジュールは、NIIDQUMS 呼び出しと同じ代替 PCB を使用して PURGE 呼び出しを発行し、モジュール NIIDQUMS で作成された複数セグメントメッセージを送信します。

リターンコード：4 バイトのリターンコードでは、そのうち 2 および 3 バイトがステータスコードになります。あるいは、リターンコードの値は 0 になります。

サンプルプログラム：NIPSQLMS

## NIIDQUMS - 複数セグメントメッセージの作成

このモジュールは、複数セグメントメッセージを作成します。これは基本的にモジュール CMQUEUE と同じ機能ですが、異なる点は、NIIDQUMS が PURGE 呼び出しを発行しないことです。



次のパラメータを使用できます。

名前	フォーマット/長さ	タイプ	コメント
Destination		入力	
Message		入力	
Message length	(B4)	入力	
TP PCB number	(B4)	入力	任意

モジュール **NIIDPURG** を使用する PURGE 呼び出しの発行は、ユーザーが行います。

使用する代替 PCB は、最後のオプションパラメータで指定します。呼び出しに TP PCB 番号を指定していない場合、NIMTRNTG マクロの **ALTPCB** パラメータで指定した代替 TP PCB が使用されます。

サンプルプログラム：NIPSQLMS

## NIIDSETT - 外部トランザクションコードの取得

外部トランザクションコードへの適切なトランザクション切り替えを実行するには、切り替え先の外部トランザクションコードのタイプを知っている必要があります。このタイプを確認するには、特定用途のモジュール **NIIDSETT** を使用することができます。NIIDSETT が使用されていない場合、外部トランザクションコードは、呼び出し元の Natural トランザクションコードと同じタイプであると判断されます。タイプが異なっていた場合、予期しない結果になったり、セッションが異常終了したりします。

次のパラメータを使用できます。

名前	フォーマット/長さ	タイプ	コメント
Transaction type	(A4)	入力	設定可能値： 会話型の場合は CONV、 非会話型の場合は NONC。



# 27

## IMS/TM 環境の Natural - ユーザー出口

---

■ NIIXACCT .....	210
■ NIIXSTAR .....	210
■ NIIXMSSP .....	210
■ NIIXSSTA .....	210
■ NIIXSRM .....	211
■ NIIXISRT .....	211
■ NIIXTGU0 .....	211
■ NIIXJESA .....	211
■ NIIXPRT0 .....	211
■ NIIXRFNU .....	212
■ NIIXTGNO .....	212

この章では、Natural IMS/TM インターフェイスで使用できるユーザー出口の概要を説明します。それぞれの出口に対しては、同じ名前のソースモジュールが提供されています。ソースモジュールごとに、パラメータリストとレジスタの表記規則を説明します。

## NIIXACCT

---


出口は、アカウントレコードを IMS ログまたは SMF に書き込む前に呼び出されます。したがって、アカウントレコードの内容を変更できるようになっています。NIIXACCT が 0 以外のレジスタ 15 値を返した場合、アカウントレコードはまったく書き込まれません。

## NIIXSTAR

---

SPA とメッセージが取得され、Natural スレッドがロールインされ、圧縮が解除された後は、出口がトランザクションステップのたびに呼び出されます。この出口では、Natural IOCB とドライバワークエリアにアクセスできます。

NIIXSSTA の制御が戻る時にレジスタ 15 内に値 12 が検出されると、Natural IMS インターフェイスによって Natural セッションが強制終了されます。その他の 0 以外の値がレジスタ 15 で検出されると、インターフェイスはレジスタ 15 内に理由コードの値を格納して、Natural IMS インターフェイスエラー 3517 を発行します。

 **注意:** 新しい Natural セッションの開始時には、この出口は呼び出されません。

## NIIXMSSP

---

この出口が呼び出されるのは、マルチセッション機能の使用時に限られます。ここでは、セッション ID の最初の 7 バイトが生成されます。デフォルトの場合、IMS 環境の Natural は LTERM 名を 7 文字に圧縮します。

## NIIXSSTA

---

この出口が呼び出されるのは、新しい Natural ユーザーセッションが開始され、SPA と Natural IOCB が初期化されたときです。この出口では、Natural IOCB とドライバワークエリアにアクセスできます。

NIIXSSTA の制御が戻る時にレジスタ 15 内に値 12 が検出されると、Natural IMS インターフェイスによって Natural セッションが強制終了されます。その他の 0 以外の値がレジスタ 15 で検

出されると、インターフェイスはレジスタ 15 内に理由コードの値を格納して、Natural IMS インターフェイスエラー 3509 を発行します。

---

## NIIXSRM

---

メッセージを IOPCB に挿入する前に、この出口が呼び出されます。

---

## NIIXSRT

---

Natural セッションの終了時であっても、SPA を IOPCB に挿入する前にこの出口が呼び出されます。セッションの終了は、トランザクションコードが空白になった状態で認識できます。

---

## NIIXTGU0

---

この出口が呼び出されるのは、サービスモジュール CMGETMSG の使用時です。ステータスコードとはかかわりなく、NIIXTGU0 は、IOPCB に対する GU 呼び出しの直後に制御を受け取ります。

---

## NIIXJESA

---

この出口が呼び出されるのは、JES API をレポートの書き込みのために使用している場合です。オプションの文字列が作成された後、この出口が呼び出され、オプションの文字列を変更するために使用することができます。

---

## NIIXPRTO

---

この出口が呼び出されるのは、IMS/TM プリンタに直接レポートを書き込む場合です。これは、「フォームフィード」と「改行」のコードを設定するために使用することができます。

## NIIXRFNU

---

新しい Natural セッションがロールファイルに割り当てられると、この出口が呼び出されます。これは、そのセッションで使用されるロールファイルの数を計算するために使用することができます。

## NIIXTGNO

---

この出口が呼び出されるのは、サービスモジュール CMGSEGO または CMGETSEG の使用時です。ステータスコードとはかかわりなく、NIIXTGNO は、メッセージセグメントの取得直後に制御を受け取ります。

# 28

## IMS/TM 環境の Natural - 特別な機能

---

■ 必要条件 .....	214
■ アカウンティング .....	214
■ モニタリング .....	216
■ ブロードキャスト .....	216
■ マルチセッション機能 .....	217
■ サーバー環境 .....	219

この章では、Natural IMS インターフェイスで使用できる特別な機能の使い方について説明します。

## 必要条件

これらの機能の一部では、Natural 認可サービスマネージャ (ASM) が必要です。

- ASM が必要な機能を使用する場合、その機能を使用する前に ASM を開始しておく必要があります。
- ASM で使用される Natural サブシステムは、Natural セッションで使用するサブシステムと同じである必要があります。
- アカウンティングとモニタリングのために、さらに SIP サーバーを有効にしておく必要があります。

## アカウンティング

アカウンティング機能は、ダイアログ指向環境のみで使用することができます。この機能は、環境テーブルパラメータ ACTACTV を YES に設定することでアクティブになります。

Natural IMS パラメータ ACTLOG の設定に応じて、端末 I/O のたびに、特定の Natural セッションに関する情報が IMS ログまたは SMF に書き込まれます。

- ACTACTV パラメータが CMD に設定されている場合、アカウンティングレコードを IMS/TM ログに書き込む /LOG コマンドが発行されます。したがって、すべてのトランザクションコードが /LOG コマンドを使用できるようにする必要があります。各レコードの先頭には、8 バイトのヘッダーが挿入されます。このヘッダーにより、IMS ユーティリティ DFSERA10 を使用してアカウンティングレコードを簡単に選択することができます。ヘッダー文字列は、環境テーブルパラメータ ACTAHDR で定義します。
- ACTACTV パラメータが LOG に設定されている場合、アカウンティングレコードは、LOG 呼び出しを使用して IMS/TM ログに書き込まれます。Natural IMS パラメータ ACTARID で、使用するログコードを指定します。
- ACTACTV パラメータが「SMF」に設定されている場合、アカウンティングレコードは、認可サービスマネージャを使用して SMF に書き込まれます。Natural IMS パラメータ ACTARID で、使用する SMF レコードタイプを指定します。

端末 I/O のたびに、各 Natural ユーザーセッションの次の情報が格納されます。

- ユーザーがアクティブになっている IMS システムの IMS ID、
- セッションが開始された IMS 端末の LTERM 名、
- Natural セッションユーザーのユーザー ID (IOPCB から取得)、



- 現在実行しているダイアログステップの番号、
- 現在アクティブなトランザクションコード、
- 現在アクティブな PSB 名、
- ユーザーがログオンしている現在の Natural ライブラリ名、
- 現在アクティブな Natural プログラム名、
- セッションが中断され、制御が渡されている Natural 以外のトランザクションコード、
- セッションが開始された時刻と日付、
- 最後の ENTER 操作の時刻と日付、
- このセッションの Natural システムファイル (FNAT) の DBID と FNR、
- このセッションの Natural ユーザーファイル (FUSER) の DBID と FNR、
- このセッションの Natural デイクシオナリファイル (FDIC) の DBID と FNR、
- このセッションの Natural Security システムファイル (FSEC) の DBID と FNR、
- このセッションの Natural スプールファイル (FSP00L) の DBID と FNR、
- このセッションの Super Natural システムファイルの DBID と FNR、
- 最後に検出した Natural エラー番号、
- 最後の端末出力の圧縮スレッド長。

この情報は、DSECT NIMACTR によってマップされます。使用する Natural システムファイルの DBID と FNR を格納するために、2つのエリアが存在しています。最初のエリアでは、DBID と FNR のそれぞれに 1 バイトが使用されます。これは、互換性保持の目的でサポートされます。第2のエリアでは、Adabas バージョン6をサポートするために、DBID と FNR のそれぞれにフルワードが使用されます。アカウントングレコードの先頭には、長さバージョンのフィールドが付けられます。

アカウントングレコードが IMS/TM ログや SMF に書き込まれる前に、ユーザー出口 NIIXACCT が呼び出されます。要件に合わせてアカウントングレコードをカスタマイズするには、このユーザー出口を使用することができます。またアカウントングレコードに情報を追加できません。この場合、長さフィールドに新しい長さを設定する必要があります。

アカウントングレコードはコマンドバッファに作成されるため、合計長が Natural IMS パラメータ CMBSIZE の指定値から 17 バイトを引いた値を超えることはできません。許容される最大長はパラメータとして渡されます。

NIIXACCT が 0 以外のレジスタ 15 値を返した場合、アカウントングレコードは書き込まれません。

## モニタリング

モニタリング機能は、ダイアログ指向環境のみで使用することができます。この機能は、環境テーブルパラメータ MONACTV を YES に設定することでアクティブになり、認可サービスマネージャの SIP 機能を使用します。Natural サブシステムは、モニタする Natural セッションで使用されるサブシステムと同じである必要があります。

SYSTP ユーティリティのモニタリング (M) 機能を使用することで、同じ Natural サブシステムを使用するすべての Natural セッションを対象として、その後のアクティビティをモニタリングできます。このユーティリティの詳細については、Natural の『ユーティリティ』ドキュメントの SYSTP を参照してください。また SYSTP セッションでは、同じ Natural サブシステムを使用する必要があります。

## ブロードキャスト

ブロードキャスト機能は、ダイアログ指向環境のみで使用することができます。この機能は、環境テーブルパラメータ BROACTV を YES に設定することでアクティブになり、認可サービスマネージャの SIP 機能を使用します。

ブロードキャストがアクティブな場合、特定の Natural サブシステムの対象ユーザーに対してブロードキャストメッセージを送信できます。対象となるユーザーは次のいずれかです。

- 送信者の接続先になっている Natural サブシステムの全ユーザー、
- メッセージ送信者と同じ IMS システムにある Natural サブシステムの全ユーザー、
- メッセージ送信者と同じ IMS システムにある Natural サブシステムのユーザーで、さらに特定のトランザクションコードに制限されている全ユーザー、
- メッセージ送信者と同じ IMS システムにある Natural サブシステムのユーザーで、さらに Natural アプリケーションに制限されている全ユーザー、
- メッセージ送信者と同じ IMS システムにある Natural サブシステムのユーザーで、さらに Natural アプリケーションと特定の FUSER システムファイルに制限されている全ユーザー。

セッションが端末出力に達すると、セッションがメッセージを受信する必要があるかどうかを判断するためにチェックが実行されます。必要ない場合は、通常の Natural 出力が送信されます。必要がある場合、通常の出力の代わりにメッセージが送信されます。さらに、ENTER キーを押すと、Natural ニュークリアスは、最後の画面を再送信するように命令されます。このように、最初にメッセージが表示され、その後で通常の Natural 出力画面が表示されます。


複数のブロードキャストメッセージがある場合、最後のメッセージが表示されるまで、メッセージが順番に表示されます。最終的に、通常の Natural 出力画面が表示されます。

ブロードキャストメッセージが表示されるのは、メッセージの作成プロシージャで指定された有効期限が過ぎていない場合に限られます。

ブロードキャストメッセージが送信されたら、ENTER キーをもう一度押す前に RESET キーを押す必要があります。使用可能なすべてのアテンション ID には、ENTER キーを押した場合と同じ効果があります。

ユーティリティ SYSTP を使用すると、ブロードキャストメッセージを作成して、送信者の LTERM/IMSID とともに、すべてのアクティブなメッセージの内容を表示できます。メッセージのテキストは、72 バイトに制限されています。

ブロードキャストするメッセージは、SIP サーバーによってメンテナンスされるプールに保存されます。これらのメッセージは、SYSTP ユーティリティを使用して削除するまで、または認可サービスマネージャをシャットダウンするまで、この場所に保持されます。

 **注意:** ブロードキャストメッセージが削除または作成されると、期限切れのすべてのメッセージも同様に削除されます。

## マルチセッション機能

以下では次のトピックについて説明します。

- [マルチセッション機能の働き](#)
- [セッション ID](#)
- [マルチセッションデータベース](#)

### マルチセッション機能の働き


マルチセッション機能は、ダイアログ指向環境のみで使用することができます。この機能は、Natural IMS パラメータ MSACTV を YES に設定することでアクティブになり、同じ論理端末上で最大 9 の Natural セッションを同時に実行できます。

Natural セッションの作成と再開は、PF キーを使用して制御します。Natural IMS パラメータの MSCRKEY と MSRSKEY では、作成キーと再開キーを定義します。

- 作成キーを押した場合、現在の Natural セッションが中断され、新しい Natural セッションが作成されます。
- 再開キーを押した場合、入力フィールドに文字列  $x$  が含まれる場合があります。 $x$  は 1~MSSESMX の数値です。この場合、現在アクティブなセッションが中断され、指定したセッションが再開されます。 $x$  が MSSESMX より大きい値である場合、入力にはアクティブなアプリケーションに渡されます。
- 入力なしで再開キーを押した場合、次に中断されたセッションがラップアラウンド方式で再開されます。例えば、アクティブなセッションがセッション番号 4 である場合、5~MSSESMX の

範囲、または1~3の範囲で次に中断されたセッションが再開されます。中断されたセッションがない場合、入力はアクティブなアプリケーションに渡されます。

- 再開するセッションで現在のセッションとは異なるトランザクションコードが使用されている場合、中断されたセッションを適切なトランザクションコードで再開するために、暗黙的にプログラム間の切り替えが実行されます。
- 会話型の環境と非会話型の環境を併用している場合、MOD名のNIIMODNCとNIIMODMSが環境間の切り替えに使用されます。フォーマット定義は、ソースモジュールNIIMODMSとNIIMODNCで提供され、MFSUTLを使用してMFSに定義する必要があります。
- アクティブなセッションが終了された場合、次に中断されたセッション（存在する場合）が再開キーと同じ順序で再開されます。
- アクティブなセッションがプログラム間の切り替えで終了された場合、中断されたすべてのセッションが破棄されます。
- Natural IMS パラメータ MSMAX が「2」に設定されている場合（2つの並行セッションを許可）、作成キーは再開キーと同じにする必要があります。この状況で作成キーを押すと、1セッションのみがアクティブな場合、第2のセッションが作成され、それ以外の場合はセッションが切り替えられます。

 **注意:** 互換性保持の目的で Natural IMS インターフェイスバージョン 2.2 モードは現在でもサポートされていますが、今後はドキュメントされなくなります。バージョン 2.2 モードは、Natural IMS パラメータ MSCPTB を YES に設定することでアクティブになります。

## セッション ID

セッション ID は、内部で使用されるセッションの一意の ID です。この ID の形式は `XXXXXXxY` のようになります。この場合、`XXXXXXx` は接頭辞であり、`Y` は 1~9 の 10 進数値で表したセッション番号です。

セッションの接頭辞は、論理端末名を 6 バイトの 2 進数 `XXXXXX` に圧縮し、`x` を 2 進数の 0 に設定することで生成されます。

圧縮アルゴリズムが使用中の LTERM 名に適していない場合（この場合は、エラー 3635 が発行されます）、セッション ID の一意な 7 バイト接頭辞を生成するために、ユーザー出口 NIIXMSSP を使用する必要があります。

## マルチセッションデータベース

マルチセッションデータベースは、中断された各セッションのために、SPA の Natural Reserved Area を格納する HDAM ルート専用データベースです。

マルチセッションデータベースを説明するために、ソースモジュール NIIMSDBD にはモデル DBD が含まれており、IMS/TM に対してこれを定義する必要があります。

DBD 名は、パラメータ MSDBD を使用して、Natural IMS パラメータモジュールに指定する必要があります。PCB 番号は、マクロ NIMTRNTG のパラメータ MSPCB を使用して、トランザクションコードテーブルに指定する必要があります。

## サーバー環境


サーバー環境では、CALL インターフェイスを使用し、3GL アプリケーションによって Natural プログラムを実行できます。このインターフェイスは、サポートされているすべての IMS/TM 環境で使用することができ、サーバーコールインターフェイス NIIBOOTS の Natural IMS ドライバ NIISRVD、およびサービス API NIIPCOM で構成されています。

NIISRVD と NIIBOOTS はソースモジュールとして提供されており、サイト上でアSEMBルとリンクエディットを行う必要があります。詳細については、Natural の『インストール』ドキュメントの「Natural IMS インターフェイスのインストール」を参照してください。

サーバー環境では、3GL プログラムから NIIBOOTS を呼び出して Natural セッションを開始することができます。Natural セッションを開始した後は、呼び出し側の 3GL プログラムに戻り、それ以降の入力を待つ状態になります。通常、入力は CMSYNIN から受信することになっています。つまり、3GL プログラムは、Natural のプライマリ入力データセットをシミュレートする必要があります。

サーバー Natural は常に NEXT 行に指定することを強くお勧めします。このような指定にすると、NIIBOOTS に対する次の呼び出しで、Natural コマンドまたは Natural プログラムを実行できるようになります。このように指定しない場合、NIIBOOTS に対する次の呼び出しは、NIIBOOTS に対する前の呼び出しで開始された Natural プログラム向けの入力として扱われます。

メッセージ指向インターフェイスの場合と同様に、通常 CMPRINT に書き込まれるすべての出力は、Natural プロファイルパラメータ SENDER で指定した IMS/TM の宛先に送信されます。Natural IMS インターフェイスで使用する特殊な宛先の詳細については、「IMS/TM 環境の Natural - 環境」の「[SENDER の宛先](#)」を参照してください。

 **注意:** MPP 環境では、リージョンでスケジュールが設定されているすべてのトランザクションが、デフォルトで同じサーバー Natural を使用します。同じ MPP リージョンで複数のサーバー Natural を使用する場合、複数のサーバーコールインターフェイスを生成する必要があります。各サーバーコールインターフェイスは、NIMBOOT パラメータ SERVERN で指定された一意の名前で生成し、一意の名前でリンクする必要があります。SERVERN に指定した名前でロードモジュールに名前を付けることをお勧めします。

## CALL インターフェイス NIIBOOTS

NIIBOOTS は、ドキュメント、および提供されたサンプルプログラムで使用されているデフォルトの名前です。このデフォルトの名前はインストール時に変更できます。

NIIBOOTS では、次のパラメータが必要です。

- PSB アドレス (PCB アドレスリストのアドレス)、
- コマンドエリア、
- 応答エリア。

コマンドエリアでは、次の要素を入力できます。

- 起動パラメータ、
- 任意の Natural コマンドとそれに続く入力データ、
- STAT や REFR など、NIIBOOTS 固有のコマンド (起動パラメータと併用)。

起動パラメータは、2つの連続する 80 バイトのエリアで渡されます。最初のエリアには、次のように環境テーブルの名前と使用するトランザクションコードの名前が格納されます。

```
ENV-TAB=environment-table-name  
TRNCODE=transaction-code-name
```

トランザクションコードが有効になるのは、NIMDRIV マクロに TRNCODE=YES が指定されている場合に限られます。トランザクションコードの使い方については、「IMS/TM 環境の Natural - 環境」の NIMBOOT マクロを参照してください。

第2のエリアには、ダイナミックな Natural パラメータが格納されます。Natural セッションは、このパラメータで開始されます。

この応答エリアは、サービス API NIIPCOM を使用して実行した Natural プログラムから応答が入力されるエリアです。

呼び出されるたびに、NIIBOOTS はサーバー Natural が初期化されたかどうかをチェックします。

- Natural が初期化されていない場合、新しい Natural セッションが開始され、受信したコマンドがダイナミックパラメータとして Natural に渡されます。
- Natural が初期化されている場合、コマンドエリアで受信した文字列は、Natural コマンドまたは Natural プログラムとして Natural に渡されます。

NIIBOOTS 固有のコマンド STAT と REFR は、次のような処理を実行します。

- STAT は、Natural が初期化されていない場合、応答エリアに COLD を返し、初期化されている場合に WARM を返します。
- Natural の現在の状態にかかわらず、REFR は Natural の初期化/再初期化を強制します。

## 推奨される ON ERROR ルーチン

呼び出し側の 3GL プログラムに対して、NIIPCOM を使用して応答エリアで情報を返すために、実行される Natural プログラムでは ON ERROR ルーチンを使用することを強くお勧めします。

## リターンコード

NIIBOOTS は、Natural の終了時に、Natural によって提供されたリターンコードを渡します。

## サンプルプログラム

NIIBOOTS と NIIPCOM の使い方を示すために、サンプルプログラムの NIPBOOTS と NIPPCOM が提供されています。NIPBOOTS は、呼び出し側の 3GL プログラムの役割を果たします。NIPPCOM は、サーバー環境で実行されるサンプル Natural プログラムであり、文字列 NIISVR を応答エリアに書き込みます。ON ERROR ルーチンは、Natural エラー番号を応答エリアに書き込みます。

これらのサンプルプログラムでは、次のシナリオを実行できます。

1. コマンド STAT を渡します。文字列 COLD が応答エリアに返されます。
2. コマンド STACK=(LOGON SYSEXTP),SENDER=S0201 を渡します。S0201 はサーバー Natural で割り当てられるプリンタデバイスの LTERM 名です。Natural が初期化され、ライブラリ SYSEXTP で Natural コマンドを受信する準備が整います。ログオン成功のメッセージが割り当てられたプリンタで発行されます。応答エリアには何も返されません。
3. コマンド STAT を渡します。文字列 WARM が応答エリアに返されます。
4. コマンド NIPPCOM を渡します。プログラム NIPPCOM が実行され、文字列 NIPSRVR が応答エリアに返されます。Natural は、ライブラリ SYSEXTP で次のコマンドを受信する準備が整います。
5. コマンド REFR STACK=(LOGON SYSEXTP;NIPPCOM),SENDER=S0201 を渡します。

Natural が再初期化され、ライブラリ SYSEXTP のプログラム NIPPCOM が実行されます。応答エリアには、文字列 NIPSRVR が書き込まれます。

6. コマンド FIN を渡します。Natural が終了され、情報は応答エリアに渡されません。リターンコードには、Natural 終了時のリターンコードが含まれます。Natural の終了メッセージは、割り当てられたプリンタデバイスで発行されます。
7. コマンド STAT を渡します。文字列 COLD が応答エリアに返されます。





## 29 IMS/TM 環境の Natural - リカバリの制御

---

■ システムアベンドとユーザーアベンド .....	224
■ リバカリ不能エラー .....	224
■ リカバリ可能エラー .....	225

この章では、Natural IMS インターフェイスでのリカバリの制御を説明します。

## システムアベンドとユーザーアベンド

---

Natural IMS インターフェイスは、アベンドの場合に制御を取得する ESTAEX 環境で保護されます。

- ユーザーアベンドが検出された場合、リソースはクリーンアップされ、Natural に制御を渡すことなくアベンドが処理されます。
- システムアベンドが検出された場合、アベンドについて Natural に通知が送信されます。Natural プロファイルパラメータ DU の設定に応じて、Natural はエラーメッセージを送信してセッションを続行するか、またはセッションを終了します。
- NIMDRIV マクロに `LE370=YES` が指定されており、LE プログラムに制御があるときにアベンドが発生した場合、ユーザー記述または言語固有の条件ハンドラが優先されます。アベンドに関する情報は、すべての LE 条件ハンドラで条件が満たされた場合に限り Natural に通知されます。この場合、アベンドは標準的なアベンド処理が実行される前に、次の手順により Natural で制御されます。
  - 対応する LE エラーメッセージが SYSOUT に書き込まれます。
  - LE ランタイムオプション TERMTHDACT に従って、LE スナップダンプが CEEDUMP に書き込まれます。
  - LE は、Natural CALL ステートメントの後で処理を再開するように命令されます。
  - 特別な Natural エラーメッセージ (DU=OFF の場合は NAT0950、DU=ON の場合は NAT9967) が発行され、LE エラー番号が示されます。

いかなる場合でも、エラー発生時の状況を示すダンプを生成できます (レジスタの内容、PSW など)。DU=ON や DU=SNAP の場合、またはユーザーアベンドが要求した場合、ダンプが生成されます。

## リバカリ不能エラー

---

リカバリ不能エラーは、Natural IMS インターフェイスで検出され、Natural で処理できない論理エラーです。これらの状況が発生するのは、一般的に起動時、終了時、または端末 I/O のときです。このいずれの場合も Natural ランタイムはアクティブではないため、エラーに対応することができません。

リカバリ不能エラーが検出された場合、Natural IMS インターフェイスは、NII エラーを発行してセッションを終了します。エラーメッセージは、IMS ログとシステムログにも書き込まれます。エラーメッセージテーブルのダンプオプションに基づいて、スナップダンプが生成されません。

メッセージが IMS ログに書き込まれないようにするには、NIMPARM マクロの `ERRLHDR` パラメータを明示的に空値に設定します。つまり、`ERRLHDR=` を指定します。

エラーメッセージを送信できない場合（例えば、GU が失敗した場合）、セッションがアベンドになります（ユーザーアベンド）。

## リカバリ可能エラー

---

レポートの宛先が無効な場合など、論理エラーが Natural IMS インターフェイスによって検出され、それが Natural で対処できる場合、Natural エラーメッセージが発行され、Natural は標準的なエラー処理でその状況に対処します。



# 30

## TIAM 環境の Natural

---

- Natural TIAM インターフェイスの構造 ..... 228
- マクロ NAMTIAM のパラメータ ..... 229
- TIAM 環境の共通メモリプール ..... 241
- Natural 共有ニュークリアス ..... 241

このドキュメントでは、Natural TIAM インターフェイス（製品コード NRT）の機能、および TIAM 環境での Natural の操作および個別のコンポーネントについて説明します。

以下の項目も参照してください。

- Natural の『インストール』ドキュメントの「*Natural TIAM* インターフェイスのインストール」
- Natural の『オペレーション』ドキュメントの、*BS2000/OSD* 環境での *Natural* に関するセクション

## Natural TIAM インターフェイスの構造

---

Natural TIAM インターフェイスは、次の 2 つのコンポーネントで構成されています。

- リエントラントではないフロントエンド部分
- リエントラント部分に当たる NATRENT（デフォルト）

どちらのコンポーネントも、マクロ *NAMTIAM* の要素であり、2 つの異なるアセンブリの実行で生成されます。「[マクロ \*NAMTIAM\* のパラメータ](#)」のパラメータ *CODE* も参照してください。

フロントエンド部分は、一般的に、Adabas インターフェイスモジュール *ADALNK* とリンクされ、初期化ルーチンを構成します。このルーチンは、TIAM セッションで *Natural* を確立するときに一度だけ実行されます。初期化段階では、対応するパラメータのオペランド値に基づいて、さまざまな機能が実行されます。例えば、*Natural* バッファプールの確立／接続、*Natural* ニュークリアスのロードまたはリンク、物理的な端末バッファの確立などです。フロントエンド部分は、ユーザー（タスク）ごとにロードする必要があります。

リエントラント部分の *NATRENT* は、*Natural* ニュークリアスにモジュール要素としてリンクされます。また、*NATRENT* には、TP システム依存ルーチンのためのさまざまなエントリポイントが含まれます（メモリ管理、端末通信など）。共有の *Natural* ニュークリアスを使用する場合、生成された *NATRENT* モジュールをフロントエンド部分にリンクする必要があります。

*Natural* ニュークリアスは完全に環境から独立しており（共有コード）、すべてのユーザーを対象として一度ロードするだけで済みます。

## マクロ NAMTIAM のパラメータ

マクロ NAMTIAM は、2 回生成する必要があります。1 回は Natural TIAM インターフェイスのフロントエンド部分のためで、もう 1 回はリエントラント部分のためです。どちらで生成するかは、NAMTIAM マクロのパラメータ **CODE** で決定されます。

フロントエンド部分とリエントラント部分での生成に対して、NAMTIAM に対するマクロコールでラベルを提供することができます。このラベルは CSECT 名を定義しており、この名前でもジュールがモジュールライブラリに格納されます。ラベルが指定されていない場合、フロントエンド部分の名前は NATFRONT になり、リエントラント部分の名前は NATRENT になります。

フロントエンド部分の **NAMTIAM** マクロの例：

```
NATTESTF NAMTIAM CODE=FRONT,NUCNAME=NB2RENT,PARMOD=31
```

この例では、フロントエンド部分の CSECT 名は NATTESTF として定義されます。

リエントラント部分の **NAMTIAM** マクロの例：

```
NATTESTR NAMTIAM CODE=RENT,CLRKEY=K4,PARMOD=31
```

この例では、リエントラント部分の CSECT 名は NATTESTR として定義されます。

パラメータ：

マクロ NAMTIAM に指定できる各パラメータを次に示します。

ADACOM | ADDBUFF | APPLNAM | ATTKEY | CLRKEY | CODE | CURPRO | DELETE | DYNPAR | HCASK | ILCS  
| LF | LINK | LINK2/LINK3/LINK4 | NUCNAME | PARMOD | PFK | REFRKEY | REQMLOC | SYSDTA | TRACE  
| TTYLS | TTYPS | T975X | UMODE | USERID |

### ADACOM - Adabas リンクモジュールの使用

このパラメータは、フロントエンド部分の生成に適用されます。このパラメータは、使用する Adabas リンクモジュールを決定します。可能な値は次のとおりです。

値	説明
ADAUSER	モジュール ADAUSER がフロントエンド部分にリンクされます (Adabas バージョン 7.1 未満)。
ADALNN	マルチパスを使用している場合、モジュール ADALNN はフロントエンド部分にリンクされます (Adabas バージョン 7.1 未満)。
ADABAS	モジュール ADAUSER および SSFB2C がフロントエンド部分にリンクされます (Adabas バージョン 7.1 以降)。
ADALNK	モジュール ADALNK (Adabas バージョン 7.1 未満)、またはモジュール ADALNK、ADAL2P、および SSFB2C (Adabas バージョン 7.1 以降) がフロントエンド部分にリンクされます。

値	説明
	これはデフォルト値です。

### ADDBUFF - 追加のページ数

このパラメータは、フロントエンド部分の生成に適用されます。このパラメータは、端末 I/O バッファの追加のページ数を決定します。可能な値は次のとおりです。

値	説明
1~8 またはオペランドなし	1 ページ当たり 4 KB の追加のページ数。
オペランドなし	デフォルトでは、オペランドが指定されていません。

### APPLNAM - アプリケーション名

このパラメータは、フロントエンド部分の生成に適用されます。可能な値は次のとおりです。

値	説明
<i>name</i>	Natural TIAM アプリケーションの名前（最大 8 文字）。Natural TIAM タスクが初期化される とき、この名前はシリアライゼーション ID の一部になります。
NATTIAM	これはデフォルト値です。

### ATTKEY - 割り込みモード

このパラメータは、リエントラント部分の生成に適用されます。可能な値は次のとおりです。

値	説明
ON	端末で K2 キーを押すと、STXIT ルーチンによる割り込みが実行されます。Natural は ATTENTION INTERRUPT を作成し、NAT1016 エラーメッセージを返します。
OFF	K2 キーを押すと、通常の BS2000/OSD 割り込みが実行されます。  これはデフォルト値です。



## CLRKEY - 代替的なクリアキー

このパラメータは、リエントラント部分の生成に適用されます。

このパラメータは、LSP と DUE1 に加えて、代替的な CLEARKEY を定義するために利用できます。可能な値は次のとおりです。

値	説明
K1～K4、F1～F4、DUE2	代替的な CLEARKEY の名前。
K4	これはデフォルト値です。

## CODE - 生成モード

このパラメータは、フロントエンド部分とリエントラント部分の両方の生成に適用されます。

このパラメータは、生成する Natural TIAM インターフェイスの部分を決定します。可能な値は次のとおりです。

値	説明
FRONT	フロントエンド部分の生成／アセンブリを示します。 これはデフォルト値です。
RENT	リエントラント部分の生成／アセンブリを示します。

## CURPRO - 保護フィールドへのカーソルの移動

このパラメータは、フロントエンド部分の生成に適用されます。

このパラメータは、保護フィールドへのカーソルの移動を許可するかどうかを制御します。可能な値は次のとおりです。

値	説明
ON	カーソルは保護フィールドには移動できません。 これはデフォルト値です。
OFF	カーソルを保護フィールドに移動できます（例えば、フィールド固有のヘルプ機能を表示するため）。

## DELETE - ダイナミックにロードされたプログラムの削除

このパラメータは、リエントラント部分の生成に適用されます。可能な値は次のとおりです。

値	説明
ON	Natural パラメータモジュールでプロファイルパラメータ DELETE を設定すると、ダイナミックにロードされた Natural 以外のプログラムを、ロード元の Natural プログラムの終了時にアンロードするかどうかを制御できます。また、このようなプログラムをコマンドモードへの切り替え時にアンロードするかどうかを制御できます。  これはデフォルト値です。
OFF	一度ダイナミックにロードされた Natural 以外のプログラムは、Natural セッション全体が存続している間保持されます。

## DYNPAR - ダイナミックパラメータの読み取り

このパラメータは、フロントエンド部分の生成に適用されます。可能な値は次のとおりです。

値	説明
DIALOG	ダイナミックパラメータは、端末での入力から読み取られます。
YES	ダイナミックパラメータは、端末での入力から読み取られます。YES の機能は DIALOG と同じです。これは、互換性維持の目的で保持されています。
SYSDTA	ダイナミックパラメータは SYSDTA から読み取られます。
SYSIPT	ダイナミックパラメータは SYSIPT から読み取られます。
FILE	ダイナミックパラメータは、シーケンシャルファイルから読み取られます。この SAM ファイルの入力は、1 つのテキスト文字列として認識されます。つまり、個々のエントリは、行末であってもコンマで他の文字列と区切る必要があります。このようなパラメータファイルは、LINK パラメータ CMPRMIN を使用して FILE コマンドで定義する必要があります。  次の例も参照してください。
NO	ダイナミックパラメータは、端末での入力から読み取られません。  これはデフォルト値です。

DYNPAR=FILE の例：

```
/FILE NAT.PARAMS, LINK=CMPRMIN
```

## HCASK - ハードコピーの出力デバイスの指定

このパラメータは、リエントラント部分の生成に適用されます。

このパラメータは、ハードコピーの出力のたびに、使用する出力デバイスをユーザーが指定するようになるかどうかを決定します（端末コマンド %H を使用します）。可能な値は次のとおりです。

値	説明
ON	ユーザーは、ハードコピーの出力のたびにデバイスを指定するように求められます。 これはデフォルト値です。
OFF	前のハードコピー出力で使用したデバイスが再び使用されます。

## ILCS - 3GL サブプログラムの呼び出し

このパラメータは、リエントラント部分の生成に適用されます。可能な値は次のとおりです。

値	説明
CRTE	3GL サブプログラムは、一般的なランタイム環境の規則に従って呼び出されます。呼び出すことができるように、ILCS 初期化ルーチン ITOSL# を、次に示すように Natural フロントエンドにリンクする必要があります。  INCLUDE ITOSL#,SYSLNK.CRTE.010 RESOLVE,SYSLNK.CRTE.010
YES	3GL サブプログラムは、機能拡張された ILCS のリンク規則に従って呼び出されます。呼び出すことができるように、ILCS 初期化ルーチン ITOINITS を、次に示すように Natural フロントエンドにリンクする必要があります。  INCLUDE ITOINITS,SYSLNK.ILCS RESOLVE,SYSLNK.ILCS
NO	標準的な処理が適用されます。  これはデフォルト値です。

## LF - 行送りの制御文字

このパラメータは、フロントエンド部分の生成に適用されます。

このパラメータでは、ローカルプリンタでの印刷のために行送りの制御文字を指定します。可能な値は次のとおりです。

値	説明
X'zz'	制御文字。
X'25'	これはデフォルト値です。

## LINK - プログラムとモジュールのリンク

このパラメータは、フロントエンド部分の生成に適用されます。可能な値は次のとおりです。

値	説明
<i>name</i> ( <i>name</i> , <i>name</i> , ...)	Naturalプログラムから呼び出され、非リエントラント部分とリンクされるプログラムとモジュールの <i>name(s)</i> は、このパラメータで指定する必要があります。下記を参照してください。

デフォルト値はありません。

逆に、名前を指定するプログラムおよびモジュールは、非リエントラント部分とリンクする必要があります。そうしないと、アプリケーションのステータスが SYSTEMERROR になり、すべてのユーザーが拒否され、エラーメッセージが表示されます。

TABLE マクロコールは、指定されたプログラムとモジュールに対して実行され、ダイナミックローダーのリンクテーブルにロードアドレスが入力されます。したがって、Naturalプログラムで呼び出すときは、これらのプログラムをダイナミックにロードする必要はありません。

例：

```
LINK=PROG1
LINK=(PROG1,PROG2,MODUL111)
```

## LINK2/LINK3/LINK4 - LINK パラメータオペランドの拡張

これらのパラメータは、フロントエンド部分の生成に適用されます。

パラメータ LINK2、LINK3、および LINK4 は LINK パラメータの拡張です。オペランドの定義は 127 文字以下（カッコを含めて）にする必要があるため、これらのパラメータはパラメータ LINK のオペランドが長くなりすぎる場合のために提供されています。構文は LINK の構文と似ています。可能な値は次のとおりです。

値	説明
<i>name</i> ( <i>name</i> , <i>name</i> ,...)	Natural プログラムから呼び出され、非リエントラント部分とリンクされるプログラムとモジュールの名前は、このパラメータで指定する必要があります。

デフォルト値はありません。

例：

```
NAMTIAM LINK=(PROG1,PROG2,...),
          LINK2=(PROG54,...)
```

```
NAMTIAM LINK=(PROG1,PROG2,PROG3,PROG4)
```

## NUCNAME - Natural のリエントラントモジュールの名前

このパラメータは、フロントエンド部分の生成に適用されます。

このパラメータでは、連結される Natural のリエントラントモジュールの名前を指定します。マクロ ADDON の Natural プールとロード情報には、この名前を使用する必要があります（BS2STUB はマクロ ADDON をアセンブルします）。可能な値は次のとおりです。

値	説明
<i>name</i>	連結される Natural リエントラントモジュールの名前。
NB2RENT	これはデフォルト値です。

## PARMOD - Natural TIAM アプリケーションのアドレスモードと場所

このパラメータは、フロントエンド部分とリエントラント部分の両方の生成に適用されます。

このパラメータの最初の部分 *nn* は、Natural TIAM アプリケーションに対して、アドレッシングモード（24 ビットまたは 31 ビットモード）の定義に使用されます。

このパラメータの第 2 の部分 *loc* は、Natural TIAM アプリケーションのフロント部分の場所を定義するために使用されます。16 MB を超えるアプリケーションのフロント部分をロードする場合、フロント部分のリンクプロシージャで次のように定義する必要があります。

```
LOADPT=*XS
```

```
LOADPT=X 'address'
```

可能な値は次のとおりです。

値	説明
<i>nn</i>	24/31
<i>loc</i>	BELOW/ABOVE
(31, ABOVE)	これはデフォルト値です。

例：

```
/EXEC TSOLINK
PROG NATvrs, FILENAM=NATvrs, LOADPT=*XS, ...
TRAITS RMODE=ANY, AMODE=31
INCLUDE.../*
PARMOD=(nn, loc) MUST BE IDENTICAL IN THE FRONT-END AND REENTRANT PARTS
```

## PFK - ファンクションキーモード

このパラメータは、フロントエンド部分の生成に適用されます。これは、次のファンクションキーモードのいずれかを設定するために使用されます。

値	説明
(KN, <i>y</i> )	リテラル %K1~%K20 と送信キーコード DÜ1、または送信キーコード F1~F20 が、ファンクションキーにロードされます。これは、デバイスタイプによって決まります。  この場合、 <i>y</i> は L または N のいずれかです。「L」はファンクションキーがロードされたことを表し、N は対応するモードがアクティブになったが、ファンクションキーがロードされないことを表します。
(K0, <i>y</i> )	リテラル 01~20 および送信キーコード F5 がファンクションキーにロードされます。

値	説明
(KS, y)	リテラル A~T および送信キーコード F5 がファンクションキーにロードされます。さらにすべての出力メッセージで、ダミーフィールドが画面の最後の2つの位置で生成されます。このダミーフィールドは、キー値を受信して渡すために使用されます。
OFF	ファンクションキーモードは生成されません。
(KS, L)	これはデフォルト値です。

## REFRKEY - リフレッシュキー

このパラメータは、リエントラント部分の生成に適用されます。

これはファンクションキーの定義に使用することができます。このファンクションキーを押すと、最後の完全な Natural 画面がリフレッシュされます。したがって、オペレータまたはオペレーティングシステムから受信したメッセージで画面が上書きされた後でも、Natural のダイアログでの操作を続行できます。可能な値は次のとおりです。

値	説明
K1~K14	ファンクションキーの指定。
N0	ファンクションキーが定義されていません。
K14 (キー ESC +:)	これはデフォルト値です。

送信キーコードは、Natural アプリケーションに渡されません。インターフェイスによって、Natural キーコードが ENTER に設定されます。

REFRKEY パラメータで定義するキーは、CLRKEY パラメータで定義するキーとは別にする必要があります。

## REQMLOC

このパラメータは、31ビットモード (PARMOD=31) のフロントエンド部分とリエントラント部分の両方の生成に適用されます。

このパラメータは、システムが要求メモリを通じて割り当てる Natural ワークエリアの場所を決定します。可能な値は次のとおりです。

値	説明
BELOW	すべてのエリアは、16 MB よりも下で要求されます。
ABOVE	すべてのエリアは、16 MB よりも上で要求されます。
RES	すべてのエリアは、リエントラント部分の場所に応じて要求されます。 これはデフォルト値です。

REQMLOC パラメータは、BS2000/OSD システムマクロ REQM の LOC パラメータに対応します。

## SYSDTA

このパラメータは、フロントエンド部分の生成に適用されます。可能な値は次のとおりです。

値	説明
PRIMARY	ダイナミックな Natural パラメータを SYSDTA から読み取った後、SYSDTA は SYSFILE SYSDTA=(PRIMARY) に設定されます。  これはデフォルト値です。
SYSCMD	ダイナミックな Natural パラメータを SYSDTA から読み取った後、SYSDTA は SYSFILE SYSDTA=(SYSCMD) に設定されます。

## TRACE - トレースファイル番号と出力レコード長

このパラメータは、リエントラント部分の生成に適用されます。

このパラメータでは、トレースファイルの番号とトレース出力レコードの最大長を指定します。可能な値は次のとおりです。

値	説明		
( <i>nn</i> , <i>ll</i> )	<i>nn</i> :	01...99	<i>nn</i> は SYSLST <i>nn</i> トレースファイルの番号です。
	<i>ll</i> :	71...132	<i>ll</i> はトレース出力レコードの最大長を表す文字数です。
(99,71)	これはデフォルト値です。		

外部 Natural トレース機能がアクティブな場合は、トレースレコードは SYSLST*nn* に書き込まれます。この場合、Natural TIAM ドライバによって次のトレースファイルが作成されます。

例：

```
NATURAL.TRACE.TIAM.TTTT,SPACE=(30,3)
SYSFILE SYSLSTnn=NATURAL.TRACE.TIAM.TTTT
/* TTTT is the task sequence number
```

Natural TIAM セッションが終了する前に、トレースファイルが次のように閉じられます。

```
SYSFILE SYSLSTnn=(PRIMARY)
```



## TTYLS - telex マシンの行の長さ

このパラメータは、フロントエンド部分の生成に適用されます。

このパラメータでは、Natural の物理的な行の長さを、telex マシンのさまざまな用紙フォーマットに合わせるすることができます。可能な値は次のとおりです。

値	説明
<i>nn</i>	<i>nn</i> は、TTY デバイスの物理的な行サイズを指定します。
80	これはデフォルト値です。

## TTYPS - telex マシンのページサイズ

このパラメータは、フロントエンド部分の生成に適用されます。

このパラメータでは、Natural の物理ページのサイズを、telex マシンのさまざまな用紙フォーマットに合わせるすることができます。可能な値は次のとおりです。

値	説明
<i>nn</i>	<i>nn</i> は、TTY デバイスの物理ページサイズ（行数）を指定します。
24	これはデフォルト値です。

## T975X - デバイスタイプ固有のメッセージの最適化

このパラメータは、フロントエンド部分の生成に適用されます。

このパラメータは、9750 として PDN に生成されたデータステーションを使用する際に、最適化されるデバイスタイプメッセージを決定します。可能な値は次のとおりです。

値	説明
9750、9755、9756、または 9763	デバイスタイプ。
9750	これはデフォルト値です。

## UMODE - オペレーションのモード

このパラメータは、フロントエンド部分の生成に適用されます。可能な値は次のとおりです。

値	説明
(S,y)	S はオペレーションモードを指定します。y は言語インジケータです。 以下の「オペランドの説明」を参照してください。
(S,E)	これはデフォルト値です。

### オペランドの説明

オペランド	説明
S	「オペレーションモードはシングル」という意味です。つまり、1つの端末/ユーザーからは、1回の Natural セッションのみを開始できます。
y	ログオフメッセージ用の言語インジケータです。
D	デンマーク語
E	英語
F	フランス語
G	ドイツ語
I	イタリア語
N	オランダ語
S	スペイン語

## USERID - Natural ユーザー ID

このパラメータは、フロントエンド部分の生成に適用されます。可能な値は次のとおりです。

値	説明
SYSTEM または YES	Natural ユーザー ID は、BS2000/OSD ユーザー ID を使用して作成されます。
USER または NO	Natural ユーザー ID は、ジョブ名、つまり LOGON コマンドの /.JOBNAME を使用して作成されます。LOGON コマンドに BS2000/OSD ジョブ名が指定されていない場合、Natural ユーザー ID は、USERID=SYSTEM または USERID=YES の場合と同様に作成されます。  これはデフォルト値です。

## TIAM 環境の共通メモリプール

---

ローカルな共通メモリプールを生成する場合、または、グローバルな共通メモリプールへの接続を定義する場合は、マクロ ADDON（これはモジュール BS2STUB をアセンブルします）を使用します。

プログラム CMPSTART および CMPEND は、グローバルな共通メモリプールの開始と停止を実行します。これらのプログラムについては、Natural の『オペレーション』ドキュメントの「グローバル共通メモリプール」で説明しています。

Natural TIAM アプリケーションには、次の共通メモリプールが必要です。

- **Natural** ロードプール

Natural のリンクされたリエントラント部分は、この共通メモリプールにロードされます。

- **Natural** バッファプール

実行可能な Natural プログラムと Natural グローバルデータエリアは、この共通メモリプールにロードされます。オブジェクトがリエントラントであるこれらのコンパイルされた Natural プログラムは、このメモリプールから実行されます。

- **Natural/Adabas** ニュークリアス通信メモリプール

Natural は、起動時に Adabas によって確立された追加的な共通メモリプールに接続します。

共通メモリプールとフロントエンドワークエリアに割り当てられるメモリの合計は、仮想ユーザーアドレススペースに完全に収まる必要があります。

Adabas プールがユーザーアドレススペースを超えている場合、OP コマンドの実行時に、エラーメッセージ 148 が生成されます。セッションの開始時に、Natural ではエラーメッセージ NAT8148 が発行され、次のセッションの終了時にはメッセージ NAT9989（不適切なシステムファイル）が発行されます。

## Natural 共有ニュークリアス

---

TIAM アプリケーションに対しては、一般的な共有の Natural ニュークリアスを使用することができます。この場合に適用されるルールは、Natural の『オペレーション』ドキュメントの「BS2000/OSD 環境での Natural 共有ニュークリアス」で説明しています。



# 31 TSO 環境の Natural

---

- Natural TSO インターフェイスに関する全般的な情報 ..... 244
- Natural TSO インターフェイスの生成パラメータ ..... 244
- TSO 環境の Natural で使用されるデータセット ..... 248
- TSO 環境の Natural からの TSO コマンドの発行 ..... 251

このドキュメントでは、Natural TSO インターフェイス（製品コード NTI）の機能、および z/OS オペレーティングシステムの TSO 環境での Natural の操作および個別のコンポーネントについて説明します。

以下の項目も参照してください。

- Natural の『インストール』ドキュメントの「Natural TSO インターフェイスのインストール」

## Natural TSO インターフェイスに関する全般的な情報

---

Natural TSO インターフェイス (NATTSO) は、z/OS オペレーティングシステムとのインターフェイスとなるいくつかのサービスルーチンで構成されています。

NATTSO はソースモジュールとして提供されており、要件に対応するためにカスタマイズできます。Natural の『インストール』ドキュメントの「Natural TSO インターフェイスのインストール」も参照してください。NATTSO をアSEMBルして Natural ニュークリアスにリンクすることも、NATTSO を共有ニュークリアスで接続して個別に実行することもできます。

NATTSO は完全にリエントラントであり、16 MB 境界より上で実行できます。1 つの TSO リージョンで複数の Natural セッションを並行して開始したり、SWAPKEY（以下を参照）によりセッション間で切り替えを行ったりすることができます。

## Natural TSO インターフェイスの生成パラメータ

---

NATTSO マクロには、Natural TSO インターフェイスの内部デフォルトを変更する複数の生成パラメータがあります。

これらのパラメータは次のとおりです。

ABEXIT | ALTSCRN | LBPNAME | LE370 | NDBFSRV | SUBPOOL | SWAPKEY | TIOBSZ | TTYTYPE

### ABEXIT - アベンド処理モード

このパラメータは、Natural 内でのアベンド処理のモードを指定します。

可能な値は次のとおりです。

値	説明
ESTAE	Natural はすべてのアベンドをインターセプトし、適切なエラーメッセージを発行します。 これはデフォルト値です。
SPIE	Natural バージョン 2.1 の場合と同様に、プログラムチェック (SOCx アベンド) のみがインターセプトされます。
NONE	Natural はアベンドまたはプログラムチェックを一切インターセプトしません。この値は、Natural プロファイルパラメータ DU=FORCE に対応します。  設定 ABEXIT=NONE は、アベンドインターセプトを必要とする一部の機能が動作しなくなるためお勧めしません。プロファイルパラメータ MT の使用によって、CPU 時間制限に到達したときに、エラー NAT0953 の代わりにアベンド U0322 が発生します。

### ALTSCRN - セッション画面モード

このパラメータは、3270の代替画面サイズを使用するかどうかを指定します。端末には、デフォルトと代替の画面サイズとして、画面の高さと幅を定義した2種類のVTAM LOGMODE 定義があります。通常、デフォルトの画面サイズは24行、80列です。代替の画面サイズは、3270 端末モデル (2、3、4または5) に応じて異なります。

可能な値は次のとおりです。

値	説明
YES	代替の画面サイズを使用します。 これはデフォルト値です。
NO	デフォルトの画面サイズを使用します。

Natural プロファイルパラメータ TMODEL では、画面サイズを上書きすることができます。

### LBPNAME - ローカルバッファプールの共有

このパラメータは、ローカルバッファプールの共有を制御します。共有ローカルバッファプール環境の *name* を定義し、ローカルバッファプールを検索および同期するために使用します。

可能な値は次のとおりです。

値	説明
<i>name</i>	<i>name</i> の長さは 1~8 文字です。
LBPNAME=	デフォルト値は "なし" で、ローカルバッファプールは共有されません。

バッチの z/OS 環境または TSO 環境で、複数の Natural セッションを同時に実行する場合、例えば、Natural RPC サーバーを実行する場合、各セッションでそれぞれのローカルバッファプールに対してストレージが割り当てられます。Natural z/OS バッチサーバーは例外ですが、ローカルバッファプールはデフォルトでは共有されません。つまり、異なるセッションで同じ Natural オブジェクトを使用する場合、セッションごとにこれらのオブジェクトを個別にロードする必要があります。 *name* を指定した場合は、すべてのローカルバッファプールが異なる Natural セッションで共有されます。

### LE370 - IBM 言語環境の使用

このパラメータは、Natural を IBM 言語環境 (LE) で実行するかどうかを指定します。

可能な値は次のとおりです。

値	説明
YES	LE の呼び出し規則に従って、外部のサブプログラムを呼び出すことができます。
NO	言語環境のメインプログラムのみを呼び出すことができます。  これはデフォルト値です。つまり、CALL ステートメントごとに新しい LE エンクレーブが作成され、個別に終了されます。
LE370=POSIX	POSIX セマンティクスの LE 呼び出し規則に従って外部サブプログラムを呼び出すことができます。LE はランタイムオプション POSIX(ON) で初期化されます。
LE370=AMODE24	アドレッシングモード 24 でリンクされる外部サブプログラムのサポート。LE はオプション ALL31=(OFF) および STACK=(, ,BELOW) で初期化されます。
LE370=NOHDLR	LE サブプログラムの呼び出し時に、Natural は LE エラーハンドラの設定を行いません。つまり、LE サブプログラムの実行時に対応不能なエラーが発生すると、LE エンクレーブが終了されて、Natural セッションが失われることとなります。

次に示すように、複数のパラメータ値はカッコで囲みます。

```
LE370=(YES,POSIX,AMODE24)
```

IBM 言語環境で実行する Natural の詳細については、Natural の『オペレーション』ドキュメントの「LE サブプログラム」を参照してください。



## NDBFSRV - Natural for DB2 ファイルサーバー

このパラメータは、Natural for DB2 ファイルサーバーを使用するかどうかを指定します。

可能な値は次のとおりです。

値	説明
NO	Natural for DB2 ファイルサーバーは使用しません。 これはデフォルト値です。
YES	Natural for DB2 ファイルサーバーは、端末 I/O ごとに呼び出されます。

## SUBPOOL - GETMAIN 要求用のストレージサブプール

このパラメータは、GETMAIN 要求用のストレージサブプールを定義します。

可能な値は次のとおりです。

値	説明
0 - 127	GETMAIN 要求に対して、z/OS サブプール番号を指定します。問題状況のプログラムに対して任意の有効な z/OS タスク関連のサブプール番号 (0~127) を使用することができます。
0	これはデフォルト値です。

## SWAPKEY - TSO/ISPF 画面分割機能のサポート

このパラメータでは、Natural の TSO/ISPF 画面分割スワップキーを定義します。Software AG 製品の Natural ISPF (Integrated Structured Programming Facility) に対応したほとんどのパネルで、このキーは PF9 に割り当てられます。

可能な値は次のとおりです。

値	説明
PFnn PAnn	次の TSO/ISPF セッションへのスワップに使用する PF または PA キーを定義します。有効なキーは、PF1~PF24 と PA1~PA3 です。  <b>注意:</b> 指定されたキーは、Natural アプリケーションでは使用できません。通常、TSO/ISPF の SWAP コマンドは、ほとんどの TSO/ISPF パネルで PF9 キーに割り当てられます。画面分割をサポートするには、ISPF ロードライブラリの TSO/ISPF インターフェイスモジュール ISPLINK を、NATTSO のリンクステップに含める必要があります。
SWAPKEY=	デフォルトでは、スワップキーは定義されません。つまり、画面分割機能のサポートは生成されません。

## TIOBSZ - 端末 I/O バッファのサイズ

このパラメータでは、端末 I/O バッファのサイズを指定します。16 MB 境界よりも下に割り当てられます。

TIOBSZ=nnnn	nnnn は、端末 I/O バッファのサイズ (バイト単位) です。使用できる値は 4096～16777216 です。デフォルト値は 8192 です。
-------------	---

『オペレーション』ドキュメントの「Natural z/OS 生成パラメータ」の生成パラメータ TIOBSZ1 および TIOBSZ2 も参照してください。

## TTYTYPE - TTY 端末のデフォルトのデバイスタイプ

このパラメータは、テレタイプ端末 (TTY または TWX) のデフォルトのデバイスタイプを定義します。

可能な値は次のとおりです。

値	説明
name	テレタイプ端末のデフォルトのデバイスタイプを定義します。指定するタイプは、NATCONFIG (Natural コンフィグレーション) モジュール内で定義する必要があります。
TTY	これはデフォルト値です。

TTYTYPE は、セッション中に、プロファイルパラメータ TTYPE または端末コマンド %T= によって上書きできます。

## TSO 環境の Natural で使用されるデータセット

Natural TSO セッションで特定の機能を使用する場合、次のデータセットが必要です。

CMEDIT	Software AG Editor ワークファイル
CMHCOPY	ハードコピー出力
CMPLOG	ダイナミックプロファイルパラメータレポート出力
CMPRMIN	ダイナミックプロファイルパラメータ入力
CMPTnn	追加レポート 01～31
CMTRACE	外部トレース出力
NATRJE	ジョブサブミット出力
STEPLIB	外部モジュール用のロードライブラリ
CMWKFnn	ワークファイル 01～32

次に、これらのデータセットについて説明します。

特に記述がない限り、デフォルトの DCB RECFM/LRECL 情報は次のとおりです。

シーケンシャル入力データセットの場合は RECFM=FB と LRECL=80

シーケンシャル出力データセットの場合は RECFM=FBA と LRECL=133

## CMEDIT - Software AG Editor ワークファイル

ローカルまたはグローバルの Software AG エディタバッファプールを使用する場合は、Software AG Editor のワークファイル VSAM データセットが必要です。JCL または TSO コマンド ALLOC で定義されていない場合、Natural ではプロファイルパラメータ EDBP のサブパラメータ DSNAME またはパラメータマクロ NTEDBP で指定されたエディタワークファイルの名前を使用して、エディタワークファイルのダイナミックアロケーションを行います。

または、プロファイルパラメータ EDPSIZE を使用して、エディタワークファイルを必要としない補助エディタバッファプールで実行することもできます。Software AG Editor のインストールの詳細については、Natural の『インストール』ドキュメントの「Software AG Editor のインストール」を参照してください。

## CMHCOPY - ハードコピー出力

ハードコピー出力データセットのデフォルト名は CMHCOPY です。これは、次のいずれかの方法で変更できます。

- 出力ファイル 0 用のプロファイルパラメータ PRINT の DEST サブパラメータ、
- PRINT=((0),DEST=...) と同等のプロファイルパラメータ HCDEST、
- セッション中のシステム変数 \*HARDCOPY の設定、
- セッション中の端末コマンド %H

出力ファイル 0 用の PRINT プロファイルパラメータのサブパラメータは、ハードコピーデータセットのデフォルト値を変更するために使用することができます。デフォルトデータセット名 CMHCOPY は、ハードコピー出力データセットに対して CLOSE=FIN の意味を持っています。つまり、出力のためにデータセットを開いた後、それ以降のハードコピー出力データセット名の変更は無視されます。開いた時点で別の名前が定義されている場合、ハードコピーデータセットは次の端末 I/O で閉じられます。

セッション中に、ハードコピーデータセットの解放とダイナミックアロケーションによる再割り当て（開く前、または閉じた後）を実行できます（Natural アプリケーションプログラミングインターフェイス USR2021N を参照）。

## CMPLOG - ダイナミックプロファイルパラメータレポート出力

プロファイルパラメータ PLOG=ON が設定され、データセット CMPLOG が使用可能な場合は、評価されたダイナミックプロファイルパラメータはセッションの初期化時にこのデータセットに書き込まれます。データセット CMPLOG を使用できない場合、評価されたダイナミックプロファイルパラメータは、行モードで TSO 端末に書き込まれます。

## CMPRMIN - ダイナミックプロファイルパラメータ入力

可能である場合、セッションの初期化時にこのデータセットが読み取られ、ダイナミックプロファイルパラメータを取得します。ダイナミックプロファイルパラメータ文字列の生成には、各レコードの最初の 72 文字のみが使用されます。

Natural ニュークリアス開始のために直接渡されるその他のプロファイルパラメータは、TSO CALL コマンドなどによって、CMPRMIN の入力から生成されたパラメータ文字列の末尾に連結されます。つまり、これらの値は、CMPRMIN のパラメータを上書きするために使用できます。

## CMPRTnn - 追加レポート 01~31

これらのデータセットは、WRITE (nn) などの Natural 出力ファイルステートメントで使用できます。DCB 情報 (例えば、RECFM、LRECL、BLKSIZE) を使用できない場合、デフォルトはプロファイルパラメータ PRINT または Natural パラメータモジュールのマクロ NTPRINT で定義されます。出力ファイルデータセットの名前はサブパラメータ DEST で上書きできます。

## CMTRACE - 外部トレース出力

プロファイルパラメータ ETRACE=ON が設定されているか、等価の端末コマンド %TRE+ が発行された場合は、セッション時のすべての Natural トレース出力が CMTRACE データセットに書き込まれます。トレース対象の Natural コンポーネントを定義するには、プロファイルパラメータ TRACE が必要です。

データセット CMTRACE が使用できない場合は、以下のようにダイナミックに割り当てられます。

```
//CMTRACE DD SYSOUT=*
```

この処理は、最初のトレースレコードの書き込み時に行われます。

## NATRJE - ジョブサブミット出力

このデータセットは、Natural ジョブサブミットユーティリティに使用します。定義されていない場合は、以下のようにダイナミックに割り当てられます。

```
//NATRJE DD SYSOUT=(A,INTRDR)
```

この処理は、最初のジョブのサブミット時に行われます。

## STEPLIB - 外部モジュール用のロードライブラリ

STEPLIB は、外部モジュールをロードするためのデフォルトロードライブラリ名です。外部モジュールには、共有ニュークリアス（プロファイルパラメータ NUCNAME）、個々の Adabas リンクルーチンモジュール（プロファイルパラメータ ADANAME）、セッションバックエンドプログラム（プロファイルパラメータ PROGRAM）、Natural パラメータモジュールにリンクされていない外部サブプログラムなどがあります。

ロードライブラリ名はプロファイルパラメータ LIBNAM で上書きできます。指定するロードライブラリ名は、TSO ジョブ制御または ALLOC ステートメント（例えば、Natural セッションを開始する CLIST など）で定義されている必要があります。

## CMWKFnn - ワークファイル 01~32

これらのデータセットは、READ WORKnn や WRITE WORKnn などの Natural ワークファイルステートメントで使用できます。JCL またはデータセットの VTOC エントリで DCB 情報（RECFM、LRECL、BLKSIZE など）を使用できない場合、デフォルトは WORK プロファイルパラメータ、または Natural パラメータモジュールの NWORK マクロで定義されます。ワークファイルデータセットの名前はサブパラメータ DEST で上書きできます。

## TSO 環境の Natural からの TSO コマンドの発行

ライブラリ SYSEXTP 内の Natural サンプルプログラム TSO を使用して、次のように TSO コマンドを発行できます。

```
TSO LISTALC STATUS
```

パラメータなしで TSO を入力すると、TSO コマンドの入力を指示するメニューが表示されます。このメニューを閉じるには、最初の位置にピリオド (.) を入力するか、PF3 キーを押します。



# 32 openUTM 環境の Natural - 概要

---

このドキュメントでは、Natural UTM インターフェイス（製品コード NUT）の機能、および openUTM 環境での Natural の操作および個別のコンポーネントについて説明します。



**注意:** 以降の説明では、openUTM を通常は UTM と記載します。

## パート 1

- **Natural UTM** インターフェイスの構造
- メッセージのフォーマット - **FREXIT**
- **UTM** アプリケーションでの **Natural** の組み込み
- 共通メモリプール
- その他のストレージエリア
- **KDCROOT** の生成
- **UTM** リソースの定義 - **KDCDEF**
- **UTM DC** トランザクション出口ルーチン **NUERROR**
- **UTM** 起動機能
- **UTM** シャットダウン機能

## パート 2

- **NATUTM** マクロキーワードパラメータ
- **NATUTM** マクロエントリ
- **NURENT** マクロキーワードパラメータ

### パート 3

- ユーザー出口
- UTM 環境での非同期トランザクション処理
- UTM 環境での印刷
- Natural 以外のプログラムの呼び出し
- チェーンで結ばれた部分的な UTM プログラムの呼び出し
- Natural UTM アプリケーションでの Natural 以外のプログラムからの Adabas の呼び出し
- UTM タスクの異常終了

### パート 4

- Natural UTM アプリケーションのアカウントティング
- UTM 環境の Natural で使用するユーティリティプログラム
- Software Exchange
- UTM TACCLASS の概念 (プライオリティ制御)
- Natural UTM アプリケーションの生成
- Natural UTM アプリケーションの最適化
- 複数のアプリケーションと 1 つの共通 Natural
- ダイナミック Natural パラメータの入力と定義
- UTM ユーザーの再スタート
- Adabas プライオリティ制御

以下の項目も参照してください。

- Natural の『インストール』ドキュメントの「Natural UTM インターフェイスのインストール」
- 『メッセージおよびコード』ドキュメントの「UTM 環境下の Natural のエラーメッセージ」
- 『メッセージおよびコード』ドキュメントの「CICS および openUTM 環境下で有効な Natural スワッププールマネージャのエラーメッセージ」
- Natural の『オペレーション』ドキュメントの、BS2000/OSD 環境での Natural に関するセクション
- 『プログラミングガイド』の「インターネットおよび XML アクセス用のステートメント」



*vrs* または *vr* の表記

次のドキュメントでは、*vrs* または *vr* の表記は関連する *version*, *release*, *system maintenance level* 番号を表しています。



# 33 UTM 環境の Natural - パート 1

---

▪ Natural UTM インターフェイスの構造 .....	258
▪ メッセージのフォーマット - FREXIT .....	259
▪ UTM アプリケーションでの Natural の組み込み .....	261
▪ 共通メモリプール .....	261
▪ その他のストレージエリア .....	264
▪ KDCROOT の生成 .....	266
▪ UTM リソースの定義 - KDCDEF .....	267
▪ UTM DC トランザクション出口ルーチン NUERROR .....	269
▪ UTM 起動機能 .....	269
▪ UTM シャットダウン機能 .....	270

## Natural UTM インターフェイスの構造

---

Natural UTM インターフェイスは、マクロ NATUTM、BS2STUB、NURENT に加えて、特別な要件に対応できるいくつかのユーティリティプログラムで構成されています。

- フロントエンド部分 - マクロ NATUTM
- リエントラント部分 - マクロ NURENT

### フロントエンド部分 - マクロ NATUTM

マクロ NATUTM は、パラメータの適切なオペランド定義に基づいて、特定の用途に合うように Natural UTM インターフェイスのフロントエンド部分を生成するのに使用します。通常は、パラメータのデフォルト値を使用します。こうすると、最初の生成でパラメータを変更せずに利用できます。

フロントエンド部分は、UTM タスクごとに一度生成され、主に次のコンポーネントで構成されます。

- UTM の KDCROOT、
- アセンブルされたマクロ NATUTM、
- アセンブルされたマクロ BS2STUB、
- フォーマット出口モジュール FREXIT、
- Adabas インターフェイスモジュール。

### リエントラント部分 - マクロ NURENT

Natural UTM インターフェイスのリエントラント部分は、マクロ NURENT のアセンブルで生成されます。これは Natural UTM アプリケーションのリエントラント部分とリンクされます。共有 Natural ニュークリアスを使用する場合、生成された NURENT モジュールをフロントエンド部分にリンクする必要があります。

Natural UTM アプリケーションのリエントラント部分は、次のコンポーネントで構成されます。

- NATINV (アドレスモジュール) (最初のモジュールとして組み込む必要があります)、
- Natural ニュークリアス、
- Natural バッファプールマネージャ、
- NURENT (アセンブルされたマクロ NURENT の CSECT 名)、
- NATSWPMG (Natural スワッププールマネージャ)、
- Natural パラメータモジュール、
- NATLAST (末尾の定義) (最後のモジュールとして組み込む必要があります)。

Natural UTM インターフェイスのリエントラント部分は、クラス 4 ストレージまたはクラス 6 ストレージの共通メモリプールにロードされる場合、Natural UTM アプリケーション（リエントラント）で一度だけ生成されます。後者のクラスのストレージをお勧めします。

可能性としては、リエントラント部分は、Natural UTM アプリケーションのリエントラントではないフロントエンド部分とリンクされる場合が多くなります。

NATUTM、NURENT、およびすべてのユーティリティプログラムをアセンブルする場合、Natural と UTM マクロライブラリが必要です。

## メッセージのフォーマット - FREXIT

- フォーマット出口モジュール FREXIT
- FREXIT マクロ

### フォーマット出口モジュール FREXIT

メッセージをVDUに送信する場合、Naturalは独自のフォーマットルーチンを使用します（UTM フォーマットタイプ「マイナス」）。メッセージはフォーマット出口モジュール FREXIT で処理されます（論理 I/O ドメインから物理 I/O ドメインへの転送、または逆方向の転送、RESTART および LOGOFF メッセージの生成など）。

モジュール FREXIT は、Natural UTM アプリケーションのフロントエンド部分とリンクする必要があり、KDCROOT または KDCDEF の生成時には、フォーマット出口モジュールとして定義されている必要があります。

例：

```
PROGRAM FREXIT,COMP=ASSEMB
EXIT PROGRAM=FREXIT,USAGE=FORMAT
```

プログラム FREXIT は、LOGOFF メッセージのために、フォーマット名 -END をサポートします。マクロ NATUTM のパラメータ LOFFMAP の説明を参照してください。フォーマット名 -END を使用し、LOGOFF メッセージが出力されると、それ以降は UTM 管理コマンド（KDCINF、KDCSHUT N など）を入力できなくなります。LOGOFF メッセージはフォーマットモードで出力されますが、UTM では行モードの管理コマンドを想定しているため、何を入力しても構文エラーになります。このエラーメッセージを受信した後、すべての有効な管理コマンドは管理 ID で入力できます。プログラム FREXIT で適切なテキスト定数を変更することで、非同期メッセージに対するメッセージ、RESTART および LOGOFF は、特定の要件に合わせて変更できます。

プログラム FREXIT にはユーザー出口 INPTEX があり、ユーティリティプログラム INPTEX で要件を満たすことができます。「ユーティリティプログラム」セクションのプログラム NATDUE および INPTEX に関する説明を参照してください。

プログラム FREXIT のその他のユーザー出口は TRMIOEX であり、入力/出力メッセージの制御に使用することができます。

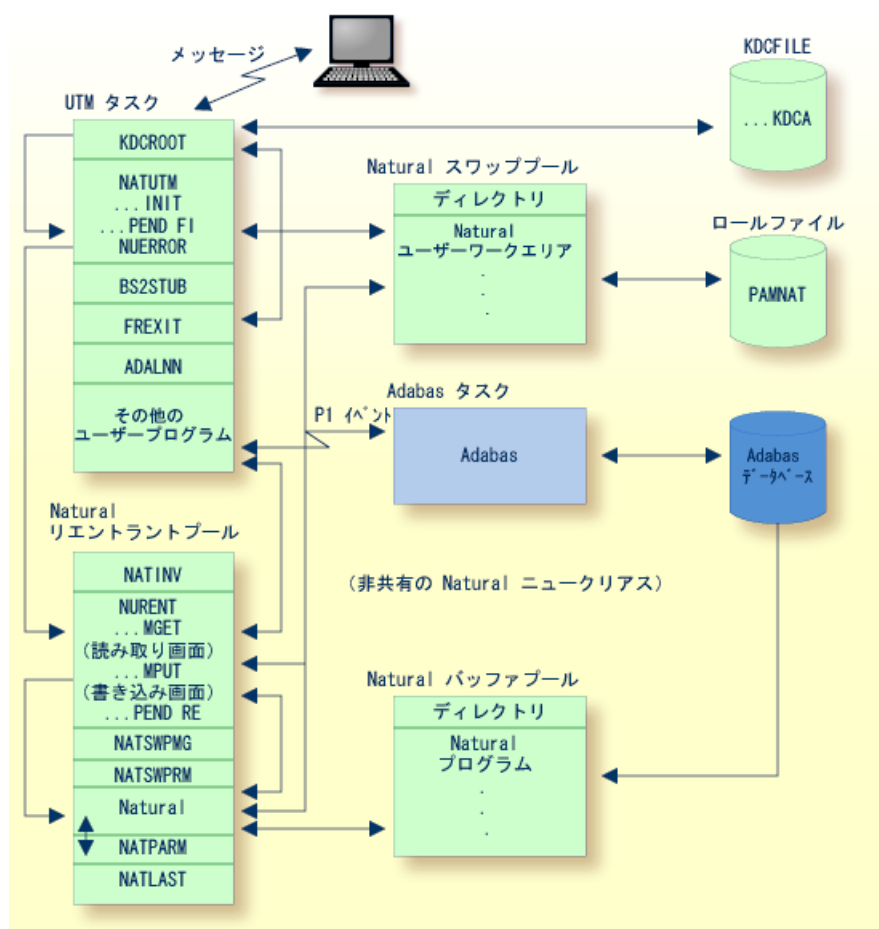
## FREXIT マクロ

マクロ FREXIT には、次のパラメータが含まれています。

AMSG=ASAP	「フリーランニング」（非同期）メッセージが存在する場合、コマンド KDCOUT でこれらのメッセージが事前に読み取られた場合に限って、Natural で詳細なダイアログボックスを表示できます。
AMSG=WAIT	（デフォルト）コマンド KDCOUT を使用して「フリーランニング」メッセージを読み取っていない場合でも、Natural で詳細なダイアログボックスを表示できます。
KDCDISP=YES	（デフォルト）KDCDISP は再スタートメッセージによりサポートされ、自動的に ENTER します。最後の画面出力が更新されます。
KDCDISP=NO	KDCDISP は再スタートメッセージによりサポートされ、更新された次の画面が表示されます。

マクロ FREXIT のデフォルトオペランドを変更する場合、FREXIT を再アセンブルする必要があります。

## UTM アプリケーションでの Natural の組み込み



## 共通メモリプール

次のトピックについて説明します。

- UTM 環境の Natural バッファプール
- UTM 環境の Natural スワッププール
- 共通メモリプールでの Natural のロード - Natural ロードプール

■ Natural Monitor プール

## UTM 環境の Natural バッファプール

Natural では、Adabas データベースから Natural プログラムを読み取って実行するための共通エリアが必要です。この共通メモリプールが、Natural バッファプールです。

ローカルな Natural バッファプールを定義したり、グローバルな Natural バッファプールへの接続を定義したりするには、マクロ ADDON（これはモジュール BS2STUB をアセンブルします）のパラメータを使用します。詳細については、Natural の『オペレーション』ドキュメントの「ADDON マクロ」を参照してください。

グローバルな Natural バッファプールを定義するには、モジュール CMPSTART のパラメータを使用します。このモジュールの詳細については、Natural の『オペレーション』ドキュメントの「CMPSTART プログラム」を参照してください。

バッファプールの統計情報を表示するには、Natural ユーティリティ SYSBPM を使用します。Natural の『ユーティリティ』ドキュメントを参照してください。

## UTM 環境の Natural スワッププール

オンライン Natural ユーザーごとに、Natural ユーザーワークエリアが必要です。ユーザーが何らかのダイアログトランザクションを開始する場合、このユーザーワークエリアは、常にコンピュータのメインストア内に確立する必要があります。ユーザーワークエリアがスワップファイルにロールアウトされ、再びロールインされる頻度を軽減するために、Natural スワッププールをセットアップすることができます。スワッププールの詳細については、Natural の『オペレーション』ドキュメントの「Natural スワッププール」を参照してください。

ローカルな Natural スワッププールを定義したり、グローバルな Natural スワッププールへの接続を定義したりするには、マクロ ADDON（これはモジュール BS2STUB をアセンブルします）のパラメータを使用します。詳細については、Natural の『オペレーション』ドキュメントの「ADDON マクロ」を参照してください。

グローバルな Natural スワッププールを定義するには、モジュール CMPSTART のパラメータを使用します。詳細については、Natural の『オペレーション』ドキュメントの「CMPSTART プログラム」を参照してください。



## 共通メモリプールでの Natural のロード - Natural ロードプール

Natural UTM アプリケーションのリエントラント部分は、クラス 4 ストレージにロードすることができ、また Natural UTM アプリケーションのフロントエンド部分とリンクできます。または、クラス 6 ストレージの共通メモリプールにもロードできます。この後者の方法をお勧めします。共通メモリプールで必要とされるストレージの容量は、Natural UTM アプリケーションのリンクされるリエントラント部分のサイズによって決まります。これは、リンク機能のリストで確認することができます。Natural がクラス 6 ストレージの共通メモリプールにロードされる場合、マクロ NATUTM のパラメータ NUCNAME が使用されます。このパラメータは、リンクされるリエントラントな Natural ニュークリアスの名前を指定します。これは Natural ロードプールの名前でもあります。「**NATUTM マクロパラメータ**」も参照してください。

ローカルな Natural ロードプールを定義したり、グローバルな Natural ロードプール（共有 Natural ニュークリアス）への接続を定義したりするには、マクロ ADDON（これはモジュール BS2STUB をアセンブルします）のパラメータを使用します。詳細については、Natural の『オペレーション』ドキュメントの「ADDON マクロ」を参照してください。

グローバルな Natural ロードプール（共有 Natural ニュークリアス）を定義するには、モジュール CMPSTART のパラメータを使用します。詳細については、Natural の『オペレーション』ドキュメントの「CMPSTART プログラム」を参照してください。

## Natural Monitor プール

Natural Monitor ユーティリティには、データストレージ用の共通メモリプールが必要です。この共通メモリプールは、Monitor ユーティリティをアクティブにすると割り当てられ、Monitor ユーティリティを非アクティブにすると解放されます。

ローカルな Natural Monitor プールを定義したり、グローバルな Natural Monitor プールへの接続を定義したりするには、マクロ ADDON（これはモジュール BS2STUB をアセンブルします）のパラメータを使用します。詳細については、Natural の『オペレーション』ドキュメントの「ADDON マクロ」を参照してください。

グローバルな Natural Monitor プールを定義するには、モジュール CMPSTART のパラメータを使用します。詳細については、Natural の『オペレーション』ドキュメントの「CMPSTART プログラム」を参照してください。

Monitor ユーティリティの詳細については、Natural の『ユーティリティ』ドキュメントの「SYSTP ユーティリティ」を参照してください。

## その他のストレージエリア

---

- Natural ユーザーズレッド
- Natural ユーザーワークエリアの非同期書き込みバッファ
- 非同期トランザクション用の Natural ユーザーエリア
- Natural ロールファイル - LINK=PAMNAT

### Natural ユーザーズレッド

各 UTM タスクに対して、サイズが MAXSIZE であるストレージエリアが生成されます。このエリアには、Natural ユーザーエリアが非圧縮形式で格納されます。

### Natural ユーザーワークエリアの非同期書き込みバッファ

Natural ユーザーワークエリアへの書き込みには、非同期的な書き込み（「待機のない書き込み」）と同期的な書き込み（「待機のある書き込み」）があります

非同期オプションを使用する場合（これがデフォルトオプション）、パラメータ `ROLLTSZ` に定義したオペランドのサイズで、UTM タスクごとに書き込みバッファが生成されます。この方法では、圧縮されたユーザーワークエリアがスワッププールから書き込みバッファにコピーされ、非同期書き込みが開始されるので、すぐに処理を続行することができます。このオプションでは、パフォーマンスが改善されますが、ストレージが増加します。

ロールアウトを同期的に実行する場合、パラメータ `ROLLACC` に値 `UPAM-SY` を指定する必要があります。この場合、書き込みバッファを割り当てる必要はありません。ユーザーワークエリアが正常にスワップファイルに書き込まれるまで、処理は中断されます。

### 非同期トランザクション用の Natural ユーザーエリア

Natural UTM アプリケーションでは、MAXSIZE のストレージエリアが、非同期トランザクションごとに割り当てられます（そのトランザクション用の Natural ユーザーワークエリア）。これはトランザクションの終了時に解放されます。Natural スワッププールは、非同期トランザクションと関連付けられたユーザーワークエリアの格納には使用されません。非同期的に実行するすべての Natural プログラムは、`TERMINATE` ステートメントで終了する必要があります。つまり、UTM DC トランザクションは `PEND 'FI(NISH)'` で終了します。これはアプリケーション内、および 2 つの Natural UTM アプリケーション間の両方で、非同期トランザクションに適用されます。「[非同期トランザクション処理](#)」も参照してください。

## Natural ロールファイル - LINK=PAMNAT

Natural ユーザーワークエリアのスワップには、PAM ファイルが必要です。このファイルでの書き込みと読み取りは、物理的にチェーンで結ばれた PAM-I/O によって実行されます。ただし、これはスワップファイルがエクステントの境界を超過しない場合に限り可能になります。この状態は、SPCCNTRL を使用してチェックできます。

Natural スワップファイルの LINK 名は PAMNAT です。ロールファイルのサイズは、次のように計算できます。

$$NP = ([ (MS + 4 + 31) / 32 ] * 32 * NT + 4) / 2$$

上記の意味は次に示すとおりです。

NP	PAM ページでのデータセットのサイズ
MS	切り上げで端数を切り捨てた KB を単位とするパラメータ ROLLTSZ
NT	オンラインの端末の数

例：

ROLLTSZ = 80 KB (1 ユーザー当たり)、オンラインの端末の数 = 40

$$\begin{aligned} \text{Size of dataset} &= ( [ ( 80 + 4 + 31 ) / 32 ] * 32 * 40 + 4 ) / 2 \\ &= ( [ 115 / 32 ] * 32 * 40 + 4 ) / 2 \\ &= ( [ 3.59375 ] * 32 * 40 + 4 ) / 2 \\ &= ( 3 * 32 * 40 + 4 ) / 2 \\ &= 3844 / 2 \\ &= 1922 \text{ PAM pages} \end{aligned}$$

FILE statement:

```
/FILE NATUTM.SWAPFILE, LINK=PAMNAT, SPACE=(1922,96)
```

ローカルのスワッププールを使用している場合、各 Natural UTM アプリケーションには、独自の Natural スワップファイルが必要です。ユーザーがアプリケーションにログオンすると、Natural UTM インターフェイスによって、Natural ロールファイル内に新しいユーザーが使用できるスペースが十分にあるかがチェックされます。十分なスペースがない場合は、エラーメッセージ NUS0033 が出力されます。

グローバルなスワッププールを使用している場合、同じグローバルスワッププールに接続されているすべての Natural UTM アプリケーションは、同じ Natural ロールファイルを使用する必要があります。

## KDCROOT の生成

Natural UTM アプリケーションに対して KDCROOT を生成する場合は、次の Natural 固有の定義を入力する必要があります。

```
MAX KB=400,SPAB=8192,NB=5120,TRMSGLTH=5120 (see Note 1)
PROGRAM NUSTART,COMP=ASSEMB (see Note 2)
PROGRAM NUERROR,COMP=ASSEMB (see Note 2)
TAC NAT,PROGRAM=NUSTART,EXIT=NUERROR (see Note 2)
EXIT PROGRAM=FREXIT,USAGE=FORMAT (see Note 3)
PROGRAM FREXIT,COMP=ASSEMB (see Note 3)
```

注	
1	<p>UTM KB に必要なエリアは、最小長が 400 バイトです。KDCDEF の MAX パラメータのオペランド KB=nnn に必要な KB 長は、次のように計算する必要があります。</p> <p>固定 KB 長の 400 バイト            + ユーザー拡張の KB 長 (パラメータ KBUSEXT)            + ダイナミックパラメータ保存エリアの長さ (パラメータ SVDYPRM)            + (MULTI-PASS が使用される場合のみ) セッションキーエリアの長さ (この値は、<math>n*72</math> で計算する必要があります。この場合、<math>n</math> は並行セッションの数から 1 を引いた数です)</p> <p>UTM I/O エリアの NB と TRMSLGTH には、5,120 バイトの長さが必要です。</p>
2	<p>Natural UTM アプリケーションでは、規則として、ユーザー固有の部分的な UTM プログラムは 1 つに限られています。</p> <p>このプログラムは、Natural UTM インターフェイスのフロントエンド部分であり、マクロ NATUTM のパラメータ CSECT のオペランドに指定された名前 (デフォルトは NUSTART) で、KDCDEF の適切なパラメータに定義する必要があります。</p> <p>命名規則に従っている限り、任意の数の UTM トランザクションコードを割り当てることができます。</p> <p>DC トランザクション出口ルーチン NUERROR の名前は、Natural UTM インターフェイスのフロントエンド部分、およびその他の部分的な各 UTM プログラムに対して定義する必要があります。</p>
3	<p>フォーマット出口モジュール FREXIT は、パラメータ EXIT と PROGRAM で定義する必要があります。</p>

KDCROOT の生成に関連するその他のすべての定義は、UTM に固有な定義であるか、適切な NATUTM マクロパラメータのオペランドに定義された値に基づいています。

## UTM リソースの定義 - KDCDEF

UTM リソースを定義するときは、次のような Natural 固有の規則に従う必要があります。

- 9755/9756 タイプ端末の特別な定義
- K キーと F キーの操作
- IBM 3270 タイプ端末のサポート
- TTY 端末のサポート

### 9755/9756 タイプ端末の特別な定義

PTERM コマンドの TERMN オペランドは、9755 タイプ端末の場合は値 X1 または FG に、9756 タイプ端末の場合は値 X2 に設定する必要があります。これらは特別な値であり、Siemens UTM ドキュメントの表では説明されていません。

その他のすべてのタイプの端末では、TERMN オペランドを表に示された値に設定する必要があります。

例：

```
PTERM ss19,lterm=ltdf1900,pronam=vr,ptype=t9755,TERMN=X1
```

### K キーと F キーの操作

Natural UTM インターフェイスでは、ファンクションキーの K1、K2、K3、K4、F1、F2、F3、F4、F5 (P キーの場合) がサポートされます。押したファンクションキーは、UTM リターンコードによって識別することができます。このリターンコードは、KDCDEF の SFUNC ステートメントを使用して定義する必要があります。

```
SFUNC K1,RET=26Z
SFUNC K2,RET=27Z
SFUNC K3,RET=28Z
SFUNC K4,RET=29Z
SFUNC F1,RET=21Z
SFUNC F2,RET=22Z
SFUNC F3,RET=23Z
SFUNC F4,RET=24Z
SFUNC F5,RET=25Z
SFUNC nn,RET=nnZ
```

(PRKEY の詳細については、パラメータ [PRKEY](#) を参照)

その他のファンクションキーを使用したり、KDCDEF に定義されていない有効なファンクションキーを使用すると、エラーメッセージが発行されます。

## IBM 3270 タイプ端末のサポート

IBM SNA (VTAM) と Siemens TRANSDATA DC の適切なシステムコンフィグレーションでは、3270 タイプ端末は Natural UTM インターフェイスによってサポートされます。

つまり、Siemens 端末と 3270 端末は、Natural UTM アプリケーションに接続できます。Natural では、使用する特定の端末タイプに応じて画面出力が調整されます。3270 タイプの端末は、PTERM コマンドの KDCDEF に適切に定義する必要があります (Siemens UTM のドキュメントを参照)。

Siemens と IBM のファンクションキーをサポートするには、KDCDEF の SFUNC ステートメントを、次のように定義する必要があります。

Siemens キー	IBM キー	UTM リターンコード
F1	PF1	21Z
F2	PF2	22Z
F3	PF3	23Z
F4	PF4	24Z
F5	PF5	25Z
K1	PA1	26Z
K2	PA2	27Z
K3	PF6 + PF13	28Z
K4	PF7 + PF14	29Z
K5	PF8 + PF15	30Z
K6	PF9 + PF16	31Z
K7	PF10 + PF17	32Z
K8	PF11 + PF18	33Z
K9	PF12 + PF19	34Z
K10	PF20	35Z
K11	PF21	36Z
K12	PF22	37Z
K13	PF23	38Z
K14	PF24	39Z

## TTY 端末のサポート

TTY モードで使用する端末に対しては、PTERM コマンドの TERMN オペランドを TERMN=X9 に設定する必要があります。

次の制限が TTY モードに適用されます。

- 非同期トランザクション処理はサポートされていません。
- MULTI-PASS はサポートされていません。

## UTM DC トランザクション出口ルーチン NUERROR

UTM DC トランザクション出口ルーチンは、Natural UTM インターフェイスのフロントエンド部分で定義します。このルーチンが呼び出されるのは、DC トランザクションの開始時、DC トランザクションの再スタート時、正常終了時、および異常終了時 (PEND ER) です。ユーザー出口 `UVGEXIT` は、これらのいずれの状況でも使用することができます。

異常終了の場合、影響を受けたユーザーは内部の端末コントロールテーブルから削除され、Natural リカバリプロシージャが実行され、必要に応じてこのユーザーのユーザーエリアがスワッププールディレクトリから解放されます。

DC トランザクション出口ルーチン NUERROR は、Natural UTM インターフェイスのフロントエンド部分に対して、KDCDEF の適切なパラメータで定義する必要があります (KDCROOT の生成)。「[KDCROOT の生成](#)」も参照してください。

## UTM 起動機能

ユーザー出口 `STARTEX` (パラメータ `STRTALL` のデフォルト値) を使用する場合、Natural UTM インターフェイスのフロントエンド部分に対する KDCDEF パラメータに `EXIT PROGRAM=NUSTART,USAGE=START` を定義する必要があります。

この定義により、タスク初期化ルーチン (共通メモリプールの割り当て、Natural のロードなど) は、各 UTM タスクの起動直後にアクティブになります。発生したエラーはコンソールに出力され、すべてのユーザーに適切なメッセージが送信されます。`SYSLST=YES` の場合、エラーは `SYSLST` にも出力されます。

UTM 起動機能を使用しない場合、ユーザーのログオン時に UTM タスクがアクティブになるまで UTM タスクは初期化されません。これらの状況でエラーが発生した場合、エラーメッセージはエラーが発生した端末に送信されます。その他のすべてのユーザーについては、アプリケーションにログオンしようとするときに、適切なメッセージが表示されます。

## UTM シャットダウン機能

ユーザー出口 SHUTEX1 および／または SHUTEX2（パラメータ SHUTALL と SHUTLST のデフォルト値）を使用する場合、Natural UTM インターフェイスのフロントエンド部分に対する KDCDEF パラメータ（KDCROOT）に、次のような設定を定義する必要があります。

```
EXIT PROGRAM=NUSTART,USAGE=SHUT
```

最後の UTM タスクが終了すると、Natural UTM インターフェイスの統計情報が出力されます。

UTM シャットダウン機能を使用しない場合、SHUTALL と SHUTLST で定義したユーザー出口は使用することができず、統計情報も出力されません。

Natural UTM インターフェイスに収集され、出力される統計情報は次のとおりです。

```
MAIN DIRECTORY IS RESIDENT, NOT PAGEABLE
-----
INITIALIZED WITH CONTROL DATA FROM NAT SYSTEM FILE
TOTAL SIZE OF SWAP POOL IN KB : 32760
SIZE OF MAIN DIRECTORY IN KB : 2
TOTAL NO. OF SWAP POOL THREADS: 209
TOTAL NO. OF LOGICAL SWP(S) : 10

LAST STATUS OF THE SWAP POOL STATISTICS
+-----+
I      Natural USER THREADS WITH LENGTH      I
I      GREATER                                LOWER      I
I                                     152 KB      I
+-----+
I + 2 KB:          0 I - 2 KB:          0 I
I + 4 KB:          0 I - 4 KB:          17 I
I + 6 KB:          0 I - 6 KB:          0 I
I + 8 KB:          0 I - 8 KB:          1 I
I + 10 KB:         0 I - 10 KB:         0 I
I + 12 KB:         0 I - 12 KB:         0 I
I + 14 KB:         0 I - 14 KB:         0 I
I + 16 KB:         0 I - 16 KB:         0 I
I + 18 KB:         0 I - 18 KB:         0 I
I + NN KB:         0 I - NN KB:         0 I
I AVER.LNG.NN:     0 KB I AVER.LNG.NN:   0 KB I
+-----+
```



LOGICAL SWP NO. 01 -----	LOGICAL SWP NO. 02 -----
LOGICAL SWP SIZE IN KB: 2402	LOGICAL SWP SIZE IN KB: 2690
DIRECTORY SIZE IN KB : 2	DIRECTORY SIZE IN KB : 2
SWP THREAD SIZE IN KB : 120	SWP THREAD SIZE IN KB : 128
NO. OF SWP ENTRIES : 20	NO. OF SWP ENTRIES : 21
MAX. USED ENTRIES : 0	MAX. USED ENTRIES : 0
NO. OF SWP GUESTS : 0	NO. OF SWP GUESTS : 0
NO. SUCCESSFUL LOCATES: 0	NO. SUCCESSFUL LOCATES: 0
NO. FAILED LOCATES : 0	NO. FAILED LOCATES : 0
LOGICAL SWP NO. 03 -----	LOGICAL SWP NO. 04 -----
LOGICAL SWP SIZE IN KB: 2858	LOGICAL SWP SIZE IN KB: 3026
DIRECTORY SIZE IN KB : 2	DIRECTORY SIZE IN KB : 2
SWP THREAD SIZE IN KB : 136	SWP THREAD SIZE IN KB : 144
NO. OF SWP ENTRIES : 21	NO. OF SWP ENTRIES : 21
MAX. USED ENTRIES : 0	MAX. USED ENTRIES : 1
NO. OF SWP GUESTS : 0	NO. OF SWP GUESTS : 0
NO. SUCCESSFUL LOCATES: 0	NO. SUCCESSFUL LOCATES: 1
NO. FAILED LOCATES : 0	NO. FAILED LOCATES : 0
LOGICAL SWP NO. 05 -----	LOGICAL SWP NO. 06 -----
LOGICAL SWP SIZE IN KB: 3194	LOGICAL SWP SIZE IN KB: 3362
DIRECTORY SIZE IN KB : 2	DIRECTORY SIZE IN KB : 2
SWP THREAD SIZE IN KB : 152	SWP THREAD SIZE IN KB : 160
NO. OF SWP ENTRIES : 21	NO. OF SWP ENTRIES : 21
MAX. USED ENTRIES : 1	MAX. USED ENTRIES : 0
NO. OF SWP GUESTS : 0	NO. OF SWP GUESTS : 0
NO. SUCCESSFUL LOCATES: 17	NO. SUCCESSFUL LOCATES: 0
NO. FAILED LOCATES : 0	NO. FAILED LOCATES : 0
LOGICAL SWP NO. 07 -----	LOGICAL SWP NO. 08 -----
LOGICAL SWP SIZE IN KB: 3530	LOGICAL SWP SIZE IN KB: 3698
DIRECTORY SIZE IN KB : 2	DIRECTORY SIZE IN KB : 2
SWP THREAD SIZE IN KB : 168	SWP THREAD SIZE IN KB : 176
NO. OF SWP ENTRIES : 21	NO. OF SWP ENTRIES : 21
MAX. USED ENTRIES : 0	MAX. USED ENTRIES : 0
NO. OF SWP GUESTS : 0	NO. OF SWP GUESTS : 0
NO. SUCCESSFUL LOCATES: 0	NO. SUCCESSFUL LOCATES: 0
NO. FAILED LOCATES : 0	NO. FAILED LOCATES : 0
LOGICAL SWP NO. 09 -----	LOGICAL SWP NO. 10 -----
LOGICAL SWP SIZE IN KB: 3866	LOGICAL SWP SIZE IN KB: 4118
DIRECTORY SIZE IN KB : 2	DIRECTORY SIZE IN KB : 2
SWP THREAD SIZE IN KB : 184	SWP THREAD SIZE IN KB : 196
NO. OF SWP ENTRIES : 21	NO. OF SWP ENTRIES : 21
MAX. USED ENTRIES : 0	MAX. USED ENTRIES : 0
NO. OF SWP GUESTS : 0	NO. OF SWP GUESTS : 0

```
NO. SUCCESSFUL LOCATES:    0    NO. SUCCESSFUL LOCATES:    0
NO. FAILED LOCATES :      0    NO. FAILED LOCATES :      0
```

-----  
 USAGE STATISTICS OF SWAP POOL AND NATURAL USER THREADS

SWP NO.	SLOT NO. LNG KB	NAT. THREADS	%	DIAGRAM
01	120	0	0,0	
02	128	0	0,0	
03	136	0	0,0	
04	144	1	5,5	***
05	152	17	94,4	*****
06	160	0	0,0	
07	168	0	0,0	
08	176	0	0,0	
09	184	0	0,0	
10	196	0	0,0	
DESERTERS:		0	0,0	

M A I N D I R E C T O R Y S T A T I S T I C S A R E A

```
NATSHARE: SWAP POOL START DATE          02-07-17
NATSHARE: SWAP POOL START TIME         11:52:14
NATSHARE: TOTAL NUMBER OF SWP SYNC. WAITS:      0
NATSHARE: TOTAL NUMBER OF ASYN. WRITE WAITS:   0
NATSHARE: TOTAL NUMBER OF DIALOGUE STEPS:     19
NATSHARE: TOTAL NUMBER OF SWAPS:             0
NATSHARE: TOTAL NUMBER OF WRITES TO ROLL FILE: 0
NATSHARE: TOTAL NUMBER OF SYNCHRONOUS WRITES: 0
NATSHARE: MAX NUMBER OF USER:               1
NATSHARE: MAX NUMBER OF DIALOGUES WITHOUT SWAPS: 19
NATSHARE: NUMBER OF SWAP POOL REORGANIZATION: 0
NATSHARE: NUMBER OF SWAP POOL REPAIR:        0
NATSHARE: NUMBER OF ABNORMAL ENDED SESSIONS: 0
NATSHARE: MAX. COMPR. L' OF NAT USER THREAD IN KB: 148
NATSHARE: REAL MAXSIZE NEEDED FROM Natural IN KB: 386
```

When Swap Pool Data Space is generated:

```
NATSHARE: D A T A S P A C E ( D S F ) S T A T I S T I C S
NATSHARE: NO. OF ESA DATA SPACE SLOTS:      30
NATSHARE: NO. OF WRITES INTO DATA SPACE:    0
NATSHARE: NO. OF FAILED WRITES (DATA SPACE FULL): 0
NATSHARE: NO. OF SUCCESSFUL DSP THREAD LOCATES: 0
NATSHARE: NO. OF FAILED DSP THREAD LOCATES: 0

NATSHARE: SWP STATISTICS PRINT DATE:        02-07-17
```

NATSHARE: SWP STATISTICS PRINT TIME: 11:52:14



# 34 UTM 環境の Natural - パート 2

---

■ NATUTM マクロパラメータ .....	276
■ NATUTM マクロエントリ .....	306
■ NURENT マクロパラメータ .....	308

Natural UTM インターフェイスのドキュメントのこの部分では、マクロのパラメータについて説明します。

## NATUTM マクロパラメータ

次のパラメータを使用できます。

ADACALL | ADACOM | ADAPRI | ADAUTM | APPLNAM | APRISTD | ASAPPLI | ASYNTAC | BADTAC | CDYNAM | CLRKEY | CURPRO | ICONTRL | INITPRG | KB | KSAVE | KBUSEXT | LFH | LINK | LINK2/LINK3/LINK4 | LOFFMAP | NATMON | NUAADDR | NUCNAME | PARMOD | PENDPR | PFK | PRKEY | REFRKEY | ROLLACC | ROLLTSZ | RSTCNT | RSTWARM | SCRNOPT | SHUTALL | SHUTLST | SPOOL | STRTALL | STRTFST | SVDYPRM | SWAMODE | SWDPAGE | SWPUSID | SYAPPLI | SYNTAC | SYSLST | TACEND | TCLA1 | TCLA2, TCLA3, TCLA4 | TCLS1 | TCLS2, TCLS3, TCLS4 | TERMTAB | TID | TRACE | TTYLS | TTYP5 | UMODE

### ADACALL - Adabas へのアクセス

このパラメータでは、サブルーチン ADACALL に対する Natural UTM インターフェイスのエントリを定義します。このサブルーチンは、Natural 以外のプログラムが Adabas にアクセスするときに呼び出す必要があります。ADACALL によって、有効な Adabas ユーザー ID が生成され、それに続いて Adabas インターフェイスモジュール ADALNN が呼び出されます。可能な値は次のとおりです。

値	説明
<i>name</i>	エントリの <i>name</i> 。
NO	サブルーチン ADACALL は生成されません。 これはデフォルト値です。

### ADACOM - Adabas リンクモジュールの使用

このパラメータは、使用する Adabas リンクモジュールを決定します。可能な値は次のとおりです。

値	説明
ADABAS	モジュール ADAUSER および SSFB2C がフロントエンド部分にリンクされます (Adabas バージョン 7.1 以降)。
ADALINK	モジュール ADALNK、ADAL2P、および SSFB2C がフロントエンド部分にリンクされます (Adabas バージョン 7.1 以降)。
, (コンマ)	モジュール ADALNN がフロントエンド部分にリンクされます (Adabas バージョン 7.1 未満)。 これはデフォルト値です。

## ADAPRI - UTM アプリケーションの Adabas プライオリティ制御のアクティブ化

詳細については「[Adabas プライオリティ制御](#)」を参照してください。可能な値は次のとおりです。

値	説明
YES	Natural UTM アプリケーションの Adabas プライオリティ制御をアクティブにします。
NO	すべての UTM トランザクションに対して Adabas プライオリティは同じです。 これはデフォルト値です。

## ADAUTM - 非同期 UTM/Adabas トランザクションの同期

このパラメータでは、UTM と Adabas 間の非同期トランザクションを対象として、同期処理と再スタートの調整を実行することができます。この処理では、モジュール ADAUTM が使用可能であることが前提条件です。このモジュールは、Natural UTM アプリケーションのフロントエンド部分にリンクする必要があります。可能な値は次のとおりです。

値	説明
YES	UTM と Adabas 間の非同期トランザクションの同期処理、および調整された再スタートが有効になります。  <b>注意:</b> ADAUTM=YES が指定されている場合、ADACALL パラメータは ADABAS に設定しないでください。
NO	これはデフォルト値です。  <b>注意:</b> このパラメータのデフォルト値を変更する場合は、必ずその前に Software AG サポートに相談してください。

## APPLNAM - Natural UTM アプリケーションの名前

このパラメータでは、Natural UTM アプリケーションの名前を指定します。このパラメータの値は、KDCDEF のパラメータ APPLINAME の値と同じにする必要があります。この名前は、タスク固有の SYSLST ファイルの名前を作成するために使用されます（下記の SYSLST パラメータも参照）。可能な値は次のとおりです。

値	説明
<i>name</i>	最大 8 文字。  デフォルト値はありません。

指定した名前は、Natural UTM インターフェイスで、初期化ルーチンのシリアライゼーションマーカーの作成にも使用されます。s は、空いている最初の文字位置に挿入されます（例えば、APPLNAM=NATUTM の場合、シリアライゼーションマーカーの名前は NATUTMS になります）。

さらに TID=N が指定されている場合、この名前は Adabas ユーザー ID の作成に使用されます。

APPLNAM のオペランドの定義された文字位置は、Adabas ユーザー ID の作成に使用することができます。パラメータ TID を参照してください。

### APRISTD - 標準的な UTM TAC に対する Adabas プライオリティ

このパラメータは、標準的な UTM TAC（デフォルト NAT）に対して、Adabas プライオリティ *nnn* を定義するために使用することができます。可能な値は次のとおりです。

値	説明
<i>nnn</i>	標準的な UTM TAC（デフォルト NAT）に対する Adabas プライオリティ <i>nnn</i> 。
144	これはデフォルト値です。

APRISTD パラメータが有効になるのは、ADAPRI パラメータが YES に設定されている場合のみです。パラメータ TCLSn と TCLAn を使用すると、各 TAC に対して優先度を個別に定義できます。「[Adabas プライオリティ制御](#)」も参照してください。

### ASAPPLI - 論理 UTM 通信パートナーの名前

このパラメータでは、非同期 UTM アプリケーションの論理 UTM 通信パートナーの名前を（KDCDEF での定義に従って）指定します。この名前が使用されるのは、2つの UTM アプリケーションの間で非同期トランザクションの処理を行う場合に限られます。可能な値は次のとおりです。

値	説明
<i>name</i>	<i>name</i> は論理 UTM 通信パートナーの名前を指定します。 <b>注意:</b> ASAPPLI= <i>name</i> が指定されている場合、パラメータ SYAPPLI のオペランドも定義する必要があります。
NO	これはデフォルト値です。



## ASYNTAC - 非同期 UTM タスクまたはアプリケーションに対する UTM トランザクションコード

このパラメータでは、非同期的に実行する UTM タスクまたはアプリケーションに対して、UTM トランザクションコード (TAC) を定義します。可能な値は次のとおりです。

値	説明
<i>tac</i>	非同期的に実行する UTM タスクまたはアプリケーションに対する UTM TAC。
NATAS	これはデフォルト値です。

指定する UTM TAC は、「標準的な」 Natural TAC や、同期 UTM アプリケーションに使用する TAC とは区別する必要があります (2 つの UTM アプリケーションの間で、非同期トランザクション処理を行う場合)。

最初の 5 文字で、非同期 UTM TAC の一意の識別子が決定されます。

## BADTAC - UTM 機能 BADTACS のアクティブ化

このパラメータでは、UTM 機能の BADTACS をアクティブにすることができます。つまり、マクロ NATUTM のアセンブル済みのプログラムでは、未定義の UTM トランザクションコードに対して起動プログラム AUTOTAC が生成されます。可能な値は次のとおりです。

値	説明
YES	UTM 機能 BADTACS をアクティブにします。
NO	これはデフォルト値です。

 **注意:** BADTAC=YES を指定する場合は、KDCDEF を定義するとき、および KDCROOT を生成するときに、次の追加的な定義を指定する必要があります。

### KDCDEF :

```
PROGRAM AUTOTAC,COMP=ASSEMB
TAC KDCBADTC,CALL=FIRST,PROGRAM=AUTOTAC,EXIT=NUERROR,TYPE=D
TAC AUTOCONN,TYPE=D,PROGRAM=NATUTM,EXIT=NUERROR,CALL=BOTH
```

## CDYNAM - ダイナミックにロードするプログラムの最大数

このパラメータでは、ダイナミックにロードするプログラム（例えば、COBOLまたはアセンブラのサブルーチン）の最大数を指定したり、フロントエンド部分とリンクされ、パラメータ **LINK** で **LINK4** に宣言されているプログラムの数を指定したりします。可能な値は次のとおりです。

値	説明
<i>nn</i>	<i>nn</i> はプログラム数を定義します。
15	これはデフォルト値です。



**注意:** ダイナミックにロードするプログラムは、Natural パラメータモジュールに指定したロードライブラリ、または開始ジョブに指定した BLSLIB ライブラリ（複数可）のどちらかに保存する必要があります。

## CLRKEY - CLEAR キーのアクティブ化／非アクティブ化

このパラメータは、CLEAR キーのアクティブ化または非アクティブ化を実行します。可能な値は次のとおりです。

値	説明
ON	CLEAR キー（キー LSP と ENTER）をアクティブにします。 これはデフォルト値です。
OFF	CLEAR キーを非アクティブにします。つまり、CLEAR キーを押した後で、最後の Natural 画面全体が再び表示されます。

## CURPRO - 保護フィールドへのカーソルの移動

このパラメータで、保護フィールドへのカーソルの移動を許可するかどうかを制御します。可能な値は次のとおりです。

値	説明
ON	カーソルは保護フィールドには移動できません。 これはデフォルト値です。
OFF	カーソルを保護フィールドに移動できます（例えば、フィールド固有のヘルプ機能を表示するため）。

## ICONTRL - マイナスフォーマットのメッセージに対する UTM 入力出口

このパラメータは、マイナス (-) フォーマットのメッセージ（つまり、Natural 画面からのメッセージ）に対して UTM 入力出口を生成できます。このような入力出口により、許可された（または許可されない）ユーザー KDC コマンドを制御します。可能な値は次のとおりです。

値	説明
(YES, KDC xxxx (, KDC xxxx ,...))	許可されない KDC コマンドは、YES とその KDC コマンドの名前を指定して、このパラメータで定義する必要があります。以下の例を参照してください。
(YES)	
(NO)	これはデフォルト値です。

例：

ICONTRL=(NO)	この例では入力出口は生成されず、すべての UTM コマンドが許可されます。
ICONTRL=(YES)	この例では名前 ICONTRL で入力出口が生成され、すべての UTM コマンドの使用が禁止されます。
ICONTRL=(YES, KDCOUT, KDCOFF)	この例では名前 ICONTRL で入力出口が生成され、コマンド KDCOUT と KDCOFF の使用が禁止されます。

YES が最初のオペランドとして指定されている場合、生成する入力出口を、次のように KDCDEF と KDCROOT に定義する必要があります。

```
EXIT PROGRAM=ICONTRL,USAGE=(INPUT,USERFORM) PROGRAM ICONTRL,COMP=ASSEMB
```

## INITPRG - Natural 変数 \*INIT-PROGRAM の値

このパラメータは、Natural 変数 \*INIT-PROGRAM の値を定義します。可能な値は次のとおりです。

値	説明
APPLNAM	Natural 変数 *INIT-PROGRAM には、パラメータ APPLNAM の値が格納されます。 これはデフォルト値です。
KCTACVG	Natural 変数 *INIT-PROGRAM には、UTMKB フィールド KCTACVG の値が格納されます（UTM は TAC を開始します）。

## KB - KB アドレスを最初のパラメータとして渡す

このパラメータは、Natural が Natural 以外のプログラムを呼び出すたびに、UTM 通信エリア KB (*Kommunikationsbereich*) のアドレスを、最初のパラメータアドレスとして渡すかどうかを指定します。Natural UTM インターフェイスのサブルーチンとユーティリティプログラムでは、この値が考慮されてきました。可能な値は次のとおりです。

値	説明
YES	Natural が Natural 以外のプログラムを呼び出すたびに、UTM 通信エリア (KB) のアドレスが最初のパラメータアドレスとして渡されます。
NO	これはデフォルト値です。

## KBSAVE - SPUT による UTM KB の保存

このパラメータは、SPUT を使用して UTM KB を保存するかどうかを指定します。可能な値は次のとおりです。

値	説明
YES	UTM KB は SPUT により、KB ヘッダーの末尾に 12 バイトを追加した位置から保存されます。この情報は、ユーザー固有の部分的な UTM プログラムに対して PEND PR が実行される前に LSSB に保存されます。
NO	UTM KB は保存されません。 これはデフォルト値です。

このパラメータを使用できるようにするには、次の KDCDEF 定義を設定する必要があります。

```
MAX LSSBS=1
```

ユーザー固有の部分的な UTM プログラムが再開されると、元の通信エリアは SGET によってリフレッシュされます。こうして、部分的な UTM プログラムは、UTM 通信エリアヘッダーの末尾に 12 バイトを追加した位置から KB を使用することができます。したがって、プログラムでこの 12 バイトを壊すことは許されません。KB ユーザー拡張が定義されている場合、このエリアは保存されません。

## KBUSEXT - UTM KB ユーザー拡張の長さ

このパラメータは、UTM KB ユーザー拡張の長さを指定します。可能な値は次のとおりです。

値	説明
<i>nnnnn</i>	<i>nnnnn</i> は UTM KB ユーザー拡張の長さを指定します。 許容される最大長は 30,720 バイトです。
0	これはデフォルト値です。

ユーザー拡張の長さとはアドレスは KB に格納されます。

USEREXTL DS	H	<i>length in bytes</i>
USEREXTA DS	F	<i>address</i>

詳細については、DSECT マクロ CMBS2TP を参照してください。

## LFH - Adabas LFH の使用

このパラメータは、Adabas Large File Handler (LFH) を使用することを指定します。可能な値は次のとおりです。

値	説明
YES	Adabas LFH を使用することを指定します。
NO	これはデフォルト値です。

YES を指定する場合、Natural パラメータモジュール内で Adabas LFH のバッファサイズ (パラメータ VSIZE) も定義する必要があります。

## LINK - Natural から呼び出すプログラムとモジュール

このパラメータでは、Natural プログラムから呼び出されて、非リエントラント部分とリンクされるプログラムおよびモジュールの名前を指定できます。可能な値は次のとおりです。

値	説明
<i>name</i> ( <i>name</i> , <i>name</i> ,...)	Natural プログラムから呼び出され、非リエントラント部分とリンクされるプログラムおよびモジュールの <i>names</i> は、このパラメータのオペランドに指定する必要があります。  逆に、名前を指定するプログラムおよびモジュールは、非リエントラント部分とリンクする必要があります。そうしないと、アプリケーションのステータスが SYSTEMERROR になり、すべてのユーザーが拒否され、エラーメッセージが表示されます。

TABLE マクロコールは、指定されたプログラムとモジュールに対して実行され、ダイナミックローダーのリンクテーブルにロードアドレスが入力されます。したがって、Natural プログラムで呼び出すときは、これらのプログラムをダイナミックにロードする必要はありません。

例：

```
LINK=PROG1
LINK=(PROG1,PROG2,MODUL111)
```

### LINK2/LINK3/LINK4 - パラメータ LINK の拡張

これらのパラメータは、パラメータ LINK の拡張です。可能な値は次のとおりです。

値	説明
<i>name</i> ( <i>name</i> , <i>name</i> ,...)	構文は LINK の構文と似ています。以下の例を参照してください。

デフォルト値はありません。

オペランド定義の長さは 127 文字（カッコを含む）を超えてはならないため、パラメータ LINK のオペランドが長すぎる場合のために、これらのパラメータ LINK2~LINK4 が提供されています。

例：

```
NATUTM LINK=(PROG1,PROG2,...),
        LINK2=(PROG54,...)
NATUTM LINK=(PROG1,PROG2,PROG3,PROG4)
```

### LOFFMAP - ログオフメッセージのためのフォーマット名

このパラメータでは、ログオフメッセージのフォーマット名を指定できます。可能な値は次のとおりです。

値	説明
'-END'	<p>フォーマット出口モジュール FREXIT で定義するメッセージは、次のように出力されます。</p> <pre>NAT9994 - YOUR SESSION WAS SUCCESSFULLY FINISHED. PLEASE GIVE "KDCOFF" (LEAVE THE APPLICATION) OR "UTM-TAC".</pre> <p>メッセージはパラメータ UMODE で指定した言語で出力されます。必要がある場合、この設定はプログラム FREXIT で変更できます。</p> <p>これはデフォルト値です。</p>

値	説明
' '	次のメッセージは行モードで出力されます。  NAT9994 - Natural TERMINATED NORMALLY
'name'	ユーザー定義メッセージが出力されます。  メッセージは、FREET のマイナス (-) フォーマット、または IFG や FHS のアスタリスク (*) フォーマットで定義します。
'KDCOFF'	FIN システムコマンドまたは TERMINATE ステートメントの実行時に、ユーザーに対して自動 KDCOFF が実行されます。

いずれの場合でも、LOFFMAP パラメータに指定したオペランドは、UTM のフォーマット名として使用されます。したがって、オペランドは最大 8 文字に制限されます。

### NATMON - アプリケーション起動時での Natural Monitor の自動的なアクティブ化

このパラメータは、アプリケーションの起動時に Natural Monitor を自動的にアクティブにするかどうかを指定します。可能な値は次のとおりです。

値	説明
ON	Natural Monitor はアプリケーションの起動時に自動的にアクティブにされます。
OFF	Natural Monitor はアプリケーションの起動時に自動的にアクティブにされません。  これはデフォルト値です。

### NUAADDR - Natural ユーザーズレッドアドレス

このパラメータでは、Natural ユーザーズレッドアドレスを指定します。特定の値を指定すると、次のように動作します。

値	説明
, (コンマ)	コンマは、値がないことを表します。Natural ユーザーズレッドが、16MB 境界よりも下で次に空いているアドレスに割り当てられます。  これはデフォルト値です。
XXXXXX	Natural ユーザーズレッドが、16 MB 境界よりも下で、クラス 6 メモリ内の 16 進数アドレスに割り当てられます。このアドレスは、4 KB のセグメント制限に合わせる必要があります。アドレスの結果と Natural ユーザーズレッドの長さ (バイト) の合計 (MAXSIZE) がアドレス H'DF0000' よりも大きくなることは許可されません。値が最も高いアドレスは H'DEFFFF' です。
ABOVE	Natural ユーザーズレッドが、16 MB 境界よりも上に割り当てられます。
(ABOVE, NNNNN)	Natural ユーザーズレッドが、16 MB 境界よりも上に割り当てられます。この場合、NNNNN は 16 MB 境界よりも上の MB 数を 10 進数で表した値です。

例：

NUAADDR=ABOVE	Natural ユーザースレッドが、16 MB 境界よりも上で次に空いているアドレスに割り当てられます。
NUAADDR=(ABOVE,258)	Natural ユーザースレッドの 16 進数アドレスは H'10200000' です (16 MB 境界よりも上)。
NUAADDR=6E000I	Natural ユーザースレッドの 16 進数アドレスは H'6E000' です (16 MB 境界よりも下)。

Natural ユーザースレッドが 16 MB 境界よりも上に割り当てられる場合、非同期書き込みバッファと非同期トランザクション用スレッドも 16 MB 境界よりも上に割り当てられます。この場合、3GL プログラムが呼び出される前に、31 ビットアドレスモードが再び 24 ビットアドレスモードに切り替えられることはありません。つまり、呼び出される 3GL プログラムは、31 ビットアドレスモードで実行できる必要があります。

### NUCNAME - 連結される Natural のリエントラントモジュールの名前

このパラメータは、連結される Natural のリエントラントモジュールの名前を指定します。可能な値は次のとおりです。

値	説明
<i>name</i>	連結される Natural リエントラントモジュールの名前。

デフォルト値はありません。

共有ニュークリアスを使用する場合、マクロ ADDON (マクロ ADDON は BS2STUB をアセンブルします) の Natural プールとロード情報に対して、およびプログラム CMPSTART に対して、連結される Natural のリエントラントモジュールの名前を使用する必要があります。

### PARMOD - フロントエンド部分とリエントラント部分の生成

このパラメータは、フロントエンド部分とリエントラント部分の両方の生成に適用されます。可能な値は次のとおりです。

値	説明		
<i>nn, loc</i>	<i>nn</i>	24/31	このパラメータの最初の値 ( <i>nn</i> ) は、Natural UTM アプリケーションに対して、アドレッシングモード (24 ビットまたは 31 ビットモード) の定義に使用されます。
	<i>loc</i>	BELOW/ABOVE	このパラメータの第2の値 ( <i>loc</i> ) は、Natural UTM アプリケーションのフロント部分の場所を定義するために使用されます。
(31, ABOVE)	これはデフォルト値です。		



16 MB を超えるアプリケーションのフロント部分をロードする場合、フロント部分のリンクプロシージャで次のように定義する必要があります。

```
LOADPT=*XS
```

または

```
LOADPT=X'address'
```

例：

```
/EXEC TSOLINK
PROG NATvrs, FILENAM=NATvrs, LOADPT=*XS, ...
TRAITS RMODE=ANY, AMODE=31
INCLUDE...
/* PARMOD=(nn, loc) MUST ALSO BE DEFINED FOR ASSEMBLING MACRO NURENT, WHICH
/* BELONGS TO THE REENTRANT PART OF Natural UTM; OPERANDS MUST BE IDENTICAL FOR
/* THE FRONT-END AND REENTRANT PARTS.
```

## PENDPR - PEND PR に対する UTM TAC の定義

このパラメータは PEND PR に対して UTM TAC を定義します。可能な値は次のとおりです。

値	説明
'zzzzzzzz'	zzzzzzzz (最大 8 文字) は UTM TAC を定義します。
' '	これはデフォルト値です (PEND PR には TAC がありません)。

PENDPR='zzzzzzzz' が指定されている場合、FIN システムコマンドが入力されたとき、TERMINATE ステートメントが実行されたとき、または PEND PR ファンクションキーが押されたときに、PEND FI(NISH) の代わりに PEND PR(OGRAM) が実行されます。指定した UTM TAC と関連付けられた部分的な UTM プログラムは、PEND PR の後に起動されます。

## PFK - ファンクションキーモード

このパラメータは、次のファンクションキーモードのいずれかを設定するために使用されます。

値	説明
(KN, y)	リテラル %K1~%K20 および送信キーコード DÜ1 がファンクションキーにロードされます。
(K0, y)	リテラル 01~20 および送信キーコード F5 がファンクションキーにロードされます。
(KS, y)	リテラル A~T および送信キーコード F5 がファンクションキーにロードされます。さらにすべての出力メッセージで、ダミーフィールドが画面の最後の 2 つの位置で生成されます。このフィールドは、キー値の送受信に使用されます。
OFF	ファンクションキーモードは生成されません。

値	説明
KS,L	これはデフォルト値です。

この場合、*y* は次のいずれかの値です。

L	ファンクションキーがロードされます
N	ファンクションキーはロードされません

### PRKEY - ファンクションキーに対する UTM リターンコード

このパラメータは、ファンクションキー (F1~F5 または K1~K14) に対する UTM リターンコードの定義に使用します。可能な値は次のとおりです。

値	説明
<i>nnZ</i>	可能な値は 20Z~39Z です。
35Z	K10 のデフォルト値 (キー ESC+>)。

このパラメータで定義したファンクションキーが Natural ダイアログでアクティブにされた場合、Natural セッションは常に中断されます。別の部分的な UTM プログラムに対する UTM TAC がある場合、PEND PR(OGRAM) が実行されます。

この UTM TAC は、以下を使用して定義できます。

- Natural プロファイルパラメータ PROGRAM=*tac*、
- パラメータ PENDPR=*tac*、
- ユーティリティプログラム TACSWTCH。

PEND PR(OGRAM) によって呼び出された部分的な UTM プログラムから Natural UTM インターフェイスに制御が返されると、Natural セッションは中断されていたポイントから続行されます。

PRKEY パラメータで指定したものと同一リターンコードを、KDCDEF の SFUNC ステートメントでも定義する必要があります。

## REFRKEY - UTM ファンクションキーの定義

このパラメータは、UTMファンクションキーの定義に使用することができます。可能な値は次のとおりです。

REFRKEY= <i>nn</i> Z	<i>nn</i> の設定可能な値は、26～39 (K1～K14) の範囲に含まれます。
REFRKEY=NO	UTM ファンクションキーが定義されません。
REFRKEY=39Z	K14 のデフォルト値 (キー ESC +.)。

定義したファンクションキーを押すと、最後の完全な Natural 画面がリフレッシュされます。したがって、オペレータまたはオペレーティングシステムから受信したメッセージで画面が上書きされた後でも、Natural のダイアログでの操作を続行できます。送信キーコードは、Natural アプリケーションに渡されません。インターフェイスによって、Natural キーコードが ENTER に設定されます。

## ROLLACC - Natural ロールファイルに対するアクセス方法

このパラメータは、Natural ロールファイルに対するアクセス方法を定義します。可能な値は次のとおりです。

値	説明
UPAM-SY	Natural ロールファイルに対するアクセス方法は、同期ロールファイル I/O を使用する UPAM です。このアクセス方法は、グローバルスワッププールでは実行できません。
UPAM-AS	Natural ロールファイルに対するアクセス方法は、非同期書き込みに対して P1 イベント制御を使用する UPAM です。  これはデフォルト値です。
(UPAM-AS, PAMWAIT)	Natural UTM インターフェイスは、完了した非同期書き込みから VPASS SVC を受信したら、PEND RE が実行されるまで待機します。非アクティブ (P2 待ち) の UTM タスクは P1 イベント制御で転送できないので、このオプションは必須です。それよりも、エラーメッセージ「非同期書き込みのタイムアウト」でユーザーセッションを終了する必要があります。
FASTPAM	Natural ロールファイルに対するアクセス方法は、非同期書き込みに対して転送イベント制御を使用する FASTPAM です (高パフォーマンス)。下記的前提条件を参照してください。

**ROLLACC=FASTPAM の前提条件**

FASTPAM オプションを使用する場合は、次の前提条件が適用されます。

BS2000/OSD バージョン 1.0 以上

- パラメータ TERMTAB は SWP として定義する必要があります。
- 常駐 FASTPAM 環境と FASTPAM I/O プールを開始するバッチジョブ内のクラス II 定義は、次のように指定する必要があります。

```
/EXEC NATUTM,CLASSII=(nnn, yy)
```

- ユーザーカタログの FASTPAM 認可は、次のように指定する必要があります。

```
/SHOW-USER-ATTRIBUTES
      FIELD:DMS-TUNING-RESOURCES=*EXCLUSIVE
/*OR ALTERNATIVELY:
/MODIFY-USER
      FIELD: DMS-TUNING-RESOURCES=EXCLUSIVE-USE>
```

- BS2000/OSD オペレーティングシステムの BIAS は、次のように定義する必要があります。

```
/MODIFY-SYSTEM-BIAS MAX-RESIDENT-PAGES=nnn
```

必要な常駐コアページ数を計算するには、次の計算式を使用します（余りの値はすべて無視します）。

```
ROLLTSZ + 3 / 4 * 2 = N1 (FASTPAM I/O areas)
ROLLTSZ + 31 / 32 * 36 + 4095 / 4096 * 2 = N2 (FASTPAM access lists)
```

$N1 + N2 = 1$  つの Natural UTM タスクの常駐ページ数

**ROLLTSZ - 最大ロールスレッドサイズ**

このパラメータは、最大ロールスレッドサイズ *nnn*（単位はKB）を決定します。つまり、Natural ロールファイルでの圧縮ユーザースレッドの最大サイズです。可能な値は次のとおりです。

値	説明
<i>nnn</i>	<i>nnn</i> は 4 (ロールファイルブロックサイズ) の倍数にする必要があります。
160	これはデフォルト値です。

ROLLACC=UPAM-AS である場合、ROLLTSZ の有効値は 4~1600 (KB) です。

ROLLACC=UPAM-SY または ROLLACC=FASTPAM である場合、ROLLTSZ の有効値は 4~3200 (KB) です。

Natural ロールファイルのサイズを計算するには、次の計算式を使用します。

$ROLLTSZ / 2 * \text{ユーザーの最大数} = nnn$

*nnn* は Natural ロールファイルの PAM ページ数です。

ユーザースレッドは一般的に圧縮形式でロールファイルに書き込まれるので、最適なロールスレッドサイズはディスクストレージの大幅な節約に役立ちます。

*nnn* の最適値は、Natural スワッププールの統計情報で確認できます。SYSLST パラメータを参照してください。

### RSTCNT - 再スタート状況の制御

このパラメータは、再スタート状況の制御に使用することができます。再スタートの状況では、ユーザーの「存続期間」は前の Natural UTM セッションの結果で決まります。可能な値は次のとおりです。

値	説明
YES	このような再スタートの状況では、ユーザーに対してメッセージが表示され、UTM タスクが PEND FI(NISH) で終了されます。ユーザーは、UTM TAC を入力して UTM タスクを再スタートする必要があります。
NO	このような再スタートの状況では、メッセージが表示されずに、Natural セッションが新たに初期化されます。  これはデフォルト値です。

### RSTWARM - 再スタート状況の制御

このパラメータは、再スタート状況の制御に使用することができます。可能な値は次のとおりです。

値	説明
YES	UTM の再スタート状況が発生した場合、Natural セッションのウォームスタートが実行されます。最後の端末画面が表示されます。この機能の前提条件はグローバルな Natural スワッププールです。 これはデフォルト値です。
NO	UTM の再スタート状況が発生した場合、Natural セッションの再スタートが実行されます。

### SCRNOPT - Natural の画面最適化機能を非アクティブにする端末タイプ

このパラメータでは、Natural の画面最適化機能を非アクティブにする端末タイプ（1 つまたは 2 つ）を定義します。可能な値は次のとおりです。

値	説明
(yy=zz)	yy は、KDCDEF での定義に従って、有効な端末名 TERMN にする必要があります。 zz は yy のシノニムです。  TERMN=zz で KDCDEF に定義した端末タイプの場合、画面最適化機能は非アクティブになります。
(yy=zz ,yy=zz)	上記と同じですが、2 つの端末が定義されています。
NO	すべての端末タイプで、画面最適化機能がアクティブになります。 これはデフォルト値です。

例：

```
SCRNOPT=(FL=Z9)
```

上記の意味は次に示すとおりです。

FL	IBM 3270 タイプ端末に対する有効な TERMN 名
Z9	3270 タイプ端末のシノニム

この例では、KDCDEF で TERMN=Z9 として定義された 3270 タイプ端末で、画面最適化機能が非アクティブになります。

## SHUTALL - ユーザー出口の名前

このパラメータでは、ユーザー出口の名前を指定できます。可能な値は次のとおりです。

値	説明
<i>name</i>	ユーザー出口の名前を指定します。
SHUTEX1	これはデフォルト値です。

KDCDEF で UTM SHUTDOWN 関数が定義されている場合、このユーザー出口は、UTM タスクが KDCSHUT で終了された場合は常に、Natural UTM インターフェイスによって呼び出されます。

## SHUTLST - ユーザー出口の名前

このパラメータでは、ユーザー出口の名前を指定できます。可能な値は次のとおりです。

値	説明
<i>name</i>	ユーザー出口の名前を指定します。
SHUTEX2	これはデフォルト値です。

KDCROOT で UTM SHUTDOWN 関数が定義されている場合、このユーザー出口は、最後の UTM タスクが KDCSHUT で終了された場合に、UTM 環境の Natural によって呼び出されます。

## SPOOL - プリンタタスクの自動的な開始と終了

このパラメータでは、スプーリングシステムを指定できます。可能な値は次のとおりです。

値	説明
(NATSPPOOL, 'enter-parms', n)	NATSPPOOL (Natural Advanced Facilities) で使用する場合は、「 <a href="#">NATSPPOOL の使用</a> 」を参照してください。
REPRO-2000	リモートスプーリングシステムで使用する場合は、「 <a href="#">REPRO-2000 リモートスプーリングシステムの使用</a> 」を参照してください。
RMSPPOOL	独自のユーザー出口プログラムで使用する場合は、「 <a href="#">RMSPPOOL ユーザー出口の使用</a> 」を参照してください。

デフォルト値はありません。

以下では次のトピックについて説明します。

- [NATSPPOOL の使用](#)
- [NATSPPOOL の処理ロジック](#)
- [REPRO-2000 リモートスプーリングシステムの使用](#)

■ RMSPOOL ユーザー出口の使用

**NATSPPOOL の使用**

NATSPPOOL (Natural Advanced Facilities) を使用する場合、SPOOL パラメータを指定すれば、Natural UTM アプリケーションの起動時に NATSPPOOL に必要なプリンタタスクを ENTER の呼び出しによって自動的に開始し、アプリケーションのシャットダウン時に終了するように指定できます。この場合、パラメータのオペランドは次のように指定する必要があります。

```
SPOOL=(NATSPPOOL,'enter-parm',n)
```

上記の意味は次に示すとおりです。

'enter-parms'	ENTER の呼び出しのパラメータ (アポストロフィで囲まれた値)
n	開始するプリンタタスクの数 (1~30 の範囲)

例：

次の ENTER ジョブは、開始と終了が自動的に実行されます。ファイル名は AF.E.PRINT です。

```
/LOGON
/OPTION MSG=FHL
/SYSFILE FILE=SYSLST
/EXEC NAFPTTSK
/LOGOFF
```

パラメータ SPOOL のオペランド定義：

```
SPOOL=(NATSPPOOL,'AF.E.PRINT,TIME=999',2)
```

この例で、NATSPPOOL は Natural スプーリングシステムの名前です。AF.E.PRINT は開始と終了を行う ENTER ジョブのファイル名です。TIME=999 は ENTER の呼び出しの追加的なオプションパラメータです (BS2000/OSD ENTER マクロの説明を参照)。2 は、2つの NATSPPOOL プリンタタスクの開始と終了を行うことを表します。

第2のサブオペランドには、ENTER マクロコールに対する有効なオペランド (アポストロフィで囲まれた値) を指定できます。

マクロ NURENT のパラメータ SPOOL のオペランドは NATSPPOOL にする必要があります。



## NATSPPOOL の処理ロジック

パラメータ SPOOL のオペランド定義に従って、指定した数の NATSPPOOL プリンタタスクがアプリケーションの起動時に開始されます。次にプロセス間通信によって、少なくとも 1 つのプリンタタスクが実行されていることがチェックされます。この条件が満たされていない場合、アプリケーションはステータス SYSTEMERROR に設定され、エラーメッセージがコンソールに出力されます。ログオンしようとしたユーザーにはこのメッセージが表示され、ログオンは拒否されます。

```
NUI0036 - SYSTEMERROR ... PLEASE GIVE KDCOFF
```

このシステムエラーの詳細については、エラーメッセージ NUI0036 を参照してください。

## REPRO-2000 リモートスプーリングシステムの使用

リモートスプーリングシステムを使用する場合（例えば、TD-SPOOL または REPRO-2000）、マクロ NATUTM と NURENT で SPOOL=REPRO-2000 を設定します。Software AG は、この機能をサポートしません。

リモートスプーリングシステムに対してインターフェイスモジュールを実装する場合は、Natural オフラインレポートで使用するロジックを考慮する必要があります（マクロ NURENT、ラベル CMWHC を参照）。オフラインレポートがアクティブになると、Natural は一度に 1 レコードずつ出力を転送します。出力レコードを送受信するロジック、出力レコードのレイアウトなどは、マクロ NURENT のサブルーチン CMWHC にあります。

## RMSPPOOL ユーザー出口の使用

リモートスプーリングインターフェイスとして、RMSPPOOL という独自のユーザー出口プログラムを使用する場合、マクロ NATUTM と NURENT に SPOOL=RMSPPOOL を設定します。ユーザー出口 RMSPPOOL の詳細については、「[ユーザー出口](#)」を参照してください。

## STRTALL - 全 UTM タスクに対するユーザー出口の名前

このパラメータでは、ユーザー出口の名前を指定できます。このユーザー出口は、UTM 環境の Natural で、UTM タスクの開始時に常に呼び出されます。可能な値は次のとおりです。

値	説明
<i>name</i>	ユーザー出口の名前を指定します。
STARTEX	これはデフォルト値です。

### STRTFST - 最初の UTM タスクに対するユーザー出口の名前

このパラメータでは、ユーザー出口の名前を指定できます。UTM STARTUP 関数を KDCDEF に定義した場合、最初の UTM タスクが開始されると、このユーザー出口が UTM 環境の Natural によって呼び出されます。可能な値は次のとおりです。

値	説明
<i>name</i>	ユーザー出口の名前を指定します。
STAPPLX	これはデフォルト値です。

### SVDYPRM - ダイナミック Natural パラメータ用の保存エリア長

このパラメータは、UTM KB 内のダイナミック Natural パラメータ用保存エリアの長さ（バイト単位）を決定します。これらのパラメータは、Natural UTM セッションの再起動時に使用されます。可能な値は次のとおりです。

値	説明
<i>nnnn</i>	UTM KB 内のダイナミック Natural パラメータ用保存エリアの長さをバイトで指定します。可能な値は 0、または 8~2048（バイト）です。
0	これはデフォルト値です。

### SWAMODE - 31 から 24 ビットアドレスモードへの切り替え

このパラメータは、PEND PR の実行前に、31 ビットアドレスモードを 24 ビットモードに切り替えるかどうかを決定します。いずれの値を設定するかは、部分的な UTM プログラムを 31 ビットアドレスモードで実行できるか（NO）、実行できないか（YES）で判断します。可能な値は次のとおりです。

値	説明
YES	31 ビットアドレスモードは 24 ビットモードに切り替えられます。
NO	これはデフォルト値です。

## SWDPAGE - スワッププールメインディレクトリのページ処理

このパラメータは、スワッププールメインディレクトリのページ処理を実行可能にするか、実行不能にするかを決定します。可能な値は次のとおりです。

値	説明
NO	スワッププールメインディレクトリのページ処理を実行不能にする指定です。
YES	これはデフォルト値です。

スワッププールディレクトリでページ処理が実行不能である場合、パフォーマンスは大幅に改善できます。この場合、スワッププールディレクトリのページ処理を実行不能にすると宣言するには BS2000/OSD マクロ CSTAT が使用されます。SWDPAGE=NO の指定を可能にするには、スタートアップジョブで常駐コアページの最大値と最小値を定義する必要があります。

例：

```
/EXEC E.NATvrs,CLASSII=(4,2)
```

詳細については、BS2000/OSD マクロ CSTAT の説明、または BS2000/OSD コマンド EXECUTE、オペランド CLASSII の説明を参照してください。また、SDFを使用する場合は、BS2000/OSD コマンド START-PROGRAM、オペランド RESIDENT-PAGES=PARAMETERS... の説明を参照してください。

マクロ CSTAT への呼び出しが失敗した場合でも、アプリケーションは実行できます。

## SWPUSID - スワッププールユーザー ID

このパラメータは、スワッププールユーザー ID を決定します。可能な値は次のとおりです。

値	説明
KCLOGTER	これは UTM KB の論理端末名です。 これはデフォルト値です。
KCBENID	これは UTM KB のユーザー名です。
INTERNID	これは内部端末 ID です (シリアル番号)。

### SYAPPLI - 論理 UTM 通信パートナーの名前

このパラメータでは、同期 UTM アプリケーションの論理 UTM 通信パートナーの名前を（KDCDEF での定義に従って）指定できます。可能な値は次のとおりです。

値	説明
<i>name</i>	パラメータ ASAPPLI のオペランドも定義する必要があります。
NO	これはデフォルト値です。

このオペランドが有効とされるのは、2つの UTM アプリケーションの間で非同期トランザクションの処理を行う場合に限られます。

### SYNTAC - 非同期から同期 UTM アプリケーションにメッセージを送信するための UTM TAC

このパラメータでは、非同期から同期 UTM アプリケーションに端末用のフリーメッセージを送信するのに使用する UTM トランザクションコードを定義します。可能な値は次のとおりです。

値	説明
<i>tac</i>	UTM トランザクションコードを指定します。
NATSY	これはデフォルト値です（同期 TAC）。

このパラメータに指定する UTM TAC は、「標準的な」Natural TAC や、非同期 UTM アプリケーションに使用する TAC とは区別する必要があります。

### SYSLST - UTM タスクに対する SYSLST ファイル生成

このパラメータでは、各 UTM タスクに対して SYSLST ファイルを生成するかどうかを定義します。SYSLST ファイルには、統計データとエラー情報（UTM タスクが異常終了した場合）が保存されます。可能な値は次のとおりです。

値	説明
YES	各 UTM タスクに対して、SYSLST ファイルが生成されます。 これはデフォルト値です。
NO	SYSLST ファイルは生成されません。

SYSLST ファイルの名前は `LST.name.tsn` です。これは、次の要素から生成されます。

LST	接頭辞
name	パラメータ <code>APPLNAM</code> の値
tsn	UTM タスクの 4 桁のタスクシーケンス番号

## TACEND - PEND でのアクション

このパラメータでは、UTM オペレーションキー `PEND` と連動して実行するアクションを定義します。可能な値は次のとおりです。

値	説明
KP	各ダイアログステップは、 <code>PEND KP (KEEP)</code> で終了されます。UTM-KB が <code>KDCFILE</code> のページプールに書き込まれるのは、UTM キャッシュストレージに使用できる追加のスペースがない場合に限られます。UTM と Adabas の間では、同期処理を実行できないことに注意してください。
RE	各ダイアログステップは <code>PEND RE (RETURN)</code> で終了されます。つまり、UTM トランザクションの終了時です。 <code>KDCFILE</code> のページプール内の UTM-KB は、各ダイアログステップで保存されます。UTM と Adabas の間で同期処理を実行する場合、この処理モードが必要です。  これはデフォルト値です。

## TCLA1 - プライオリティレベル 1 の非同期トランザクションの UTM TAC

このパラメータは、UTM TACCLASS の概念に従って、非同期トランザクションの UTM TAC にプライオリティレベル 1 を割り当てます。サブルーチン `NATTAC` で Natural プログラムからアクセスできる TAC テーブルが作成され、プライオリティレベルはパラメータとして渡されます。「[UTM TACCLASS の概念 \(プライオリティ制御\)](#)」を参照してください。

可能な値は次のとおりです。

値	説明
<code>tac (tac , nn)</code>	<code>nn</code> は対応する UTM TAC (TACCLASS) に対して Adabas プライオリティを制御するために指定できます。「 <a href="#">Adabas プライオリティ制御</a> 」を参照してください。
<code>-</code> <code>(-, 0)</code>	<code>TCLA1=-</code> (ダッシュはアポストロフィで囲まない) の指定は、UTMTAC が割り当てられないことを表します。
<code>(NATAS1,64)</code>	これはデフォルト値です。

### TCLA2、TCLA3、TCLA4-プライオリティレベル2、3、4の非同期トランザクションのUTM TAC

これらのパラメータは、UTM TACCLASS の概念に従って、非同期トランザクションの UTM TAC にプライオリティレベル 2、3、および 4 を割り当てます。これらの値は、TCLA1（上記を参照）と同様に使用されます。可能な値は次のとおりです。

値	説明
TCLA $n = tac$ TCLA $n = (tac, nn)$ TCLA $n = -$	TCLA1 と似ていますが、プライオリティレベル $n$ は 2、3、4 のいずれかです。
TCLA2=(NATAS2,48) TCLA3=(NATAS3,32) TCLA4=(NATAS4,16)	これらはデフォルト値です。

### TCLS1-プライオリティレベル1の非同期トランザクションのUTM TAC

このパラメータは、UTM TACCLASS の概念に従って、同期トランザクションの UTM TAC にプライオリティレベル 1 を割り当てます。サブルーチン NATTAC で Natural プログラムからアクセスできる TAC テーブルが作成され、プライオリティレベルはパラメータとして渡されます。

「[UTM TACCLASS の概念 \(プライオリティ制御\)](#)」を参照してください。

可能な値は次のとおりです。

値	説明
$tac (tac, nn)$ - (-, 0)	$nn$ は対応する UTM TAC (TACCLASS) に対して Adabas プライオリティを制御するために指定できます。「 <a href="#">Adabas プライオリティ制御</a> 」を参照してください。
(NAT1,128)	TCLS1=- (ダッシュはアポストロフィで囲まない) の指定は、UTM TAC が割り当てられないことを表します。
	これはデフォルト値です。

### TCLS2、TCLS3、TCLS4-プライオリティレベル2、3、4の非同期トランザクションのUTM TAC

これらのパラメータは、UTM TACCLASS の概念に従って、非同期トランザクションの UTM TAC にプライオリティレベル 2、3、および 4 を割り当てます。これらの値は、TCLS1（上記を参照）と同様に使用されます。

可能な値は次のとおりです。

値	説明
TCLS $n = tac$ TCLS $n = (tac, nn)$ TCLS $n = -$	TCLS1 と似ていますが、プライオリティレベル $n$ は 2、3、4 のいずれかです。
TCLS2=(NAT2,112) TCLS3=(NAT3,96) TCLS4=(NAT4,80)	これらはデフォルト値です。

## TERMTAB - Natural ロールファイル管理の端末コントロールテーブル

このパラメータでは、Natural ロールファイルの管理に必要な端末コントロールテーブルを定義します。可能な値は次のとおりです。

値	説明
(SWP, TERMNAME, CHECKPNT) (SWP, INTERNID, CHECKPNT) (SWP, TERMNAME) (SWP, INTERNID) (N, TERMNAME) (N, INTERNID)	20 バイト 長 12 バイト 10 バイト 2 バイト 10 バイト 2 バイト 以下の「 <a href="#">オペランドの説明</a> 」を参照してください。
(SWP, TERMNAME)	これはデフォルト値です。

端末コントロールテーブルは、Natural スワッププールまたは Natural ロールファイルで割り当てられます。このテーブルには、ヘッダー（48 バイト）と各アクティブユーザーまたはアクティブセッションのエントリが格納されます。テーブルのサイズは、Natural ロールファイルのサイズ、パラメータ `ROLLTSZ` の値、およびテーブル自体のエントリの長さによって決まります。

Natural UTM インターフェイスは、端末コントロールテーブルの長さを次のように計算します。

$$\text{Roll file pages} / (\text{ROLLTSZ} / 2) = N$$

$$N * \text{terminal control table entry length} + 48 = \text{length of the terminal control table}$$

## オペランドの説明

オペランド	意味
SWP	端末コントロールテーブルは、Natural スワッププールで割り当てられます。
TERMNAME	端末コントロールテーブルのエントリを識別するために、論理端末名が使用されます。
INTERNID	端末コントロールテーブルのエントリを識別するために、内部端末 ID（シリアル番号）が使用されます。INTERNID の長さは 2 バイトです。
CHECKPNT	これを使用できるのは、端末コントロールテーブルが Natural スワッププールで割り当てられる場合に限られます。端末を <code>RESTART=NO</code> で定義する場合、または端末プールを <code>KDCDEF</code> で定義する場合、この指定が必要になります。端末コントロールテーブルのエントリには、

オペラント	意味
	<p>ロールアウトした最後の Natural ユーザーズレッドのチェックポイント（タイムスタンプ）が含まれます。Natural ロールファイルのユーザーズレッドは、端末コントロールテーブルエントリのタイムスタンプよりも小さいタイムスタンプを持つスレッドで上書きしないでください。</p> <p>CHECKPNT の長さは 10 バイトです。</p>
N	<p>Natural ロールファイル内の端末コントロールテーブルに対する PAM ページ数。このオペラントに設定できる値は 1~16（PAM ページ）です。各端末には、10 バイトが端末コントロールテーブルに必要です。各セッションには 2 バイトが必要です。</p>

例：

```
TERMTAB=(2,TERMNAME)
```

端末コントロールテーブルのエントリの最適数： $2 * 2048 - 48 / 10 = 404$

```
TERMTAB=(1,INTERNID)
```

端末コントロールテーブルのエントリの最大数： $1 * 2048 - 48 / 2 = 1000$

### TID - Adabas ユーザー ID の作成方法

このパラメータは、「一意の」 Adabas ユーザー ID を作成する方法を指定します。可能な値は次のとおりです。

値	説明
$n$	<p>Adabas ユーザー ID は、パラメータ <b>APPLNAM</b>（デフォルト値：N）のオペラントの定義済み（<math>n</math>）文字に、ユーザーの最初の SWAPPAMKEY に含まれる最後の 2 文字を組み合わせて作成されます。<math>n</math> は 1~8 の範囲の数値にする必要があります。</p> <p><b>注意:</b> TID=<math>n</math> を指定する場合、パラメータ APPLNAM に指定された値の「定義済み文字」は、他の Natural UTM アプリケーションと同じ Adabas を使用する場合はそれらのアプリケーションの定義済み文字と区別する必要があります。そうしないと、Adabas ユーザー ID が一意でなくなるため、データの整合性が保証されなくなります。</p>
(T, $n$ )	<p>一意な 4 バイトのユーザー ID が、論理 UTM 端末名 (KCLOGTER) の文字 <math>n</math>~（最大 <math>n + 3</math>）を取得して作成されます。<math>n</math> は 1~8 の範囲の数値にする必要があります。結果の文字列は、有効な文字（0~9 および A~F）で構成され、一意である必要があります。下記の例を参照してください。</p>
(U, $n$ )	<p>文字は、UTM ユーザー ID (KCBENID) から取得され、第 2 のサブパラメータで指定した位置から開始します。結果の文字列は、有効な文字で構成され、一意である必要があります。</p>
(TID=1)	<p>Adabas ユーザー ID は、パラメータ <b>APPLNAM</b> のオペラントから取得した先頭の数字と、端末コントロールテーブル内の 2 バイトのエントリ番号で作成されます。</p> <p>これはデフォルト値です。</p>



例：

```
TID=(T,4)
```

	KCLOGTER	Adabas ユーザー ID
第 1 端末	LTU9A110	X'00009A11'
第 2 端末	LTU9F110	X'00009F11'
第 3 端末	LTU9F120	X'00009F12'

「組み合わせの」 Adabas の呼び出し（つまり、Natural プログラムと Natural 以外のプログラムからの両方の呼び出し）が 1 つの Natural UTM アプリケーションで実行される場合、マクロ NATUTM 内で ENTRY CMTRMID を使用して、Adabas ユーザー ID を検出することができます。現在の Adabas ユーザー ID（4 バイト）は、アドレス CMTRMID で検出できます。パラメータ ADACALL も参照してください。

例：


```
EXTRN CMTRMID
.
.
L R1,CMTRMID
MVC ADAID(4),0(R1)
```

### Adabas バージョン 5.2 の場合：

Adabas バージョン 5.2 では、Adabas ユーザー ID は、論理 UTM 端末名（KCLOGTER）または UTM ユーザー ID（KCBENID）から作成されます。いずれになるかは、パラメータ TID の設定によって決まります。互換性保持のため、TID で指定するオペランドの 2 桁目は 1~8 の数値にする必要があります。例えば、次のようになります。

```
KCLOGTER=LTU9A110   KCBENID=HARRIS

TID=(T,3) -> Adabas user ID: D9A110
TID=(U,1) -> Adabas user ID: HARRIS__
```

 **注意：**「組み合わせの」 Adabas の呼び出しの場合、守らなければならない特別な注意事項はありません。

非同期トランザクション処理の場合、Adabas ユーザー ID は非同期タスクの「パック」 TSN から作成されます。

**Adabas バージョン 5.3 以上の場合：**

Adabas バージョン 5.3 以上では、TID=(T, n) の場合は完全な端末名 (KCLOGTER) が Adabas ユーザー ID に使用され、TID=(U, n) の場合は完全なユーザー ID (KCBENID) が使用されます。デフォルトは TID=((T,1) です。

**TRACE - トレースファイル番号とトレース出力レコード長**

このパラメータでは、トレースファイルの番号とトレース出力レコードの最大長を指定します。可能な値は次のとおりです。

値	説明
(nn, ll)	nn は SYSLST nn トレースファイルの番号です。有効な範囲：01~99。 ll はトレース出力レコードの最大長を表す文字数です。有効な範囲：71~132。
(99,71)	これはデフォルト値です。

外部 Natural トレース機能がアクティブな場合は、トレースレコードは SYSLST nn に書き込まれます。この場合、Natural UTM ドライバによって次のトレースファイルが作成されます。

例：

```
applname.Natural.TRACE,SPACE=(90,60)
SYSFILE SYSLSTnn=applname.Natural.TRACE
/* applname is the application name
```

このファイルは、Natural UTM アプリケーションの全タスクに使用されます。Natural UTM アプリケーションが終了する前に、トレースファイルが次のように閉じられます。

```
SYSFILE SYSLSTnn=(PRIMARY)
```

Natural トレース機能をアクティブにするには、Natural パラメータモジュールのパラメータ ETRACE と ITRACE を参照してください。

## TTYLS - TTY デバイスの物理的な行サイズ

このパラメータでは、Natural の物理的な行の長さを、telex マシンのさまざまな用紙フォーマットに合わせることができます。可能な値は次のとおりです。

値	説明
<i>nn</i>	<i>nn</i> は、TTY デバイスの物理的な行サイズを指定します。
80	これはデフォルト値です。

## TTYPS - TTY デバイスの物理ページのサイズ

このパラメータでは、Natural の物理ページのサイズを、telex マシンのさまざまな用紙フォーマットに合わせることができます。可能な値は次のとおりです。

値	説明
<i>nn</i>	<i>nn</i> は、TTY デバイスの物理ページサイズ（行数）を指定します。
24	これはデフォルト値です。

## UMODE - セッション処理モード

このパラメータは、Natural セッションの処理モードを決定します。可能な値は次のとおりです。

値	説明
(S, <i>y</i> )	<i>S</i> はオペレーションモードを指定します。 <i>y</i> は言語インジケータです。 以下の「オペランドの説明」を参照してください。
(S,E)	これはデフォルト値です。

### オペランドの説明

オペランド	説明	
S	「オペレーションモードはシングル」という意味です。つまり、1つの端末/ユーザーからは、1回の Natural セッションのみを開始できます。	
<i>y</i>	再スタートメッセージ、「フリーランニングメッセージ」やログオフメッセージ用のメッセージに対する言語インジケータ	
	D	デンマーク語
	E	英語
	F	フランス語
	G	ドイツ語

オペランド	説明	
	I	イタリア語
	N	オランダ語
	S	スペイン語

## NATUTM マクロエントリ

- [CMKBADR - UTM KB の現在のアドレス](#)
- [スワッププールディレクトリのユーザーエリア](#)

### CMKBADR - UTM KB の現在のアドレス

エントリ CMKBADR には、UTM 通信エリア KB (*Kommunikationsbereich*) の現在のアドレスが格納されます。

この通信エリアには、次の例に示す方法でアクセスできます。この例では、Natural プログラムから呼び出すことができるアセンブラプログラムを説明します。

例：

```

EXAMPLE CSECT
  STM 14,12,12(13)
  USING EXAMPLE,15
  L 2,VCONST          LOAD ADDRESS OF KB-ADDRESS
  L 3,0(,2)           LOAD ADDRESS OF KB
  .
  .
  LM 14,12,12(13)
  BR 14
VCONST DC V(CMKBADR)  ENTRY ADDRESS
  END
    
```

この場合、プログラム名 EXAMPLE をマクロ NATUTM のパラメータ [LINK](#) または [LINK2](#) で定義する必要があります。プログラム自体は、Natural UTM インターフェイスのフロントエンド部分とリンクする必要があります。

## スワッププールディレクトリのユーザーエリア

Natural スワッププールディレクトリでは、ユーザー定義の目的で1つのフルワードを使用することができます。マクロ NAMSWDIR に含まれる DSECT MEMPOOL のラベル USERWRD を参照してください。このワードは同期に使用することができます。例えば、Natural UTM アプリケーションの実行中に、アカウントのオンとオフを切り替えるのに使用できます。

次の例では、このエリアのアドレス制御の方法について説明します。

例：

```

PROG      WXTRN CMKBADR      ENTRY IN MACRO NATUTM
          CSECT
          STM 14,12,12(13)  SAVE REGISTERS
          USING PROG,15    BASE OF PROGRAM
          USING KB,4       BASE OF UTM KB
          USING MAINDIR,5  BASE OF SWAP POOL DIRECTORY
          L 3,KBADR        LOAD ADDRESS OF KB ADDRESS
          L 4,0(,3)        LOAD ADDRESS UTM KB
          L 5,ASWPDIR      ADDRESS SWAP POOL DIRECTORY
          OI USERWRD+3,1  SET THE LOW ORDER BIT OF FIELD
*
          LM 14,12,12(13)  RELOAD REGISTERS
          BR 14            RETURN
KBADR     DC A(CMKBADR)    ENTRY IN MACRO NATUTM
          NAMSWDIR        MACRO CALL FOR SWAP POOL DSECT
MAINDIR   DSECT
          .
          .
USERWRD   DS F            DIRECTORY USER AREA
          .
          CMKBEX          MACRO CALL FOR UTM KB DSECT
KB        DSECT
          .
          .
ADRSWAP   DS F            ADDRESS OF Natural SWAP POOL
          .
          END

```

このエリアで処理を実行する場合、スワッププールディレクトリにある他のデータを上書きしないように注意する必要があります。操作を誤ると、UTMタスクが異常終了する可能性があります。

## NURENT マクロパラメータ

次のパラメータを使用できます。

ACCNT | ATTKEY | AUTOLINK | CALLM31 | CLR3270 | EXTAPPL | FPUT | ILCS | K2 | PARMOD | SCRNTRC  
| SPOOL | UINPEX | UOUTEX

### ACCNT - ユーザーアカウントルーチンに対する呼び出しロジック

このパラメータでは、ユーザーアカウントルーチン（ユーザー出口 ACCEXIT）の呼び出しに対してロジックを定義します。可能な値は次のとおりです。

値	説明
APPL	ACCEXIT はアプリケーションの変更時に呼び出されます（新しい Natural ログオン ID）。これはデフォルト値です。
DIAL	ACCEXIT は、すべてのダイアログステップの後で呼び出されます。

### ATTKEY - アテンション割り込みキー

このパラメータは、アテンション割り込みキーの定義に使用します。このようなキー定義が有効になるのは、非会話型モードで出力された場合に限られます。可能な値は次のとおりです。

値	説明
nnZ	nnZ には、26Z~39Z の範囲で値を指定できます。
ATTKEY=	デフォルト値：値なし

### AUTOLINK - AUTOLINK 機能の使用

このパラメータは、3GL プログラムをロードするダイナミックバイнда/ローダーの AUTOLINK 機能をアクティブにするかどうかを指定します。可能な値は次のとおりです。

値	説明
YES	AUTOLINK 機能がアクティブになります。これはデフォルト値です。
NO	AUTOLINK 機能が非アクティブになります。

## CALLM31 - 31 から 24 ビットアドレスモードへの切り替え

このパラメータが役立つのは、Natural が 31 ビットアドレッシングモードに生成され、フロント部分が下にロードされた場合のみです (PARMOD=31、[下記参照](#))。

値	説明
YES	Natural プログラムから 3GL プログラムに対する呼び出しは、31 ビットアドレッシングモードで実行されます。
NO	呼び出しは 24 ビットモードで実行されます。アドレッシングモードが 31 ビットから 24 ビットに切り替えられてから、3GL プログラムが Natural プログラムから呼び出されます。  これはデフォルト値です。

例外：

- 3GL プログラムは、16 MB 境界よりも上にロードされます。
- パラメータリストのアドレスは、16 MB 境界よりも上になります。

## CLR3270= xxx - CLEAR キーの定義

このパラメータは、3270 タイプデバイス (IBM) の AID 文字テーブル V (AID3270) で CLEAR キーを定義します。

値	説明
xxx	xxx は CLEAR キーを定義します。
PA1	デフォルトでは、PA1 が CLEAR キーです。

## EXTAPPL - 外部 DCAM または PDN アプリケーションの UTM TERMN 名

このパラメータは、外部 DCAM または PDN アプリケーションの UTM TERMN 名 (UTM KDCDEF のパラメータ PTERM を参照) を定義します。これらの TERMN 名に対しては、MGET リターンコード 05Z (フォーマットが変更された) は無視されます。可能な値は次のとおりです。

値	説明
xx	xx と yy は、UTM TERMN 名を定義します。
(xx, yy)	

デフォルト値はありません。

## FPUT - KDCS 呼び出しの FPUT での出力操作に関する補足情報

このパラメータは、UTM KDCDEF で定義されたプリンタへの KDCS 呼び出しの FPUT による出力に関して、操作の補足情報を定義します。

値	説明
NE	出力メッセージの全体。
NT	出力メッセージの一部。

## ILCS - CRTE または ILCS のサポート

このパラメータは、3GL プログラム呼び出しのための共通ランタイム環境 (CRTE) または ILCS インターフェイスをサポートするかどうかを指定します。可能な値は次のとおりです。

値	説明
CRTE	<p>3GL プログラム呼び出しのための共通ランタイム環境がサポートされます。</p> <p>前提条件：プログラム ITOSL# をフロント部分に含める必要があります。</p> <pre>INCLUDE ITOSL#,SYSLNK.CRTE.010 RESOLVE,SYSLNK.CRTE.010</pre>
YES	<p>3GL プログラム呼び出しのための ILCS インターフェイスのみがサポートされます。</p> <p>前提条件：プログラム ITOINITS をフロント部分に含める必要があります。</p> <pre>INCLUDE ITOINITS,SYSLNK.ILCS RESOLVE,SYSLNK.ILCS</pre>
NO	<p>CRTE または ILCS はサポートされません。</p> <p>これはデフォルト値です。</p>



## K2 - K2 キーに対する UTM リターンコード

このパラメータは、K2 キーに対して UTM リターンコードを指定します (Natural PA2 向け)。

値	説明
<i>nnn</i>	<i>nnn</i> には、26Z~39Z の範囲で値を指定できます。
27Z	これはデフォルト値です。

## PARMOD - フロントエンド部分とリエントラント部分の生成

このパラメータは、フロントエンド部分とリエントラント部分の両方の生成に適用されます。可能な値は次のとおりです。


値	説明		
( <i>nn</i> , <i>loc</i> )	<i>nn</i>	24/31	このパラメータの最初の部分 ( <i>nn</i> ) は、Natural UTM アプリケーションに対して、アドレッシングモード (24ビットまたは31ビットモード) の定義に使用されます。
	<i>loc</i>	BELOW/ABOVE	このパラメータの第2の部分 ( <i>loc</i> ) は、Natural UTM アプリケーションのフロント部分の場所を定義するために使用されます。
(31, ABOVE)	これはデフォルト値です。		

マクロ NATUTM をアセンブルするために、PARMOD=(*nn*, *loc*) も定義する必要があります。オペランドは、フロントエンド部分とリエントラント部分に対して同一にする必要があります。

## SCRNTRC - 画面 I/O のトレース

このパラメータは、特定のエラー状況の理由を明らかにするために画面 I/O のデバッグに使用します。このパラメータが ON/(ON, *nn*) に設定されている場合、各ユーザーに対して特別なデバッグバッファが割り当てられます (デフォルトのバッファサイズは 3 KB)。可能な値は次のとおりです。

値	説明
ON	各ユーザーにデバッグバッファが、デフォルトのバッファサイズ 3 KB で割り当てられます。
(ON, <i>nn</i> )	各ユーザーにデバッグバッファが割り当てられますが、 <i>nn</i> はデフォルト値の 3 KB ではなく、特定の画面デバッグバッファサイズを定義するために使用します。
OFF	これはデフォルト値です。

 **注意:** このパラメータを ON/(ON, *nn*) に設定する場合は、事前に必ず Software AG 技術サポートにお問い合わせください。

## SPOOL - ハードコピーの出力先

このパラメータでは、スプーリングシステムを指定できます。このパラメータ値は、マクロ NATUTM の SPOOL パラメータ値に対応している必要があります。可能な値は次のとおりです。

値	説明
NATSPPOOL	ハードコピーは NAF (Natural Advanced Facilities) によって出力されます。「 <a href="#">NATSPPOOL の使用</a> 」を参照してください。
REPRO-2000	リモートスプーリングシステムで使用する場合は、「 <a href="#">REPRO-2000 リモートスプーリングシステムの使用</a> 」を参照してください。
RMSPPOOL	独自のユーザー出口プログラムで使用する場合は、「 <a href="#">RMSPPOOL ユーザー出口の使用</a> 」を参照してください。
No value	ハードコピーは UTM (FPUT) によって出力されます。これはデフォルトです。

## UINPEX - ユーザー出口の名前

このパラメータでは、ユーザー出口の名前を指定できます。このユーザー出口は、端末メッセージの送信後に UTM 環境の Natural によって呼び出されます。「[ユーザー出口](#)」も参照してください。可能な値は次のとおりです。

値	説明
name	name はユーザー出口の名前を指定します。
INPSCR	デフォルトでは、ユーザー出口 INPSCR が使用されます。

## UOUTEX - ユーザー出口の名前

このパラメータでは、ユーザー出口の名前を指定できます。このユーザー出口は、端末メッセージの送信前に UTM 環境の Natural によって呼び出されます。「[ユーザー出口](#)」も参照してください。

値	説明
name	name はユーザー出口の名前を指定します。
OUTSCR	ユーザー出口 OUTSCR が使用されます。 これはデフォルト値です。

# 35 UTM 環境の Natural - パート 3

---

■ ユーザー出口 .....	314
■ UTM 環境での非同期トランザクション処理 .....	318
■ UTM 環境での印刷 .....	326
■ Natural 以外のプログラムの呼び出し .....	328
■ チェーンで結ばれた部分的な UTM プログラムの呼び出し .....	329
■ Natural UTM アプリケーションでの Natural 以外のプログラムからの Adabas の呼び出し .....	330
■ UTM タスクの異常終了 .....	330

表記の *vrs* または *vr* : 以下のドキュメントでは、表記の *vrs* または *vr* は関連するバージョン、リリース、システムメンテナンスレベル番号を表します。

## ユーザー出口

---

Natural UTM インターフェイスでは、いくつかのユーザー出口が提供されています。これらについて、以下で説明します。

これらの出口を使用するには、対応するユーザープログラムを、Natural UTM アプリケーションのフロントエンド部分にリンクする必要があります。ユーザー出口 RP2PRNT は例外です。

ユーザー出口ルーチンは、通常のレジスタ規則で呼び出されます。

ACCEXIT | ACCINIT | INPTEX | RP2PRNT | RMSPOOL | SHUTALL | SHUTLST | STRTALL | STRTFST | TRMIOEX | UINPEX | UOUTEX | UVGEXIT | WHCEXT

### ACCEXIT - マクロ NATUTM

ユーザー出口 ACCEXIT は、アカウント情報取得に使用できます。マクロ NURENT のパラメータ ACCNT の値に応じて、各ダイアログステップの終了時またはアプリケーションの各変更時（新しい Natural ログオン ID）に、このユーザー出口がアクティブにされます。「[Natural UTM アプリケーションのアカウント情報](#)」も参照してください。

### ACCINIT - マクロ NATUTM

ユーザー出口 ACCINIT は、アカウント情報の収集に使用できます。この出口は、各ダイアログステップの開始時にアクティブになります。「[Natural UTM アプリケーションのアカウント情報](#)」も参照してください。

### INPTEX - プログラム FREXIT


ユーザー出口 INPTEX は、入力メッセージが読み取られると常にアクティブになります。「[UTM 環境の Natural で使用するユーティリティプログラム](#)」にあるプログラム INPTEX の説明も参照してください。

## RP2PRNT - マクロ NURENT

ユーザー出口 RP2PRNT は、他の製造元のスプーリングシステムに対するインターフェイスとして使用されます。このユーザー出口ルーチン（スプーリングプログラム）はリエントラントにする必要があります、また Natural UTM アプリケーションのリエントラント部分とリンクしなければなりません。「[その他のスプーリングシステム](#)」、および「[NATUTM マクロパラメータ](#)」にあるパラメータ SPOOL の説明も参照してください。

## RMSPOOL - マクロ NATUTM および NURENT

独自のスプーリングインターフェイスプログラムを記述する場合は、RMSPOOL という名前を付けてください。プログラム RMSPOOL は、Natural UTM アプリケーションの（非リエントラントの）フロントエンド部分またはリエントラント部分にリンクすることができます。リエントラント部分にリンクする場合、プログラム自体がリエントラントになるように記述する必要があります。

 **重要:** プログラム RMSPOOL を使用する場合、マクロ NATUTM および NURENT の SPOOL パラメータは SPOOL=RMSPOOL に設定する必要があります。

Natural UTM インターフェイスは、次のパラメータをプログラム RMSPOOL に渡します。

アドレス（フォーマット／長さ）	内容	
第1アドレス（A2）	ファンクションコード。使用可能なファンクションコードは次のとおりです。	
	OP	出力ファイルを開く必要があります、最初の出力レコードが渡されます。
	PR	それ以降の出力レコードが渡されます。
	CL	出力ファイルを閉じる必要があります。
第2アドレス	出力レコード（出力するデータ）。  出力レコードの最初のバイトには、改行／フォームフィードの制御文字が含まれます（ファンクションコード CL の場合、これはダミーアドレスです）。	
第3アドレス（B2）	出力レコード長（フィード文字を含む）。  （ファンクションコード CL の場合、これはダミーアドレスです）。	
第4アドレス（A8）	プリンタ名。	
第5アドレス	出力バッファ。  このバッファは、特定の目的のために RMSPOOL でワークエリアとして利用できます（RMSPOOL もリエントラントである場合）。このバッファは、ダイアログ入力とダイアログ出力の間で、RMSPOOL 専用として使用できます。	
第6アドレス（B2）	出力バッファ長。	
第7アドレス（A8）	現在のユーザー ID（システム変数 *USER と同じ）。	
第8アドレス（A8）	現在の端末 ID（システム変数 *INIT-ID と同じ）。	

アドレス (フォーマット/長さ)	内容
第 9 アドレス (A8)	現在の Natural ライブラリ名 (システム変数 *LIBRARY-ID と同じ)。
第 10 アドレス (A8)	現在の Natural プログラム名 (システム変数 *PROGRAM と同じ)。
第 11 アドレス (A4/B4)	リターンコード。  RMSPPOOL が呼び出されると、Natural UTM インターフェイスはこのフィールドを 2 進数 0 に設定します。RMSPPOOL から制御が返されると、2 進数の 0 以外の値はエラーコードと評価され、その値が端末画面に表示されます (表示可能な場合)。またこの値が SYSLST にも出力されます。

### SHUTALL - マクロ NATUTM

マクロ NATUTM に SHUTALL パラメータで指定されたユーザー出口は、UTM タスクの終了時 (KDCSHUT $n$ ) に常にアクティブになります。デフォルトでは、このユーザー出口は SHUTEX1 です。

SHUTALL で指定したユーザー出口を使用する場合、Natural UTM インターフェイスに対する KDCDEF のパラメータ USAGE=SHUT を、KDCROOT の生成時に設定しておく必要があります。

### SHUTLST - マクロ NATUTM

マクロ NATUTM に SHUTLST パラメータで指定されたユーザー出口は、最後の UTM タスクの終了時 (KDCSHUT $n$ ) にアクティブになります。デフォルトでは、このユーザー出口は SHUTEX2 です。

SHUTLST で指定したユーザー出口を使用する場合、Natural UTM インターフェイスに対する KDCDEF のパラメータ USAGE=SHUT を、KDCROOT の生成時に設定しておく必要があります。

### STRTALL - マクロ NATUTM

マクロ NATUTM に STRTALL パラメータで指定されたユーザー出口は、UTM タスクの開始時に常にアクティブになります。デフォルトでは、このユーザー出口は STARTEX です。

### STRTFST - マクロ NATUTM

マクロ NATUTM に STRTFST パラメータで指定されたユーザー出口は、最初の UTM タスクの開始時にアクティブになります。デフォルトでは、このユーザー出口は STAPPLX です。

## TRMIOEX - プログラム FREXIT

ユーザー出口 TRMIOEX は、フォーマットされた入力または出力メッセージごとにアクティブになります。

## UINPEX - マクロ NURENT

マクロ NURENT に **UINPEX** パラメータで指定されたユーザー出口は、端末メッセージの送信後にアクティブになります。デフォルトでは、このユーザー出口は INPSCR です。

UTM 環境の Natural は、次のパラメータをユーザー出口に渡します。

アドレス (フォーマット/長さ)	内容
第1アドレス	アドレス入力バッファ。
第2アドレス (B2)	アドレスメッセージ長。

## UOUTEX - マクロ NURENT

マクロ NURENT に **UOUTEX** パラメータで指定されたユーザー出口は、端末メッセージの送信前にアクティブになります。デフォルトでは、このユーザー出口は OUTSCR です。

UTM 環境の Natural は、次のパラメータをユーザー出口に渡します。

アドレス (フォーマット/長さ)	内容
第1アドレス	アドレス出力バッファ。
第2アドレス (B2)	アドレスメッセージ長。

## UVGEXIT - マクロ NATUTM

ユーザー出口 UVGEXIT は、UTM DC トランザクションの開始時、再スタート時、終了時（正常終了または異常終了）にアクティブになります。現在のタスク ID (*Vorgangskennzeichen*、KCKNZVG) は、ユーザー出口ルーチンに渡されます。

## WHCEXT - マクロ NURENT

ユーザー出口 WHCEXT を使用すると、出力データが FPUT によって UTM に渡される前に、この出力データを変更することができます。WHCEXT が呼び出されると、レジスタ 9 には出力データのアドレスが格納され、レジスタ 13 には保存エリアのアドレスが格納されます。

WHCEXT はリエントラントにする必要があります、Natural UTM アプリケーションのリエントラント部分にリンクする必要があります。詳細については、アセンブルされるマクロ NURENT (ラベル 'NUWHC') のソースリストを参照してください。

## UTM 環境での非同期トランザクション処理

非同期トランザクションを開始するには、Natural UTM インターフェイスのサービスルーチン NATASYN を呼び出す必要があります。

Natural プログラムでの非同期トランザクションの開始は、次のパターンに従ってダイナミックパラメータを渡すことで実行されます。

```
...
COMPRESS dynamic parameters INTO field
CALL 'NATASYN'[parameter area
SET CONTROL 'H'
WRITE NOTITLE NOHDR field
[WRITE ...]
INPUT 'text' ifield (A1)
END
```

ダイナミックパラメータの長さが 250 バイトを超える場合（つまり、複数の WRITE ステートメントが必要な場合）、パラメータエリアは、CALL 'NATASYN' ステートメントで渡す必要があります。

パラメータエリアは、非同期トランザクションを UTM DPUT で開始する場合、つまり、特定の時刻に開始する場合も必要になります。ダイナミックパラメータの合計長が 3,750 バイトを超えることは許可されません。CALL 'NATASYN' のパラメータエリアは、次のような構造になります。

バイト	内容	
01~02	WRITE ステートメントの数。	
03	DPUT 時刻インジケータ：	
	R	相対時間
	A	絶対時間
	<i>blank</i>	FPUT
04~06	年内での日付。	
07~08	時間。	
09~10	分。	
11~12	秒。	

バイト 03~12 の内容に対しては、Siemens UTM の各ドキュメントで DPUT 呼び出しについて説明しているのと同じ規則が適用されます。Natural のプログラミング例は、Natural ライブラリ SYSEXTP（プログラム STARTAS1、ASYNMULT、STARTAS、READAUTO、AWINDOW1、AWINDOW2）にあります。



非同期トランザクション処理については、KDCROOT、KDCDEF、および UTM スタートアップジョブを必要に応じて変更する必要があります（Siemens UTM のドキュメントを参照）。

非同期トランザクションのすべての UTM TAC は、マクロ NATUTM のパラメータ ASYNTAC で非同期 TAC に対して一意な ID として定義した文字シーケンスで始まる必要があります。逆に同期トランザクションに対する UTM TAC の最初の 5 文字は、この文字列にすることはできません。

組み合わせのトランザクション処理（つまり、1 つの UTM アプリケーション内での場合と、2 つの UTM アプリケーション間での場合の両方）は実行できません。

### Natural UTM アプリケーションでの非同期処理

Natural UTM アプリケーション内でトランザクションを非同期で処理する場合、マクロ NATUTM のパラメータ SYAPPLI と ASAPPLI のオペランドは NO に設定する必要があります（これはデフォルト値）。

例：

これは Natural UTM アプリケーション内で非同期トランザクションを初期化する Natural プログラムの例です。

```
* STARTAS - EXAMPLE OF THE INITIALIZATION FOR ASYNCHRONOUS
*          TRANSACTION WORKING WITHIN ONE UTM APPLICATION
*          PARMS ARE SEPARATED BY ','
*          SUBLIST IN STACK IS SEPARATED BY ';'
FORMAT LS=145
RESET PARM1(A144) PRDEST(A8) LTDEST(A8)
MOVE 'PRINTER1' TO PRDEST           /* --> Note 1
MOVE *INIT-ID TO LTDEST             /* --> Note 2
COMPRESS 'SENDER=' PRDEST ',OUTDEST=' LTDEST ','
'MENU=F,STACK=(LOGON APPL1;READAUTO)'
INTO PARM1 LEAVING NO               /* --> Note 3
CALL 'NATASYN'                      /* --> Note 4
SET CONTROL 'H'                     /* --> Note 5
WRITE NOTITLE NOHDR PARM1           /* --> Note 6
INPUT 'ASYNTASK INVOKED - HOPEFULLY' IFELD(A1) /* --> Note 7
END
```

注	
1	プリンタの名前（ダミー）がフィールド PRDEST に移動されます。
2	UTM 端末の論理名がフィールド LTDEST に移動されます。
3	Natural によって送信および処理されるメッセージは、次の情報とともにアセンブルされます。 <ul style="list-style-type: none"> <li>■ プリンタ名（この例では、8 文字の任意の名前）、</li> <li>■ メッセージの送信先とする端末の論理名（KCLOGTER）、</li> <li>■ メインメニューの非表示（これは指定する必要があります）、</li> </ul>

注	
	<ul style="list-style-type: none"> <li>■ アプリケーション名 (Natural ログオン ID)、</li> <li>■ 非同期UTMトランザクションで開始し、実行するプログラムの名前 (この例では、READAUTO)。</li> </ul>
4	サブルーチン NATASYN (マクロ NATUTM内の) が呼び出されると、マーカーは非同期トランザクションを初期化する必要があると示すように設定されます。サブルーチン NATASYN は、Natural 以外のプログラムを呼び出す場合の規則に従います。
5	Natural オフラインレポートがアクティブになります。
6	メッセージ (PARM1) が非同期トランザクションとして FPUT によって出力されます。
7	Natural オフラインレポートは、INPUT ステートメントによって「オフに切り替えられます」。このステートメントには少なくとも 1 つの入力フィールドが必要です。

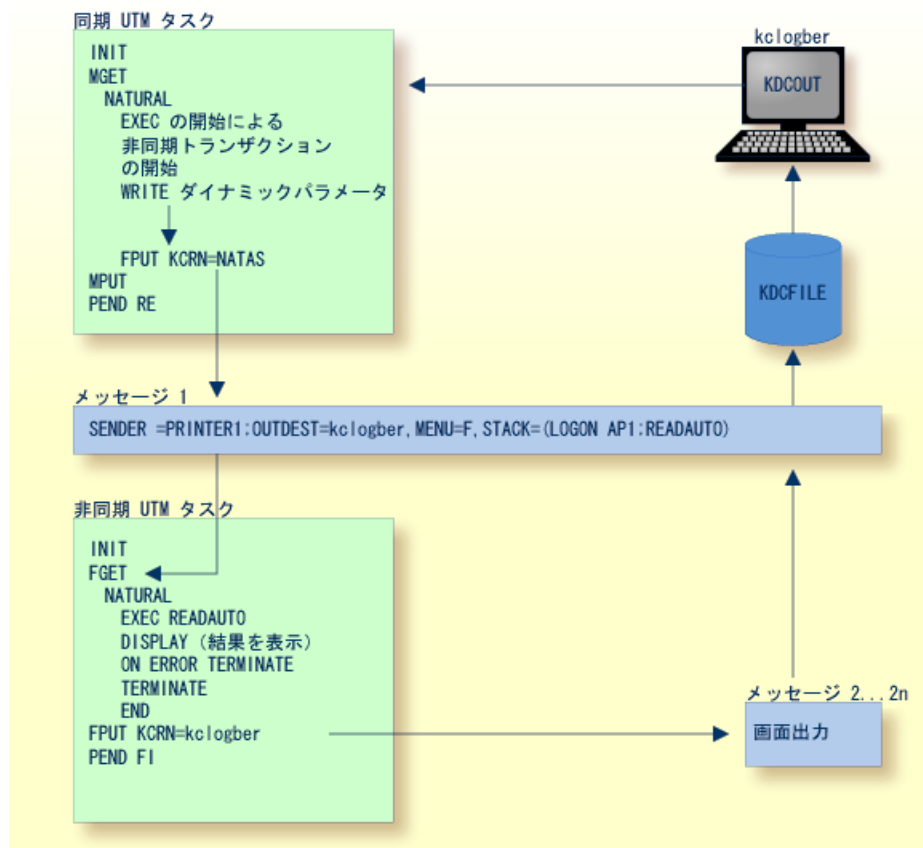
非同期的に実行するプログラムの例：

```

* READAUTO - EXAMPLE FOR ASYNCHRONOUS TRANSACTION WORKING
READ (75) AUTOMOBILES BY MAKE
WRITE MAKE MODEL HORSEPOWER YEAR
LOOP
ON ERROR DO                                /* --> Note 1
  ERRNO(A4) = *ERROR
  WRITE '*****'
      /'ERROR NO.: ' ERRNO ' IN ASYNCHRONOUS PROGRAM ' *PROGRAM
      /'*****'
  TERMINATE
DOEND
TERMINATE                                /* --> Note 2
END
    
```

注	
1	ON ERROR ルーチンは、非同期的に実行する各プログラムで定義する必要があります。このルーチンは、必ず TERMINATE ステートメントで終了するようにします。
2	非同期的に実行する各プログラムは、TERMINATE ステートメントで終了する必要があります。

1つの Natural UTM アプリケーション内での非同期トランザクションのロジック：



## 2つの Natural UTM アプリケーション間での非同期処理

2つの Natural UTM アプリケーションの間でトランザクションを非同期的に処理する場合、同期アプリケーションの論理 UTM 端末名 (LTERM 名) をマクロ NATUTM のパラメータ SYAPPLI で定義する必要があります。また、非同期アプリケーションの論理 UTM 端末名 (LTERM 名) はマクロ NATUTM のパラメータ ASAPPLI で定義する必要があります。

**注意:** KDCROOT と KDCDEF は、両方のアプリケーションに対して適切な設定で生成する必要があります。

例：

```
NUSTART NATUTM SYAPPLI=LNATUTM,ASAPPLI=LNATASY,...
ASYNDRV NATUTM SYAPPLI=LNATUTM,ASAPPLI=LNATASY,...
```

同期アプリケーションの例：

```
OPTION GEN=ALL,ROOTSRC=INPUT.KDCROOT.KDCNATS
ROOT.KDCNATS
MAX KB=400,SPAB=8192,NB=5120,TRMSGLTH=520
MAX APPLINAME=NATUTM,APPLIMODE=S,KDCFILE=(NATUTM,S)
MAX TASKS=10,ASYNTASKS=5
.
.
EXIT PROGRAM=NUSTART,USAGE=START
EXIT PROGRAM=NUSTART,USAGE=SHUT
EXIT PROGRAM=FREEIT,USAGE=FORMAT
.
.
DEFAULT PROGRAM COMP=ASSEMB
PROGRAM NUSTART
PROGRAM FREEIT
PROGRAM NUERROR
.
.
DEFAULT TAC TYPE=D,PROGRAM=NUSTART,EXIT=NUERROR,CALL=BOTH,...
TAC NAT,ADMIN=NO,TIME=0
TAC NAT1,ADMIN=NO,TIME=0
.
.
DEFAULT TAC TYPE=A,PROGRAM=NUSTART,EXIT=NUERROR,CALL=FIRST,...
TAC NATSY
TAC NATAS
.
.
PTERM NATASY,PRONAM=HOST,PTYPE=APPLI,TERMN=A1,LTERM=LNATASI
DEFAULT PTERM PRONAM=PCDF,PTYPE=T9750,TERMN=FE,CONNECT=N,STATUS=ON
PTERM DFDSS001,LTERM=DF97501
PTERM DFDSS002,LTERM=DF97502
.
.
LTERM LNATASY
DEFAULT LTERM USAGE=D,STATUS=ON,ANNOAMSG=Y,RESTART=YES
LTERM DF97501
LTERM DF97502
.
.
SFUNC F1,RET=21Z
.
.
END
```

非同期アプリケーションの例：

```

OPTION GEN=ALL,ROOTSRC=INPUT.KDCROOT.KDCNATA
ROOT.KDCNATA
MAX KB=400,SPAB=8192,NB=5120,TRMSGLTH=520
MAX APPLNAME=NATASY,APPLIMODE=S,KDCFILE=(NATASY,S)
MAX TASKS=10,ASYN TASKS=5
.
.
EXIT PROGRAM=ASYNDRV,USAGE=START
EXIT PROGRAM=ASYNDRV,USAGE=SHUT
EXIT PROGRAM=FREEIT,USAGE=FORMAT
.
.
DEFAULT PROGRAM COMP=ASSEMB
PROGRAM ASYNDRV
PROGRAM FREEIT
PROGRAM NUERROR
.
.
DEFAULT TAC TYPE=D,PROGRAM=ASYNDRV,EXIT=NUERROR,CALL=BOTH,...
TAC NAT,ADMIN=NO,TIME=0
TAC NAT1,ADMIN=NO,TIME=0
.
.
DEFAULT TAC TYPE=A,PROGRAM=ASYNDRV,EXIT=NUERROR,CALL=FIRST,...
TAC NATSY
TAC NATAS
.
.
PTERM NATUTM,PRONAM=HOST,PTYPE=APPLI,TERMN=A1,LTERM=LNATUTM
DEFAULT PTERM PRONAM=PCDF,PTYPE=T9750,TRMN=FE,CONNECT=N,STATUS=ON
PTERM DFDSS001,LTERM=97501
PTERM DFDSS002,LTERM=97502
.
.
LTERM LNATUM
DEFAULT LTERM USAGE=D,STATUS=ON,ANNOAMSG=Y,RESTART=YES
LTERM DF97501
LTERM DF97502
.
.
SFUNC F1,RET=21Z
.
.
END

```

Siemens UTM のドキュメントも参照してください。非同期アプリケーションが主に非同期トランザクションの処理を目的としている場合、このアプリケーションを約 64 KB の小さい（ローカルの）Natural スワッププールとともに生成することでストレージを節約できます。

**!** **重要:** パラメータ **SYNTAC** で定義した TAC (デフォルト値は NATSY) は常に、TYPE=A で KDCDEF に対して定義する必要があります。これは、UTM TAC の命名規則の例外です。また同期アプリケーションで UTM TACCLASS の概念を利用する場合、非同期 TAC クラスをこの TAC に割り当てる必要もあります。

2つの **Natural UTM** アプリケーションの間で非同期トランザクションを初期化する **Natural** プログラムの例:

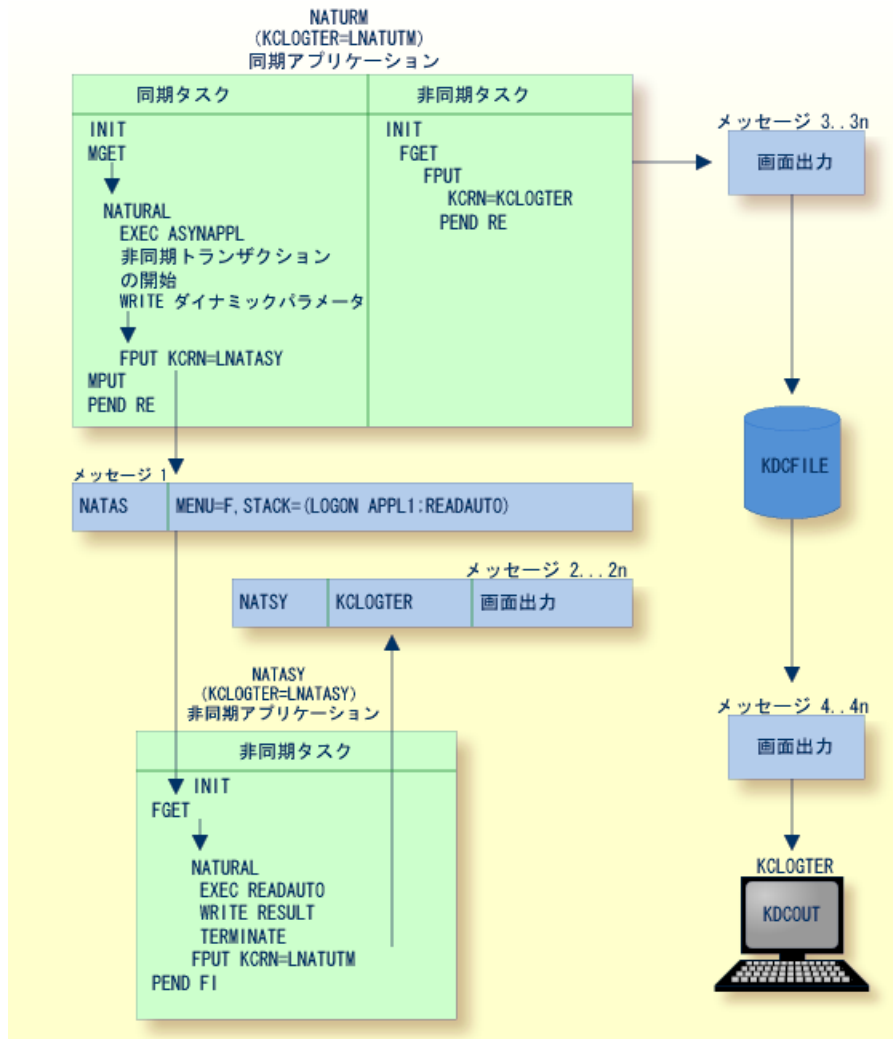
```
* ASYNAPPL - EXAMPLE OF INITIALIZATION FOR ASYNCHRONOUS
*          TRANSACTION WORKING BETWEEN TWO UTM APPLICATIONS
FORMAT LS=145
RESET PARM1(A144) PRDEST(A8) LTDEST(A8) ASYNTAC(A8)
MOVE 'PRINTER1' TO PRDEST /* --> Note 1
MOVE *INIT-ID TO LTDEST /* --> Note 2
MOVE 'NATSY' TO ASYNTAC /* --> Note 3
COMPRESS 'NATAS' 'SENDER=' PRDEST ',OUTDEST=' LTDEST
          ',ASYNNAME=' ASYNTAC ',
          'MENU=F,STACK=(LOGON APPL1;READAUTO)'
          INTO PARM1 LEAVING NO /* --> Note 4
CALL 'NATASYN' /* --> Note 5
SET CONTROL 'H' /* --> Note 6
WRITE NOTITLE NOHDR PARM1 /* --> Note 7
INPUT 'ASYNTASK INVOKED - HOPEFULLY' IFELD(A1) /* --> Note 8
END
```

注	
1	プリンタの名前 (シミュレーション) がフィールド PRDEST に移動されます。
2	UTM 端末の論理名 (KCLOGTER) がフィールド LTDEST に移動されます。
3	「フリーランニング」メッセージを非同期アプリケーションから同期アプリケーションに送信するための標準的な TAC は、フィールド ASYNTAC に格納されます。マクロ NATUTM のパラメータ <b>SYNTAC</b> の説明も参照してください。
4	<p>Natural によって送信および処理されるメッセージは、次の情報とともにアセンブルされます。</p> <ul style="list-style-type: none"> <li>■ 非同期トランザクション用のトランザクションコード (NATAS)、</li> <li>■ プリンタ名 (この例では、8 文字の任意の文字列)、</li> <li>■ メッセージの送信先とする端末の論理名 (KCLOGTER)、</li> <li>■ フリーランニングメッセージを非同期アプリケーションから同期アプリケーションに送信するための標準的な TAC の名前、</li> <li>■ メインメニューの非表示 (これは指定する必要があります)、</li> <li>■ アプリケーションの名前 (Natural ログオン ID)、</li> <li>■ 非同期トランザクション/アプリケーションで開始し、実行する Natural プログラムの名前 (この例では、READAUTO)。</li> </ul>

注	
5	サブルーチン NATASYN (マクロ NATUTM 内の) が呼び出されると、非同期トランザクションを初期化するためのマーカーが設定されます。このサブルーチンは、Natural 以外のプログラムを呼び出す場合の規則に従います。
6	Natural オフラインレポートがアクティブになります。
7	メッセージ (PARM1) が非同期トランザクションとして FPUT で出力されます。
8	Natural オフラインレポートは、少なくとも1つの入力フィールドがある INPUT ステートメントによって「オフに切り替えられます」。

非同期的に実行するプログラム (READAUTO) は、1つの Natural UTM アプリケーションで非同期トランザクション処理を実行する場合の規則に従う必要があります。

2つの Natural UTM アプリケーション間での非同期トランザクションのロジック：



## UTM 環境での印刷

- ローカルの非プールプリンタの使用
- NATSPOOL (Natural Advanced Facilities) の使用



## ■ その他のスプーリングシステム

### ローカルの非スプールプリンタの使用

Natural プログラムでスプーリングなしでローカルプリンタを使用したい場合は（つまり、UTM で FPUT を実行する）、次の例のように処理を実行します。

例：

```
* TESTPRNT - TEST FOR THE Natural OFFLINE REPORT
RESET PARAM(A9)
REDEFINE PARAM (PARAM1(A1) PARAM2(A8))
MOVE 'H' TO PARAM1 /* --> Note 1
MOVE 'PRINTER1' TO PARAM2 /* --> Note 2
SET CONTROL PARAM /* --> Note 3
READ (50) AUTOMOBILES BY MAKE
WRITE NOTITLE NOHDR MAKE MODEL HORSEPOWER YEAR /* --> Note 4
LOOP
EJECT
INPUT 'PRINT ORDER WAS EXECUTED' IFELD(A1) /* --> Note 5
END
```

注	
1	Natural オフラインレポートは、フィールド PARAM の最初の位置に H を格納することでアクティブになります。
2	論理 UTM プリンタ名は、フィールド PARAM の第 2 の位置から定義します。
3	SET CONTROL ステートメントは、フィールド PARAM の内容とともに Natural オフラインレポートをアクティブにし、プリンタの名前を指定します。Natural バージョン 1 を使用して記述した既存のプログラムの互換性を保持するために、プログラム CMLIST および NATPRNT は引き続き使用することができます。「 <a href="#">UTM 環境の Natural で使用するユーティリティプログラム</a> 」を参照してください。
4	出力レコードは、FPUT を使用して、Natural UTM インターフェイスから UTM に渡されます。
5	INPUT ステートメント（少なくとも 1 つの入力フィールドが必要）は、Natural オフラインレポートを非アクティブにします。

UTM を生成する場合は、ローカルプリンタの使用に必要なすべての手順を行う必要があります。詳細については、Siemens UTM のドキュメントを参照してください。適切な UTM 管理コマンドを使用すると、定義したプリンタへの接続が存在していることを確認できます。

## NATSPPOOL (Natural Advanced Facilities) の使用

マクロ NATUTM のパラメータ **SPPOOL** は、UTM 環境で NATSPPOOL を使用するために提供されています。詳細については、「[NATUTM マクロパラメータ](#)」を参照してください。また、『*Natural Advanced Facilities*』ドキュメントにある BS2000/OSD 固有のインストール情報も参照してください。

非同期 UTM トランザクションの出力を NATSPPOOL で実行する場合、TERMINATE ステートメントは END OF TRANSACTION ステートメントの後に指定する必要があります。

## その他のスプーリングシステム

ユーザー出口 RP2PRNT は、他のスプーリングシステムに対するインターフェイスとして提供されています。このユーザー出口は、マクロ NURENT のパラメータ **SPPOOL** に REPRO-2000 が指定されている場合にアクティブになります（この値は、すべてのスプーリングシステムに使用する必要があります）。

Natural UTM アプリケーションのリエントラント部分とリンクする必要があるので、ユーザー出口ルーチン RP2PRNT はリエントラントにする必要があります。

バッファ処理など、Natural からの出力レコードの転送ロジックは、マクロ NURENT（ラベル CMWTERM および CMWHC）や、マクロ NATUTM の適切なルーチンで確認することができます。

または、ユーザー出口 **RMSPOOL** を使用することもできます。上記の「[ユーザー出口](#)」を参照してください。

前の段落で説明されている場合を除き、Software AG は、その他のスプーリングシステムに対してこのインターフェイスをサポートしていません。


## Natural 以外のプログラムの呼び出し

Natural 以外のプログラムは、プログラム間通信の標準的なレジスタ規則に従って呼び出します。呼び出すプログラムがリエントラント（共有コードを使用）である場合、Natural パラメータモジュール（マクロ NTPRM）のプロファイルパラメータ **CSTATIC** で定義し、UTM アプリケーションのリエントラント部分とリンクすることができます。それ以外の場合、次の手順のいずれかを実行することができます。

- プログラムはランタイムにダイナミックにロードできます。これを実行するには、Natural パラメータモジュールのプロファイルパラメータ **LIBNAM** で定義したライブラリ、または UTM 開始ジョブに指定した **BLSLIB** ライブラリに、このプログラムを含める必要があります。
- プログラムは、Natural UTM アプリケーションのフロントエンド部分とリンクすることができます。リンクするには、プログラムの名前をマクロ NATUTM のパラメータ **LINK**、**LINK2**、**LINK3**、または **LINK4** のオペランドに定義する必要があります。Natural UTM インターフェイスのフロントエンド部分にすでに存在する **ENTRY** に対して、**EXTRN** 参照があるプログラムの

場合、常にこの手順が必要になります。Natural UTM インターフェイスは、この方法で定義されたプログラムに対して TABLE マクロコールを実行します。この処理では、ダイナミックローダーの LINK テーブルにエントリが作成されます。これは、Natural プログラムで別のプログラムを呼び出す場合、それらのプログラムをダイナミックにロードする必要がないことを示します。

どちらの場合でも、呼び出される Natural 以外のプログラムの最大数は、マクロ NATUTM のパラメータ CDYNAM で定義する必要があります。「[NATUTM マクロパラメータ](#)」を参照してください。

 **注意:** マクロ NATUTM のパラメータ KB を YES に設定した場合、Natural は常に UTM 通信エリア KB (*Kommunikationsbereich*) のアドレスを最初のパラメータアドレスとして渡します。これはプロファイルパラメータ CSTATIC で定義したプログラムには適用されません。

## チェーンで結ばれた部分的な UTM プログラムの呼び出し

PEND FI(NISH) ではなく、PEND PR(OGRAM) で Natural セッションを終了 (FIN または TERMINATE) するための方法がいくつか提供されています。これらの方法では、UTM の別の部分的なプログラムを呼び出すことができます。

- 呼び出される部分的な UTM プログラムの UTM TAC は、Natural セッションの開始時に Natural ダイナミックパラメータ PROGRAM を使用して渡すことができます。例えば、以下のようになります。

```
STACK=(LOGON APPL1;MENU),PROGRAM=NAT10
```

- 呼び出される部分的な UTM プログラムの UTM TAC は、マクロ NATUTM のパラメータ PENDPR のオペランドで定義できます。例えば、以下のようになります。

```
NATUTM PENDPR='NAT10'
```

- ユーティリティプログラム TACSWTCH を使用することができます。

いずれの場合でも、Natural UTM インターフェイスは、Natural セッションの終了時に UTM TAC NAT10 で PEND PR(OGRAM) を実行します。つまり、この TAC に関連付けられた部分的な UTM プログラムが開始されます。

PEND PR(OGRAM) を実行するもう 1 つの方法は、その目的のために定義したファンクションキーを押す方法です。このキーを押すと Natural セッションが中断されますが、終了はしません。PEND PR(OGRAM) によって呼び出された部分的な UTM プログラムから制御が返されると、Natural セッションは中断されていたポイントから処理を続行することができます。パラメータ PRKEY も参照してください。UTM の別の部分的なプログラムに UTMTAC を定義せずに PEND PR(OGRAM) のファンクションキーを押した場合、適切なエラーメッセージが表示されます。



**注意:** プログラム NUEXAMPL、UTMNAV、および **UTMCOB** では、Natural UTM インターフェイス（したがって、Natural 自体）と通信する部分的な UTM プログラムに必要なロジックの例が示されます。「[Software Exchange](#)」のプログラム UTMCOB および UTMNAV の説明を参照してください。

## Natural UTM アプリケーションでの Natural 以外のプログラムからの Adabas の呼び出し

---

Natural プログラムが Adabas の呼び出しも含んでいる Natural 以外のプログラムを呼び出す場合、Adabas コントロールブロック内の適切なフィールドを現在の Adabas ユーザー ID とともに指定する必要があります。この場合、Natural UTM インターフェイスで CSECT ADACALL を生成します。

ADACALL には、マクロ NATUTM のパラメータ **ADACALL** で定義したエントリが格納されます（このパラメータのデフォルト値は ADABAS です）。

このエントリは、すべての CALL [ADABAS] USING ... に対してアクティブになります。現在の Adabas ユーザー ID は Adabas コマンドブロックのフィールド ADDITIONS2 に渡され、Adabas の呼び出しのそれ以降の処理は Adabas インターフェイスモジュール ADALNN に渡されます。パラメータ **ADACALL** も参照してください。

## UTM タスクの異常終了

---

Natural パラメータモジュールの CANCEL パラメータの値（デフォルトは大文字の \*CANCEL）を入力することで、Natural セッション（したがって UTM タスクも）が異常終了する可能性があります。

# 36

## UTM 環境の Natural - パート 4

---

■ Natural UTM アプリケーションのアカウントティング .....	332
■ UTM 環境の Natural で使用するユーティリティプログラム .....	333
■ Software Exchange .....	338
■ UTM TACCLASS の概念 - プライオリティ制御 .....	340
■ Natural UTM アプリケーションの生成 .....	351
■ Natural UTM アプリケーションの最適化 .....	354
■ 複数のアプリケーションと 1 つの共通 Natural .....	356
■ ダイナミック Natural パラメータの入力と定義 .....	357
■ UTM ユーザーの再スタート .....	357
■ Adabas プライオリティ制御 .....	358

## Natural UTM アプリケーションのアカウントティング

Natural UTM アプリケーションでのリソース利用管理の向上を目的として、アカウントティングレコードをユーザー出口 ACCINIT および ACCEXIT によって利用可能にします。

- ユーザー出口 ACCINIT は、各ダイアログステップの開始時に Natural UTM インターフェイスによってアクティブにされます。
- ユーザー出口 ACCEXIT は、マクロ NURENT のパラメータ ACCNT に基づいて、Natural UTM インターフェイスによってアクティブにされます。

ACCNT=DIAL	ユーザー出口 ACCEXIT は、各ダイアログステップの終了時にアクティブになります。
ACCNT=APPL	ユーザー出口 ACCEXIT は、アプリケーションの変更のたびに（新しい Natural ログオン ID）、アクティブになります。

どちらの場合でも、セッションの終了時にアカウントティングレコードが提供されます（FIN システムコマンドまたは TERMINATE ステートメント）。

### アカウントティングレコードの構造

0 - 7	論理 UTM 端末名	DS	CL8
8 - 15	ユーザー ID	DS	CL8
16 - 23	現在の Natural アプリケーション名	DS	CL8
24 - 27	Adabas の呼び出しの数	DS	F
28 - 31	累積メッセージ長	DS	F
32 - 35	サブルーチンを含む Natural での経過時間（ミリ秒）	DS	F
36 - 37	出力するページ数	DS	H
38 - 39	端末 I/O 転送の数	DS	H
40 - 49	（ユーザーエリア）	DS	CL10
50 - 51	未使用	DS	CL2
52 - 55	Adabas コマンドタイム（ミリ秒）	DS	F
56 - 63	最後のトランザクションプログラムの名前	DS	CL8

アカウントティングレコードのユーザーエリアは、必要に応じて、追加的なアプリケーション固有のアカウントティング情報に使用することができます。アカウントティングエリアは、ユーザー固有の UTM 通信エリア KB (*Kommunikationsbereich*) にあります。

UTM KB の現在のアドレスは、必要に応じて、マクロ NATUTM のエントリ CMKBADR で検出することができます。それ以外の場合、マクロ NATUTM のパラメータ KB のオペランドは YES に設定する必要があります。この場合、Natural は通信エリアのアドレスをすべてのサブルーチン呼び出しの最初のパラメータとして渡します。

ユーザー出口ルーチン **ACCEXIT** は、アカウントレコードを Adabas ファイル、共有シーケンシャル PAM データセット、またはタスク固有の SAM データセットに保存することができます。プログラム **ACCEXIT** では、アカウントレコードを保存する方法の例が示されます。「**Software Exchange**」を参照してください。

## UTM 環境の Natural で使用するユーティリティプログラム

UTM 環境の Natural で使用するためのユーティリティプログラムがいくつか提供されています。

これらのプログラムを使用する場合、次のルールが適用されます。

- これらのユーティリティをアSEMBルする場合、Natural および UTM のマクロライブラリを使用する必要があります。
- 特定のプログラムを使用する場合：
  - プログラムの名前はマクロ NATUTM のパラメータ **LINK** または **LINK2** で指定する必要があります。
  - プログラム自体は、Natural UTM アプリケーションのフロントエンド部分とリンクする必要があります。

これらのユーティリティプログラムについて、以下で簡単に説明します。

**NATDUE** | **INPTEX** | **NATPRNT** | **UTMTAC** | **TACSWTCH**

インターフェイス、有効なパラメータ値、ロジックの概要などの詳細な情報は、それぞれのプログラムのメンテナンスログで確認できます。

### ユーティリティプログラム NATDUE

プログラム **NATDUE** を使用すると、ユーザーが現在のダイアログステップにデータを入力したかどうか、単に **EM/DÜ** または **DÜ** が押されたのかを Natural プログラム内で確認することができます。


**NATDUE** を呼び出す場合、ユーティリティプログラム **INPTEX** を使用する必要があります。プログラム **INPTEX** は、フォーマット出口モジュール **FREXIT** でユーザー出口 **INPTEX** に対応し、各ダイアログステップでデータが入力されたかどうかをチェックします。このテストの結果に従って、それ以降プログラム **NATDUE** によるチェックの対象となるフラグが、通信エリア **KB** (*Kommunikationsbereich*) 内で設定されます。


NATDUE を呼び出す Natural プログラムの例：

```
* PROG1 - EXAMPLE FOR CALLING THE SUBROUTINE 'NATDUE'
RESET P1(A1) ...
...
INPUT USING MAP ...
CALL 'NATDUE' P1
IF P1 = 'Y' DO ... /* INPUT FROM USER
IF P1 = 'N' DO ... /* NO INPUT FROM USER
IF P1 = 'E' DO ... /* ERROR
...
END
```

## ユーティリティプログラム INPTEX

ユーティリティプログラム INPTEX は、フォーマット出口 [FREXIT](#) にある同じ名前のユーザー出口に対応します。

 **重要:** INPTEX は、Natural UTM アプリケーションのフロントエンド部分とリンクする必要があります。

 **注意:** このプログラムに変更を加える場合は（例えば端末画面の特定の行に入力されたデータを無視するようにするなど）、ユーザーの責任において行ってください。

このプログラムの機能は、各入力メッセージをチェックして、端末からの入力があるかどうか、または単に EM/DÜ または DÜ キーが押されただけなのかを確認することです。

マクロ NATUTM のパラメータ [LINK](#) または [LINK2](#) でプログラム名 INPTEX を定義する必要はありません。

## ユーティリティプログラム NATPRNT

プログラム NATPRNT は、ローカルプリンタの動作のために、次のような特別なサービス機能を提供しています。

- ターゲットプリンタの論理名の受け入れ、
- 有効なプリンタ名のリストによるプリンタ名の確認、
- 可変長の出力レコードを作成するためのマーカー設定。



## ユーティリティプログラム UTMTAC

プログラム UTMTAC は Natural プログラムから呼び出すことができ、現在の UTM TAC を生成します。このプログラムにより、中央の Natural プログラムは、Natural UTM アプリケーション内で UTM TAC 制御の「ナビゲーション」を実行できるようになります。

## ユーティリティプログラム TACSWTCH

ユーティリティプログラム TACSWTCH は、Natural プログラム内から PEND PR(OGRAM) に UTM TAC をダイナミックに割り当てることができるマクロです。指定された UTM TAC は、生成された UTM テーブルに対してチェックされ、適切に保存されます。また、情報は PEND PR(OGRAM) に渡すことができます。このユーティリティを使用するには、次の手順に従います。

1. 有効な UTM TAC を定義して、TACSWTCH マクロをアセンブルします。
2. 例えば、TACSWTCH TAC=(tac1,tac2,tac3,...tac n)。これらの TAC は、同様に KDCDEF で定義する必要があります。KDCROOT の生成に関しては、これらに対応する部分的な UTM プログラムに割り当てる必要があります。
3. マクロ NATUTM のパラメータ LINK~LINK4 でプログラム TACSWTCH を定義します。
4. プログラム TACSWTCH を Natural UTM インターフェイスのフロントエンド部分にリンクします。
5. インターフェイスの説明：CALL 'TACSWTCH' P1 [P2] P3

P1 (A8)	PEND PR に使用する UTM TAC が格納されます。	
P2 (A n)	指定は任意であり、PEND PR に対するメッセージの長さでデータが格納されます。	
	P2 の構造：LLLLDD.....	
	LLL	メッセージ長 (3 桁、長さフィールドなし)、最小長：000、最大長：160。
	DDD	メッセージエリア。
P3 (A1)	2つの機能があります。	
	呼び出すとき、および P3 に値 G (Go) が含まれている場合に、PEND PR(OGRAM) が次の Natural 出力 (INPUT、WRITE、DISPLAY) で実行されます。PEND PR で Natural UTM インターフェイスを呼び出した後、中断されていた場所から Natural セッションが続行されます。つまり、最後の出力がユーザーに表示されます。	
	制御が返されると、P3 には TACSWTCH からのリターンコードが格納されます。	
	表示されるリターンコード：	
	0	エラーなしで処理が実行されました。
	1	TAC テーブルで TAC が見つかりません。
	2	メッセージ長が 000 未満です。
3	メッセージ長が 160 を超えています。	

	エラーなしで TACSWTCH が呼び出された場合、FIN コマンドまたは TERMINATE ステートメントを実行するか、PEND PR のファンクションキーを押して、PEND PR(OGRAM) を実行することができます。パラメータ PRKEY を参照してください。
--	---

### 特殊な TACSWTCH 関数

次の値で最初の TACSWTCH パラメータを使用することができます。

値	説明
RESET	現在使用可能な UTM TAC がクリアされます。つまり、PEND FI でセッションが終了されます。
GETP	データは、出力バッファから呼び出し元の Natural プログラムの容量が十分にあるデータエリアに移動されます。
GETU	データは、KB ユーザー拡張から、呼び出し元の Natural プログラムの容量が十分にあるデータエリアに移動されます。

出力バッファまたは KB ユーザー拡張の最初の 2 バイト（フォーマット：2 進数）には、データ長（これらの最初の 2 バイトを含む）が格納される必要があります。

値	説明
PUTP	データは、呼び出し元の Natural プログラムの容量が十分にあるデータエリアから、出力バッファに移動されます。
PUTU	データは、呼び出し元の Natural プログラムの容量が十分にあるデータエリアから、KB ユーザー拡張に移動されます。

Natural プログラムのデータエリアの最初の 2 バイト（フォーマット：2 進数）には、データ長（これらの最初の 2 バイトを含む）が格納される必要があります。最初の 2 バイトを含めて、データが移動されます。

**PUTP** および **GETP** の例：

```

DEFINE DATA LOCAL
01 P1(A8)                /* FUNCTION CODE/UTM TAC
01 P2(A252)              /* FIRST PART OF DATA AREA
01 REDEFINE P2
  02 P21(B2)              /* DATA LENGTH INCLUDING FIRST TWO BYTES
  02 P22(A250)
01 A1(A250)              /* SECOND PART OF DATA AREA
01 P3(N1)                /* RETURN CODE
END-DEFINE
...                      /* PROGRAM LOGIC
MOVE 'PUTP' TO P1        /* MOVE FUNCTION CODE FOR TACSWTCH
MOVE 502 TO P21          /* MOVE TOTAL LENGTH OF DATA
CALL 'TACSWTCH' P1 P2 P3 /* PUT DATA INTO PRINT BUFFER
IF P2 NE 0               /* RETURN CODE CONTROLLING
    
```

```

DO... /* ERROR LOGIC
MOVE 'NAT1' TO P1 /* MOVE ADEQUATE UTM TAC
MOVE 'G' TO P3 /* EXECUTE PEND PR WITH TAC NAT1
CALL 'TACSWTCH' P1 P3
IF P3 NE 0 /* RETURN CODE CONTROLLING
DO... /* ERROR LOGIC
INPUT ' ' /* DUMMY MESSAGE FOR DRIVER CONTROL

```

現在、Natural UTM ドライバが制御を取得しており、次のロジックで動作します。

1. このドライバは、ダミーメッセージ (INPUT ' ') を無視します。
2. 部分的な UTM プログラムの場合、LENGTH=0 の MPUT、および TAC 'NAT1' の PEND PR です。
3. 部分的な UTM プログラムは、出力バッファを通じて Natural プログラムデータを取得します。出力バッファは UTM SPAB に配置され、出力バッファのアドレスは、UTM KB 内にあるフィールド 'KBAPBUFF' で定義されます。
  - Natural プログラムのデータは、出力バッファに移動されます（最初の2バイトには、この2バイトの長さフィールドを含めて、2進数フォーマットでデータ長が格納される必要があります）。
  - LENGTH=0 の MPUT、Natural UTM ドライバの TAC が指定された PEND PR が実行されます。
4. Natural UTM ドライバが制御を取得します (INIT/MGET)。
5. Natural に対して、ONLY ENTER をシミュレートします。
6. 次のように、Natural で処理を再開します。

```

MOVE 'RESET' TO P1 /* MOVE FUNCTION CODE FOR TACSWTCH
CALL 'TACSWTCH' P1 P3 /* RESET PEND PR TAC (NAT1)
IF P3 NE 0 /* RETURN CODE CONTROLLING
DO... /* ERROR LOGIC
MOVE 'GETP' TO P1 /* MOVE FUNCTION CODE FOR TACSWTCH
CALL 'TACSWTCH' P1 P2 P3 /* GET DATA FROM PRINT BUFFER
IF P3 NE 0 /* RETURN CODE CONTROLLING
DO... /* ERROR LOGIC
... /* PROGRAM LOGIC
END

```

マクロ NATUTM のパラメータ **KBSAVE** を YES に設定した場合、呼び出された部分的な UTM プログラムは、UTM KB を使用することができます（ヘッダーの末尾に最初の 12 バイトを加えた位置から）。この場合、UTM KB は SPUT で保存され（KB ヘッダーに最初の 12 バイトを加えた位置を始点とする）、SGET で更新されます。

Natural とその他の部分的な UTM プログラムの間で、トランザクションロジックに対して UTM トランザクションコードを定義する場合、次のルールが適用されます。

UTM の別の部分的なプログラムから Natural UTM ドライバに対して PEND PR を実行した場合、先行する開始 TAC が使用されない可能性があります。Natural UTM ドライバが PEND PR によって呼び出されたという事実を認識できるのは、フィールド KCTACVG 内の先行する開始 TAC の内

容がフィールド KCTACAL 内の現在の TAC と異なっている場合のみです（通常、フィールド KCTACVG には、ユーザーがアプリケーションの起動に使用した TAC が格納されます）。

## Software Exchange

Software AG のお客様は、Natural UTM アプリケーションで生じた特定のニーズに合わせてプログラムを開発してきました。これらのプログラムは、「Software Exchange」を通じて関心のあるあらゆるユーザーに公開されています。このことは、Software AG が開発し、特定の問題に対するサンプルソリューションとして示されたプログラムにも同様に適用されます。

これらのプログラムは無料でご利用いただけますが、Software AG によるメンテナンスは行われません。各プログラムの完全なドキュメントは、通常、ソースリストのメンテナンスログに含まれています。

各プログラムについて、以下で簡単に説明します。

[XAMDUSA](#) | [UTMCOB](#) | [UTMNAV](#) | [NUEXAMPL](#) | [ACCEXIT](#) | [TABMOD](#)

### プログラム XAMDUSA

このプログラムでは、呼び出し元の COBOL プログラムを対象として、現在のユーザー固有の WORKING-STORAGE SECTION が保存およびリストアされます。

これらの処理により、ユーザーが現在実行している UTM タスクにかかわらず、テーブルなどユーザー固有のデータエリアに多くのダイアログステップでアクセスできるようになります。データは、論理的／物理的にチェーンで結ばれた PAM-I/O を使用して PAM ファイルに保存されます。

### プログラム UTMCOB

プログラム UTMCOB は、Natural UTM アプリケーション内にあるユーザー固有の部分的な UTM プログラムの例です。この例はプログラムの基本的な論理構造を示し、部分的な UTM プログラムとして以下のことが可能です。

- ユーザーが、関連する UTM TAC によってアクティブにすることができます。
- Natural UTM インターフェイスをアクティブにします。これにより、ダイナミックな Natural パラメータを指定した PEND PR(OGRAM) を使用して Natural アプリケーションをアクティブにします。
- PEND PR(OGRAM) を使用して、Natural UTM インターフェイスからアクティブにすることができます。

「[チェーンで結ばれた部分的な UTM プログラムの呼び出し](#)」も参照してください。

## プログラム UTMNAV

プログラム UTMNAV も、Natural UTM アプリケーション内にあるユーザー固有の部分的な UTM プログラムの例です。

- このプログラムは、関連する UTM TAC を使用して、ユーザーまたは PEND PR(OGRAM) がアクティブにすることができます。
- 渡されたメッセージはダイナミック Natural パラメータとして解釈されます。
- プログラムロジックの情報を画面に出力します。
- 以前に受信した画面入力 (Natural ダイナミックパラメータ) は MPUT によって送信され、PEND PR(OGRAM) で Natural UTM インターフェイスに渡されます。

プログラム UTMNAV では、UTM KB を「共通」ユーザーエリアとして使用方法を例示します。

## プログラム NUEXAMPL

プログラム NUEXAMPL は、ユーザー固有の部分的な UTM プログラムの例であり、Natural プログラムとデータを交換できます。NUEXAMPL と呼び出し元の Natural プログラムのそれぞれのプログラムロジックは、NUEXAMPL のメンテナンスログで説明されています。

## プログラム ACCEXIT

プログラム ACCEXIT は、アカウントデータデータを共有 ISAM データセットに保存するプログラムの例です。Natural UTM インターフェイスのユーザー出口 ACCEXIT および SHUTEX2 が使用されます。「[Natural UTM アプリケーションのアカウントデータ](#)」も参照してください。

## プログラム TABMOD

プログラム TABMOD は Natural プログラムから呼び出すことができ、次のような機能を実行します。

- アプリケーションの起動時とアプリケーションの実行時に、一意のキーを使用して、テーブルなどのデータレコードを共通メモリプールにロードします。
- 呼び出し元の Natural プログラムの要件に応じて、データレコードを転送します。

こうして、高い頻度で必要になるデータを一度だけストレージにロードし、その後は常駐データとして保持することができます。

TABMOD は、ライブラリ NUT $nnn$ .MAC 内のマクロとして使用することができます。このプログラムには、これ自体のインストールと利用に必要なあらゆる情報が含まれています。

## UTM TACCLASS の概念 - プライオリティ制御

Natural プログラムは、Natural UTM アプリケーション内で UTM TACCLASS の概念を利用することで、UTM TAC クラスを割り当ててリソースの制御を最適化できます。

Natural UTM アプリケーションを生成するとき、および Natural プログラムを作成するときは、以下の手順に従う必要があります。

### 手順 1 : KDCDEF および KDCROOT 定義で UTM TAC と TAC クラスを指定する

例：

```
OPTION GEN=ALL,ROOTSRC=INPUT.KDCROOT.KDCNATP
ROOT KDCNATP
MAX APPLINAME=NATUTM,APPLIMODE=S,KDCFILE=(NATUTM,S)
MAX KB=400,SPAB=8192,NB=5120,TRMSGLTH=5120
MAX TASKS=10
MAX ASYNTASKS=3
...
EXIT PROGRAM=NUSTART,USAGE=START
EXIT PROGRAM=NUSTART,USAGE=SHUT
EXIT PROGRAM=FREXIT,USAGE=FORMAT
...
DEFAULT PROGRAM COMP=ASSEMB
PROGRAM NUSTART
PROGRAM FREXIT
PROGRAM NUERROR
PROGRAM KDCADM,COMP=SPL4
...
DEFAULT TAC TYPE=D,PROGRAM=NUSTART,EXIT=NUERROR,CALL=BOTH,...
TAC NAT,TIME=(3600000,5400),TACCLASS=1,...
TAC NAT1,TIME=(3600000,5400),TACCLASS=2,...
...
DEFAULT TAC TYPE=A,PROGRAM=NUSTART,EXIT=NUERROR,CALL=FIRST,...
TAC NATAS,TACCLASS=9
TAC NATAS1,TACCLASS=10
...
TACCLASS 1,TASKS=3
TACCLASS 2,TASKS=1
TACCLASS 9,TASKS=2
TACCLASS 10,TASKS=1
...
END
```

Siemens のドキュメント『*UTM Generierung und Administration (UTM Generation and Administration)*』も参照してください。

## 定義された UTM TAC での注意事項

UTM TAC	説明
NAT	これは、リソースの使用量が少ない同期トランザクション、つまり、短時間のトランザクション向けの UTM TAC です。
NAT1	これは、リソースの使用量が多い同期トランザクション、つまり、長時間のトランザクション向けの UTM TAC です。
NATAS	これは、リソースの使用量が少ない非同期トランザクション向けの UTM TAC です。
NATAS1	これは、リソースの使用量が多い非同期トランザクション向けの UTM TAC です。

## 手順 2 : UTM 開始ジョブの構造

ジョブ名は EN.NATUTM です。

例 :

```

/.NATUTM LOGON Natural,E,,TIME=10000
/SYSFILE SYSOUT=PROT.UTMSTAT
/FILE NATUTM.KDCA,LINK=KDCFILE
/ERASE NATUTM.PRINTCONTROL
/STEP
/FILE LOG.NATUTM,LINK=SYSLOG
/FILE NATUTM.SWAPFILE,LINK=PAMNAT,SHARUPD=Y
/SYSFILE TASKLIB=NAT210.MOD
/.REPEAT EXEC NATUTM.E
.UTM START FILEBASE=NATUTM
START TASKS=7
START ASYNTASKS=3
START STARTNAME=EN.NATUTM
.UTM END
/SKIP .REPEAT
/STEP
/SYSFILE SYSOUT=(PRIMARY)
/STEP
/SYSFILE SYSLST=(PRIMARY)
/CAT NATUTM.PRINTCONTROL,SHARE=YES
/PRINT LST.NATUTM.,SPACE=E
/ERASE LST.NATUTM.
/STEP
/LOGOFF NOSPOOL

```





その代わりに、Natural プログラムは、CALLNAT ステートメントで Natural サブプログラム NATTAC を呼び出すことができます。このために、INPUT 'NATTAC' ステートメントが省略されます。オペランドは CALL ステートメントのオペランドと同じです（上記を参照）。

```
CALLNAT 'NATTAC' operand1[operand2] [operand3]
```

このプロシージャは、同期トランザクションでも非同期トランザクションでも使用できます。NATTAC は、ライブラリ SYSTEM に含まれています。

### 例 1：

次の Natural プログラムは、UTM TAC を明示的に割り当てて新しい TAC クラスを指定し、次に START UTM TAC に指定を変更します。

```
* TACCLASS - EXAMPLE FOR A TACCLASS SWITCH
RESET CONTROL(A3) NEWTAC(A8) NR(N3)
REDEFINE CONTROL (ERRFLD(A1))
INPUT 'TEST FOR A TACCLASS SWITCH - NEW TAC: NAT1' IFELD(A1)
MOVE 'S=0' TO CONTROL /* SYNCHR. TAC, EXPLICIT --> Note 1
MOVE 'NAT1' TO NEWTAC /* SET NEW TAC --> Note 2
CALL 'NATTAC' CONTROL NEWTAC /* INVOKE TAC SWITCH --> Note 3
IF ERRFLD = 'E' DO /* ERROR CHECK --> Note 4
  DISPLAY 'ERROR' CONTROL 'FROM NATTAC'
  TERMINATE
DOEND
INPUT 'TACCLASS' /* ACTIVATE NEW TAC --> Note 5
READ (50) AUTOMOBILES BY MAKE /* NOW IN NEW TACCLASS --> Note 6
ADD 1 TO NR
WRITE NOTITLE NOHDR NR MAKE MODEL /* START TAC IS USED --> Note 7
LOOP
ON ERROR DISPLAY 'ERROR IN PROGRAM TACCLASS'
END
```

注	
1	値 S=0 は、それが同期トランザクションであり、TAC が CALL 'NATTAC' の第 2 パラメータで明示的に渡されることを表します。つまり、TAC テーブルは使用されません。
2	NATTAC への呼び出しのために、新しい TAC (NAT1) が設定されます。
3	NATTAC を呼び出すと、TAC クラスの変更が初期化されます。
4	サブルーチン NATTAC から制御が返された後、エラーチェックが実行されます。
5	擬似 MPUT および PEND PA が新しい TAC で実行されます。
6	プログラムは現在、NAT1 に対する TAC クラスで実行されています。
7	最初の端末出力が開始されると、START UTM TAC が再び有効になります。

この例では、UTM TAC NAT1 を使用して、AUTOMOBILE ファイルが読み取られます。最初の端末出力が開始されると、START UTM TAC (NAT) が再び有効になります。

内部処理ロジック：

NATTAC が呼び出されると、TACCLASS の変更が保留になっていることを示すフラグが UTM 通信エリア (*Kommunikationsbereich*) に設定されます。

またプログラムによって渡される UTM TAC も、ユーザー固有の通信エリアに格納されます。INPUT 'TACCLASS' の実行により、Natural からの端末出力が発生し、この出力に基づいて、UTM インターフェイスは新しい UTM TAC で MPUT と PEND 'PA' を発行します (メッセージは、Natural UTM インターフェイス自体によって受信されます)。新しい TAC クラスでメッセージを受信すると、TACCLASS 変更フラグがあるために、インターフェイスは入力エリアで ETX/DÜ のシミュレーションを実行します。その後の処理は新しい TAC クラスで実行されます。

以前の NATTAC 呼び出しのオペランド値に基づいて、端末に最初に送信されるメッセージを利用すると、ユーザーの START UTM TAC で MPUT と PEND 'PR' を発行することができます。つまり、TACCLASS の追加の変更が可能になります。

例 2：

次の Natural プログラムは、UTM TAC を明示的に割り当てて新しい TAC クラスを指定しますが、START UTM TAC への変更は行いません。

```
* TACCLAS1 - EXAMPLE FOR A TACCLASS SWITCH
RESET CONTROL(A3) NEWTAC(A8) SWOFF(A1)
INPUT 'TEST FOR A TACCLASS SWITCH - NEW TAC: NAT1' IFELD(A1)
MOVE 'S=0' TO CONTROL /* SYNCHR. TAC, EXPLICIT
MOVE 'NAT1' TO NEWTAC /* SET NEW TAC
MOVE 'Y' TO SWOFF /* NO RESET TO START TAC
CALL 'NATTAC' CONTROL NEWTAC SWOFF /* INVOKE TAC SWITCH
INPUT 'TACCLASS' /* ACTIVATE NEW TAC
FETCH 'TACCLAS2' /* NOW IN NEW TACCLASS
END
* TACCLAS2 - THIS PROGRAM IS FETCHED FROM PROGRAM TACCLAS1
RESET NR(N3)
READ (25) AUTOMOBILES BY MAKE /* TACCLASS IS NAT1
ADD 1 TO NR
WRITE NOTITLE NOHDR NR MAKE MODEL HORSEPOWER YEAR
LOOP
FETCH 'MAINMENU' /* TACCLASS = NAT1
END
```

この例では、処理が TAC NAT1 で新しい TAC クラスに割り当てられます。ユーザーの START UTM TAC への切り替えは、NATTAC の呼び出しで、第 3 のパラメータ (SWOFF) に値 Y を指定することで回避されます。

また 1 つの Natural プログラムで、複数の TACCLASS の変更を実行できます。

## 例 3 :

次の Natural プログラムは、2つの明示的な TACCLASS の変更と、1つの暗黙的な TACCLASS の変更を実行します。

```
*TACMULT - EXAMPLE FOR TWO TACCLASS SWITCHES IN ONE PROGRAM
RESET CONTROL(A3) NEWTAC(A8) SWOFF(A1) NR(N4)
INPUT 'TEST FOR 2 TACCLASS SWITCHES' IFELD(A1)
MOVE 'S=0' TO CONTROL /* SYNCHR. TAC, EXPLICIT
MOVE 'NAT1' TO NEWTAC /* SET NEW TAC
MOVE 'Y' TO SWOFF /* NO RESET TO START TAC
CALL 'NATTAC' CONTROL NEWTAC SWOFF /* INVOKE TAC SWITCH
INPUT 'TACCLASS' /* ACTIVATE NEW TAC
READ (50) AUTOMOBILES BY MAKE /* NOW IN NEW TACCLASS
ADD 1 TO NR
WRITE NR MAKE MODEL YEAR
LOOP
EJECT /* ACTIVATE NEW OUTPUT *****
MOVE 'S=0' TO CONTROL /* SYNCHR. TAC, EXPLICIT
MOVE 'NAT2' TO NEWTAC /* SET NEW TAC
CALL 'NATTAC' CONTROL NEWTAC /* INVOKE TAC SWITCH
INPUT 'TACCLASS' /* ACTIVATE NEW TAC
READ (100) AUTOMOBILES BY MAKE /* NOW IN NEW TACCLASS
WRITE MAKE MODEL YEAR /* NOW START TAC IS USED
LOOP
ON ERROR DISPLAY 'ERROR IN PROGRAM TACMULT'
END
```

前記の例では UTM TAC NAT2 を考慮していません。これは KDCROOT と KDCDEF で定義する必要があります。

WRITE、PRINT、または DISPLAY ステートメントの後で明示的な TACCLASS の変更が実行された場合、新しい TAC を割り当てる前に EJECT を発行する必要があります。この処理では、INPUT 'TACCLASS' の実行前に、端末に対して無条件の出力が実行されます。EJECT の代わりに、次のステートメントを使用することができます。

```
STACK TOP DATA 'A'
INPUT A(A1)
```

このシーケンスでも、INPUT 'TACCLASS' の実行前に、端末に対して無条件の出力が実行されません。

## 例 4 :

次の Natural プログラムは、UTM TAC を暗黙的に割り当てて新しい TAC クラスを指定し、次に START UTM TAC に指定を変更します。この例では、サブルーチン NATTAC 内で同期トランザクションに対して TAC テーブルを使用します。

```
* TACIMP1 - EXAMPLE FOR AN IMPLICIT TACCLASS SWITCH
RESET CONTROL(A3) NR(N3)
REDEFINE CONTROL (ERRFLD(A1))
INPUT 'TEST FOR AN IMPLICIT TACCLASS SWITCH' IFELD(A1)
MOVE 'S=1' TO CONTROL /* USE 1ST TAC IN TABLE --> Note
CALL 'NATTAC' CONTROL /* INVOKE TAC SWITCH
IF ERRFLD = 'E' DO /* ERROR CHECK
    DISPLAY 'ERROR' CONTROL 'FROM NATTAC'
    TERMINATE
DOEND
INPUT 'TACCLASS' /* ACTIVATE NEW TAC
READ (100) AUTOMOBILES BY MAKE /* NOW IN NEW TACCLASS
ADD 1 TO NR
WRITE NOTITLE NOHDR NR MAKE MODEL /* START TAC IS USED
LOOP
ON ERROR DISPLAY 'ERROR IN PROGRAM TACIMP1'
END
```

注：値 s=1 は、それが同期トランザクションであり、TAC が TAC テーブルの最初のエントリから取得されることを表します。これは、パラメータ TCLS1 のオペランド値として定義された TAC です（デフォルト値：NAT1）。CALL 'NATTAC' にある最初のパラメータの第3の文字は、4つのパラメータ TCLS1～TCLS4 のうちいずれを適用するかを示します。

## 例 5 :

次の Natural プログラムは、UTM TAC を暗黙的に割り当てて新しい TAC クラスを指定しますが、次に START UTM TAC に指定を変更しません。この例では、サブルーチン NATTAC 内で同期トランザクションに対して TAC テーブルを使用し、この TAC で処理を続行します。

```
* TACIMP2 - EXAMPLE FOR AN IMPLICIT TACCLASS SWITCH
RESET CONTROL (A3) SWOFF(A1) NR(N3)
REDEFINE CONTROL (ERRFLD(A1))
MOVE 'S=4' TO CONTROL /* USE 4TH TAC IN TABLE --> Note
MOVE 'Y' TO SWOFF /* NO RESET TO START TAC
CALL 'NATTAC' CONTROL SWOFF /* INVOKE TAC SWITCH
IF ERRFLD = 'E' DO /* ERROR CHECK
    DISPLAY 'ERROR' CONTROL 'FROM NATTAC'
    TERMINATE
DOEND
INPUT 'TACCLASS' /* ACTIVATE NEW TAC
READ (100) AUTOMOBILES BY MAKE /* NOW IN NEW TACCLASS
ADD 1 TO NR
WRITE NR MAKE MODEL YEAR
```

```
LOOP
ON ERROR DISPLAY 'ERROR IN PROGRAM TACIMP2'
END
```

注：値  $s=4$  は、それが同期トランザクションであり、TAC が TAC テーブルの第 4 のエントリから取得されることを表します。これは、パラメータ `TCLS4` のオペランド値として定義された TAC です（デフォルト値：NAT4）。KDCROOT および KDCDEF の例では、TAC NAT4 が定義されていません。実際に行うときは、ユーザーが適切な定義を指定する必要があります。

TAC テーブルを使用する場合、Natural プログラム内で UTM TAC を明示的にコーディングする必要がないという利点があります。Natural プログラムには、実行するトランザクションの相対的なプライオリティの「重み」があるだけです。システム管理者は、Natural プログラムを変更しなくても、UTM TAC の名前を割り当てたり変更したりすることができます。

同期トランザクションに対して TACCLASS を変更する Natural プログラムをテストする場合、注意すべきことがあります。TACCLASS 変更の適切な処理を確認する場合は、ステートメント `CALL 'NATTAC' operand1 (operand2) (operand3)` を指定しないで Natural プログラムをテストできるということです。INPUT 'TACCLASS' ステートメントで端末に出力 'TACCLASS' のみが生成される場合、そのプログラムは適切に動作しています。NATTAC の呼び出しのオペランドは、適切に設定する必要があります。KDCROOT および KDCDEF に定義されていない UTM TAC が使用された場合、UTM 処理は常にエラーコード KM01 で終了します。

#### 手順 4：1 つの Natural UTM アプリケーション内で非同期トランザクションに TAC クラスを割り当てる

Natural UTM アプリケーション内の非同期トランザクションに対する TAC クラスは、次のステートメントで変更できます。

```
CALL 'NATTAC' operand1[operand2]
```

<i>operand1</i>	値 $A=n$ を指定する必要があります。この場合、 $A$ は「非同期」を表し、 $n$ は整数値の 0~4 になります。この整数値は、非同期 TAC に対するトランザクションコードのサブルーチン NATTAC のテーブルで、トランザクションのプライオリティレベルを表します。オペランドの指定形式は、同期トランザクションのオペランドの指定形式に似ています。
<i>operand2</i>	指定は任意です。 <i>operand1</i> の値が $A=0$ である場合、目的の TAC クラスに対して UTM TAC を指定します。

非同期トランザクションのすべての UTM TAC は、マクロ NATUTM のパラメータ `ASYNTAC` で非同期 TAC に対して一意な ID として定義した文字列で始まる必要があります。逆に同期トランザクションに対する UTM TAC は、この文字列で開始することができません。

## 例 6 :

次の Natural プログラムは、UTM TAC NATAS を使用して、非同期トランザクション処理のために初期化を実行します。これは、非同期トランザクションに対する標準的な TAC です。マクロ NATUTM のパラメータ ASYNTAC の説明も参照してください。

```
* STARTAS - EXAMPLE FOR ASYNCHRONOUS TRANSACTION WORKING
*          WITHIN ONE APPLICATION - USING THE STANDARD TAC
FORMAT LS=145
RESET PARM1(A144) PRDEST(A8) LTDEST(A8)
MOVE 'PRINTER1' TO PRDEST
MOVE *INITID TO LTDEST
COMPRESS 'SENDER=' PRDEST ',OUTDEST=' LTDEST ', '
        'MENU=F,STACK=(LOGON APPL1;READAUTO)' INTO PARM1
        LEAVING NO
CALL 'NATASYN'
SET CONTROL 'H'
WRITE NOTITLE NOHDR PARM1
INPUT 'ASYNTASK INVOKED - HOPEFULLY' IFELD(A1)
END
```

## 例 7 :

次の Natural プログラムは、非同期トランザクション処理を初期化し、別の TAC クラスに対する割り当てのために UTM TAC NATAS1 を割り当てます。

```
* STASTAC - EXAMPLE FOR ASYNCHRONOUS TRANSACTION WORKING
*          WITHIN ONE APPLICATION
*          AND SWITCH TO A NEW TACCLASS
FORMAT LS=145
RESET PARM1(A144) PRDEST(A8) LTDEST(A8) CONTROL(A3) NEWTAC(A8)
REDEFINE CONTROL (ERRFLD(A1))
MOVE 'PRINTER1' TO PRDEST
MOVE *INIT-ID TO LTDEST
COMPRESS 'SENDER=' PRDEST ',OUTDEST=' LTDEST ', '
        'MENU=F,STACK=(LOGON APPL1;READAUTO)' INTO PARM1
        LEAVING NO
MOVE 'A=0'      TO CONTROL          /* ASYNCHR. TAC, EXPLICIT --> NOTE
MOVE 'NATAS1'  TO NEWTAC           /* SET NEW TAC
CALL 'NATTAC'  CONTROL NEWTAC     /* INVOKE TAC SWITCH
IF ERRFLD = 'E' DO                /* ERROR CHECK
    DISPLAY 'ERROR' CONTROL 'FROM NATTAC'
    TERMINATE
DOEND
CALL 'NATASYN'                    /* INVOKE ASYNCHRONOUS TAC
SET CONTROL 'H'
WRITE NOTITLE NOHDR PARM1
INPUT 'ASYNTAC INVOKED - HOPEFULLY' IFELD(A1)
END
```

値 A=0 は、それが非同期トランザクションであり、TAC が CALL 'NATTAC' の第 2 パラメータで明示的に渡されることを表します。つまり、TAC テーブルは使用されません。

```
MOVE 'A=1' TO CONTROL
CALL 'NATTAC' CONTROL
```

TAC テーブル（「[NATUTM マクロパラメータ](#)」のパラメータ TCLA1～TCLA4 を参照）を使用するためのプロシージャは、同期トランザクションのプロシージャに対応しています。

非同期的に実行するプログラムの例（READAUTO）：

```
* READAUTO - ASYNCHRONOUS Natural PROGRAM
READ (75) AUTOMOBILES BY MAKE
WRITE MAKE MODEL HORSEPOWER BODY-TYPE YEAR
LOOP
ON ERROR TERMINATE
TERMINATE
END
```

目的の UTM TAC は常に、非同期トランザクション処理を初期化する Natural プログラム内で割り当てられる必要があります（非同期トランザクション処理のために標準的な TAC を使用することは例外的な方法です。マクロ NATUTM のパラメータ [ASYNTAC](#) の説明を参照）。非同期的に実行するプログラムは、次に目的の TAC クラスで実行されます。各非同期 Natural プログラムは TERMINATE ステートメントで終了する必要があるため、プログラムの終了時に UTM DC トランザクションも終了されます（PEND 'FI'）。

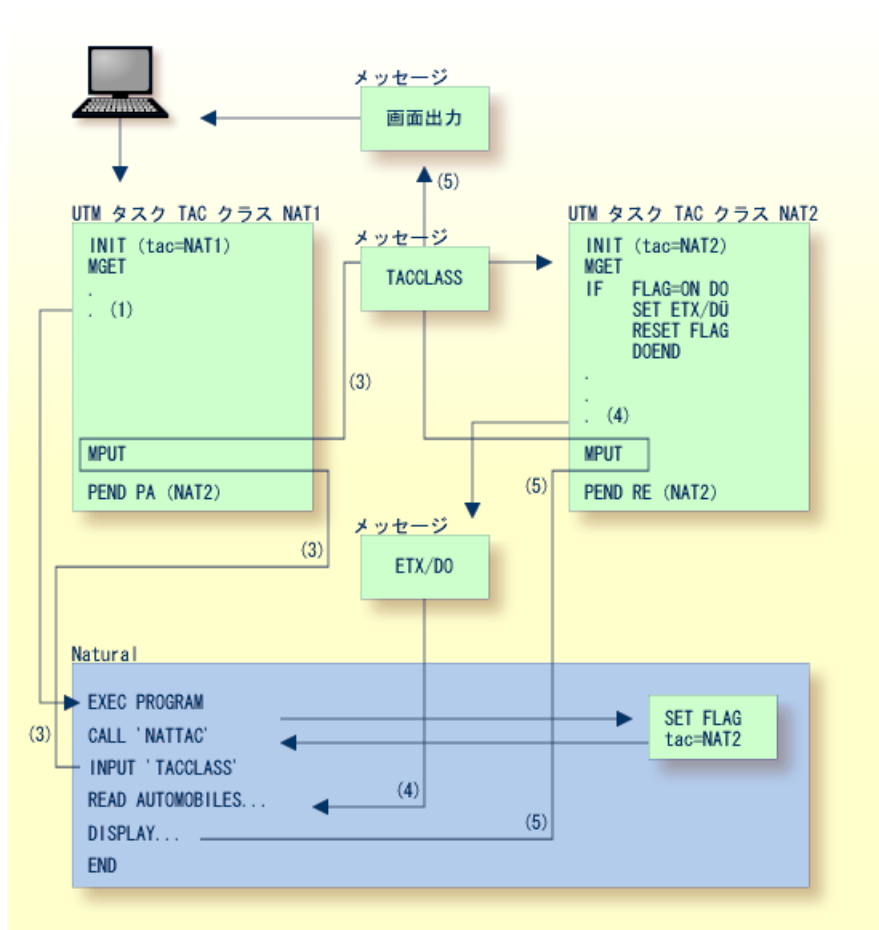
非同期トランザクション処理を初期化するプログラムは、常に同期トランザクション内で実行されます。したがって、同期トランザクションのためのプロシージャを使用して TACCLASS を変更する処理は可能です。この変更を実行できるのは、非同期 TAC を割り当てる前、または非同期トランザクションを初期化した後（INPUT ステートメント）です。

### 手順 5：2 つの Natural UTM アプリケーション間で非同期トランザクションに TAC クラスを割り当てる

2 つの Natural UTM アプリケーション間の非同期トランザクション処理のために NATTAC を呼び出す必要はありません。必要な UTM TAC は、Natural プログラムで明示的に割り当てられます。「[非同期トランザクション処理](#)」も参照してください。

### UTM TACCLASS の切り替え

次の図では、同期トランザクションのために UTM TACCLASS を切り替えるロジックを示します。





## Natural UTM アプリケーションの生成

次のプログラムとマクロは、Natural UTM アプリケーションを生成するためにアSEMBルする必要があります。

KDCROOT	UTM インターフェイスモジュール。
NATUTM	Natural UTM インターフェイスのフロントエンド部分。
BS2STUB	共通メモリプールの定義。
FREXIT	フォーマット出口モジュール（デフォルトパラメータを変更する場合のみ）。
NURENT	Natural UTM インターフェイスのリエントラント部分。
NTPRM	Natural パラメータモジュール。
NTSWPRM	スワッププールパラメータモジュール。

このリストには、Natural UTM インターフェイスのユーティリティプログラムは含まれていません。

次に、アプリケーションを生成する方法の例を示します。

```

OPTION GEN=ALL,ROOTSRC=INPUT.KDCROOT.KDCNATP
ROOT KDCNATP
MAX APPLINAME=NATUTM,APPLIMODE=S,KDCFILE=(NATUTM,S)
MAX KB=400,SPAB=8192,NB=5120,TRMSGLTH=5120
MAX TASKS=10,ASYNTASKS=3
MAX GSSBS=0,KSSBS=1
MAX LOGACKWAIT=600,RESWAIT=(600,1200),TERMWAIT=(1800,0)
MAX PGPOOL=(88,80,95),CONRTIME=2,RECBUF=(400,2048)
MAX DPUTLIMIT1=(001,23,59,59),DPUTLIMIT2=(001,23,59,59)
MAX LPUTLTH=0
*+-----+
*I          EXIT DEFINITIONS: STARTUP (CSECT NAME OF NATUTM)          I
*I          SHUTDOWN (CSECT NAME OF NATUTM)                          I
*I          FORMAT (FREXIT)                                           I
*+-----+
EXIT PROGRAM=NUSTART,USAGE=START
EXIT PROGRAM=NUSTART,USAGE=SHUT
EXIT PROGRAM=FREXIT,USAGE=FORMAT
*+-----+
*I          P R O G R A M   D E F I N I T I O N S                      I
*+-----+
DEFAULT PROGRAM COMP=ASSEMB
PROGRAM NUSTART
PROGRAM FREXIT
PROGRAM NUERROR
PROGRAM AUTOTAC
PROGRAM KDCADM,COMP=SPL4

```

```

*+-----+
*I          SYNCHRONOUS TACS FOR Natural/UTM                      I
*I          THE ERROR EXIT 'NUERROR' MUST BE DEFINED FOR EACH TAC  I
*+-----+
DEFAULT TAC TYPE=D,PROGRAM=NUSTART,EXIT=NUERROR,CALL=BOTH
TAC NAT,ADMIN=NO,TIME=0
TAC AUTOCONN
*+-----+
*I          BADTACS DEFINITION FOR Natural/UTM                    I
*I          THE ERROR EXIT 'NUERROR' MUST BE DEFINED FOR EACH TAC  I
*+-----+
TAC KDCBADTC,CALL=FIRST,PROGRAM=AUTOTAC,EXIT=NUERROR,TYPE=D
*+-----+
*I          ASYNCHRONOUS TACS FOR Natural/UTM                      I
*I          THE ERROR EXIT 'NUERROR' MUST BE DEFINED FOR EACH TAC  I
*+-----+
DEFAULT TAC TYPE=A,PROGRAM=NUSTART,EXIT=NUERROR,CALL=FIRST
TAC NATAS
TAC NATSY
*+-----+
*I          UTM ADMINISTRATOR TACS                                  I
*+-----+
DEFAULT TAC PROGRAM=KDCADM,ADMIN=Y,TYPE=D,CALL=BOTH
TAC KDCTAC
TAC KDCLOG
TAC KDCSHUT
TAC KDCAPPL
TAC KDCINF
TAC KDCUSER
TAC KDCSEND
TAC KDCDIAG
TAC KDCLTERM
TAC KDCPTERM
TAC KDCSWTCH
TAC KDCHELP
*+-----+
*I          PTERM 9750 DEFINITION                                    I
*+-----+
DEFAULT PTERM PRONAM=VR,PTYPE=T9750,TERMN=FE,CONNECT=N
PTERM DFDSS001,LTERM=DF97501
PTERM DFDSS002,LTERM=DF97502
PTERM DFDSS003,LTERM=DF97503
*+-----+
*I          LTERM DEFINITION                                        I
*+-----+
DEFAULT LTERM USAGE=D,STATUS=ON,ANNOAMSG=YES,RESTART=YES
LTERM=DF97501
LTERM=DF97502
LTERM=DF97503
*+-----+
*I          SFUNC DEFINITION                                        I
*+-----+

```

```

SFUNC F1,RET=21Z
SFUNC F2,RET=22Z
SFUNC F3,RET=23Z
SFUNC F4,RET=24Z
SFUNC F5,RET=25Z
SFUNC K1,RET=26Z
SFUNC K2,RET=27Z
SFUNC K3,RET=28Z
SFUNC K4,RET=29Z
END

```

Siemens のドキュメント『*UTM Generierung und Administration (UTM Generation and Administration)*』も参照してください。

## Natural UTM インターフェイスの生成

1. マクロ `NATUTM` のパラメータのオペランドは必要に応じて正しい値に設定し、その後にマクロ `NATUTM` をアセンブルしてください。
2. `NATUTM` マクロコールの例：

```

NUSTART NATUTM APPLNAM=NATUTM,          --> Note 1 -
                NUCNAME=NAT
vrs
,
                --> Note 2 -
                LINK=TACSWTCH           --> Note 3 -
                PARMOD=24,              --> Note 4 -
                ROLLACC=UPAM-AS,        --> Note 5 -
                ROLLTSZ=180,           --> Note 6 -
                TERMTAB=(SWP,TERMNAME), --> Note 7 -
                UMODE=(S,G)            --> Note 8

```

注	
1	Natural UTM インターフェイスのフロントエンド部分の CSECT 名は、NUSTART (デフォルト値) として指定されます。Natural UTM アプリケーションの名前は NATUTM として指定されます。
2	Natural UTM アプリケーションのリンクエディットされたリエントラント部分の名前は NATvrs として指定されます。これは、リエントラント部分がロードされる共通メモリプールの名前でもあります。
3	プログラム TACSWTCH に対して、TABLE マクロコールが実行されます。つまり、このプログラムは、Natural UTM アプリケーションのフロントエンド部分でリンクする必要があります。
4	Natural UTM アプリケーションは、24 ビットアドレッシングモードで実行されます。
5	Natural ロールファイルに対するアクセス方法は、非同期書き込みに対して P1 イベント制御を使用する UPAM として指定されます。
6	Natural ロールファイルの最大スレッドサイズは 180 (KB) として指定されます。
7	内部の端末コントロールテーブルは、Natural スワッププール内に割り当てられます。端末コントロールテーブル内のエントリを識別するために、論理端末名が使用されます。

**注**

8 Natural とのユーザーダイアログは「シングル」モードで実行されます。つまり、1つの端末で1つの Natural セッションを開始することができます。再スタート時やログオフ時のメッセージ、またフリーランニングメッセージ（非同期処理）は、ドイツ語で出力されます。

デフォルト値が適用されるので、マクロ NATUTM のその他のパラメータのオペランドは指定されません。

3. マクロ NURENT（Natural UTM インターフェイスのリエントラント部分）をアセンブルします。この例では、パラメータの変更は必要ありません。アセンブルされたマクロ NURENT の CSECT 名は NURENT です。
4. マクロ ADDON に指定された共通メモリプール定義でマクロ BS2STUB をアセンブルします。
5. Natural パラメータモジュールをアセンブルします。サンプル NTPRM マクロコールは、ローカル環境に合わせて調整する必要があります。
6. スワッププールパラメータモジュール（マクロ NTSWPRM）をアセンブルします。

### リエントラントではないフロントエンド部分とリエントラント部分のリンク

Natural UTM アプリケーションのフロントエンド部分とリエントラント部分は、提供された JCL を使用してリンクできます。JCL を使用する前にこの JCL をチェックし、ローカル環境（ライブラリ名など）に合わせて、JCL を適切に変更する必要があります。JCL の特別な機能は、REMARK ステートメントで示しています。

### Natural ロールファイルの設定

Natural スワップファイルのサイズを計算して、このファイルをリンク名 PAMNAT で割り当てる必要があります。

### Natural UTM アプリケーションの開始ジョブ

Natural UTM アプリケーションを開始する JCL の例が提供されています。JCL を使用する前にこの JCL をチェックし、JCL を適切に変更する必要があります（UTM 起動パラメータ、データセット名など）。

## Natural UTM アプリケーションの最適化

Natural UTM アプリケーションのパフォーマンスが不十分である場合、次のようなポイントを考慮してください。

- パフォーマンス低下の原因は 1 つ以上の特定の **Natural** プログラムにありますか？  
このような状況の場合は、プログラムを再設計して最適化してください。これらのプログラムは、ライブラリ SYSTP 内の Natural Monitor を使用することで特定できます。
- スワップ I/O 率が過剰に高いですか？  
ライブラリ SYSTP のプログラム MENU を使用すると、Natural スワッププールを利用する場合の効率性をチェックできます。またスワッププールに関する統計情報は、次の質問に答えるのに役立ちます。
  - 論理スワッププールの数とそのスロットの長さは適切ですか？  
SYSTP のメインメニューにある機能 SW は、Natural スワッププールの最適化を制御するためのさまざまな方法を提供しています。
  - **Natural** スワッププールに定義された大きさは十分ですか？  
スワッププールのサイズを拡張したり、スワッププールのデータスペースを生成したりすると、スワップ I/O 率が大幅に低下します。
- **Natural** バッファプールが小さすぎませんか？  
Natural バッファプールのサイズと使用率に関する情報は、Natural ユーティリティ SYSBPM で確認できます。このユーティリティについては、Natural の『ユーティリティ』ドキュメントで説明しています。
- **UTM** タスクの数が適切に選択されていますか？  
この値は、個々のトランザクションのパス長と接続される端末の数に大きな影響を受けます。
- 特定のトランザクション（いわゆる長いジョブ）が利用可能な **UTM** タスクをロードしてシステムの負荷を高めているので、短いトランザクションの処理量が低下したという可能性はありますか？  
このような場合、UTM TACCLASS の概念を取り入れて、非同期トランザクション処理機能を使用する必要があります。
- **Natural** ロールファイルを構成するエクステントが 1 つのディスクドライブ上で多くなりすぎていませんか？（物理的にチェーンで結ばれた I/O は、エクステントの境界を超過することはできません）あるいは、**Natural** ロールファイルが非常に使用頻度の高いディスクドライブに格納されていませんか？  
可能な場合は、使用頻度の低い 1 台以上のディスクドライブに Natural ロールファイルを割り当てて、各ドライブではエクステントを 1 つに制限してください。

これらの推奨事項を考慮するときは、利用可能なストレージ、ストレージページング率、ディスクやチャンネルの I/O トラフィックの負荷など、システム環境全体の要素を計算に入れる必要があります。

## 複数のアプリケーションと1つの共通 Natural

Natural の『オペレーション』ドキュメントの「BS2000/OSD 環境での Natural 共有ニュークリアス」も参照してください。

ストレージスペースを節約するには、共通の Natural リエントラント部分をクラス6ストレージの共通メモリプールに格納し、いくつかの Natural UTM アプリケーションでこれを共有することをお勧めします。Natural UTM アプリケーションを生成するときは、次の手順を実行する必要があります。

- グローバル Natural ロードプールは、次のようにモジュール CMPSTART のパラメータで定義する必要があります。

```
NAME=NATSHARE, POSI=ABOVE, ADDR=250, PFIIX=YES, SIZE=2MB
LIBR=NAT vrs. USER. MOD
```

詳細については、Natural の『オペレーション』ドキュメントの「CMPSTART プログラム」を参照してください。

### 注意:

1. NATSHARE は、リンクされる Natural リエントラント部分の名前です。また、共通メモリプールの名前でもあります。
2. パラメータ PFIIX のオペランドは YES にする必要があります。
3. パラメータ ADDR のオペランドを定義する必要があります。
4. パラメータ LIBR のオペランドには、Natural リエントラント部分のロード元となるモジュールライブラリの名前を指定する必要があります。
5. Natural UTM ドライバのリエントラント部分（マクロ NURENT のアセンブルされたモジュール）は、いくつかのアプリケーションのフロントエンド部分とリンクする必要があります。
6. パラメータ NUCNAME のオペランドは、マクロ NATUTM の各アセンブリに対して同じように定義する必要があります（この例では、NUCNAME=NATSHARE）。
7. マクロ BS2STUB のアセンブリに対する ADDON マクロ内の Natural ロードプールの定義は、すべてのアプリケーションに対して同じにする必要があります。例えば、次のように指定します。

```
STUBSHAR BS2STUB PARMOD=31, PROGMOD=ANY ADDON NAME=NATSHARE, STAT=GLOBAL
```

詳細については、Natural の『オペレーション』ドキュメントの「ADDON マクロ」を参照してください。

## 共有パラメータモジュールとアプリケーション固有のパラメータモジュールのリスト

アプリケーション固有の Natural パラメータモジュールを使用する場合、これらを Natural UTM アプリケーションのフロントエンド部分にリンクする必要があります。つまり、それぞれのフロントエンド部分にパラメータモジュールがあります。これはスワッププールパラメータモジュールにも適用されます。

ここでは、フロントエンド部分のパラメータモジュールの CSTATIC リストに定義されるアドレスのみを考慮しています。これらのアドレスがリエントラント部分を参照していることが原因でフロントエンド部分で解決できない場合、Natural はリエントラント部分のパラメータモジュールにある CSTATIC リストでこれらのアドレスを解決しようとします。したがって、アドレスをリエントラント部分で解決できる場合、フロントエンド部分とリンクするときに、未解決の CSTATIC アドレスが残される可能性があります。

リエントラント部分の CSTATIC リストは、フロントエンド部分で解決できないアドレスだけを対象として使用するの、使用するすべての CSTATIC エントリ（フロントエンド部分にあるか、リエントラント部分にあるかに関係なく）は、本来、フロントエンド部分のパラメータモジュールの CSTATIC リストに定義する必要があります。

## ダイナミック Natural パラメータの入力と定義

Natural ダイナミックパラメータの入力と定義では、次のことを実行できます。

- アプリケーションへのログオン時に、UTM TAC とともにダイナミックパラメータを入力する。
- MPUT および PEND PR(OGRAM) を使用して、UTM の別の部分的なプログラムからダイナミックパラメータを渡す。
- パラメータ MSPAR1 のオペランドでダイナミックパラメータを定義する。これらのパラメータはそのアプリケーションの全ユーザーに適用され、変更することはできません。

## UTM ユーザーの再スタート

Natural セッションを開始すると、定義された Natural ダイナミックパラメータは、最大で、マクロ NATUTM でパラメータ SVDYPRM のオペランドに定義された長さまで保存されます。ユーザーの再スタートの状況では、Natural セッションが再び開始されたとき、これらの保存されたデータが自動的に再利用されます。また、UTM の別の部分的なプログラムの PEND PR(OGRAM) の結果、Natural セッションが開始される場合も同様です。

Natural の『オペレーション』ドキュメントの「UTM 環境下の再スタート可能なグローバルスワッププール」も参照してください。

## Adabas プライオリティ制御

Adabas のプライオリティ制御と BS2000/OSD のプライオリティ制御には関連性がありません。BS2000/OSD のプライオリティ制御とは異なり、Adabas の場合は、プライオリティ値が高いほど優先度も高くなります。Adabas コマンドキューに同時に複数の要求が格納されている場合、最高のプライオリティ値を持つ要求が Adabas によって最初に処理され、その時点でコマンドキューにある他の要求のプライオリティに 1 が加算されます。

状況によっては、UTM タスクよりも低い BS2000/OSD プライオリティを Adabas タスクに割り当てる方が有効な場合もあります。

UTM トランザクションに対する Adabas プライオリティを制御するには、次の **NATUTM** マクロパラメータを使用することができます。

<b>ADAPRI</b>	UTM トランザクションに対して Adabas プライオリティ制御をアクティブ化します。
<b>APRISTD</b>	プライオリティが個別に割り当てられていないすべての UTM トランザクションに対して、標準的な Adabas プライオリティを割り当てます。
<b>TCLS</b> <i>n</i>	個々の同期 UTM トランザクションに対して、Adabas プライオリティを割り当てます。
<b>TCLA</b> <i>n</i>	個々の非同期 UTM トランザクションに対して、Adabas プライオリティを割り当てます。

UTM トランザクションに対して Adabas プライオリティ制御をアクティブにした場合（パラメータ **ADAPRI=YES**）、この設定はサブルーチン **ADACALL** で Adabas にアクセスする Natural 以外のプログラムにも適用されます。マクロ **NATUTM** のパラメータ **ADACALL** を参照してください。

上記のパラメータで異なるトランザクションに異なる Adabas プライオリティを定義し、同時に、UTM **TACCLASS** の概念を取り入れることで、非常に高度なプライオリティ制御システムを設定できます。ただし、Adabas プライオリティを UTM トランザクションに明示的に割り当てる場合は、Adabas が他の処理（例えば、TIAM やバッチ処理）に割り当てる標準的なプライオリティも考慮する必要があります。



# 索引

---

## U

UTM インターフェイス, 253

