

Natural for Mainframes

オペレーション

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 オペレーション	1
2 Natural の構成	3
3 Natural ニュークリアスへの Natural オブジェクトのリンク	5
利点	6
ULDOBJ ユーティリティ	7
ULDOBJ を使用したオブジェクトモジュールの生成	7
サブルーチンのリンクに関するその他の考慮事項	9
オブジェクトモジュール生成のオペレーティングシステム依存性	10
Natural ニュークリアスへの Natural オブジェクトのリンク例	11
4 Natural ユーザー出口	15
NATUEX1 - 認証制御のユーザー出口	16
NATSREX2 および NATSREX3 - ソート処理のユーザー出口	17
NATUSKnn - ソートキー計算のユーザー出口	18
NATPM - インバーテッド出力のユーザー出口	19
NREXPG - NATRJE のユーザー出口	20
USR0070P - エディタプロファイルのユーザー出口	21
USR2002P - ヘルプウィンドウテキスト文字列のユーザー出口	21
USR2003P - メインメニューのユーザー出口	22
5 出力/ワークファイルへの Natural ユーザーアクセスメソッド	23
NATAMUSR モジュールの説明	24
NATAMUSR モジュールのインストール	24
サードパーティ製品の起動	24
6 Natural スクラッチパッドファイル	25
Natural スクラッチパッドファイルの目的	26
スクラッチパッドファイルを定義する方法	26
スクラッチパッドファイルに保存される内容とファイルサイズの決め方	27
スクラッチパッドファイルのメンテナンス	28
7 Natural テキストモジュール	29
テキストモジュールの機能と使用方法	30
NATTEXT モジュール	30
NATTXT2 モジュール	31
NATTXT3 モジュール	34
8 Natural コンフィグレーションテーブル	37
NATCONFIG モジュール	38
NATCONFIG で使用されるマクロの概要	39
NTDVCE - 端末デバイス指定テーブル	40
NTSTAT - Natural ニュークリアスにリンクされる Natural オブジェクトの定義	40
NTCPAGE - コードページ定義	41
コードページのサポート	42
サポートされる出力デバイス	42
NTDVCE の指定	44
変換テーブル	44

大文字／小文字変換	47
CMULT エントリ	48
出力変換	48
入力変換	49
DBCS データのコード変換	49
NTTZ - タイムゾーン定義	50
9 Natural ストレージ管理	53
スレッドおよび非スレッド環境	54
バッファタイプ	54
固定バッファ	55
可変バッファ	55
バッファ特性のカスタマイズ	56
10 プロファイルパラメータの使用方法	59
11 Natural パラメータ階層	61
Natural パラメータ階層の概要	62
パラメータ使用の一般ルール	62
Natural 標準パラメータモジュール	63
代替パラメータモジュール	63
事前定義されたダイナミックパラメータ設定	64
事前定義されたユーザーパラメータプロファイル	64
ダイナミックパラメータ入力	64
Natural Security 定義	65
セッションパラメータ設定	65
プログラム／ステートメントレベル設定	65
開発環境設定	66
さまざまなパラメータ文字列の例	66
12 パラメータ値の割り当て	69
パラメータ値の割り当てのソース	70
パラメータ値のスタティックな割り当て	71
パラメータ値のダイナミックな割り当て	72
パラメータ値のランタイム割り当て用のセッションパラメータ	74
13 プロファイルパラメータの概要	75
システムファイル	76
バッファサイズ	77
外部サブプログラム	77
出力レポートおよびワークファイル	78
日付／時刻設定	78
制限	79
文字割り当て	79
端末通信	80
バッファプール	80
変換テーブル	81
コードページおよび Unicode のサポート	81
プロファイルパラメータの使用	81
コンパイラオプション	82

デバッグ	82
バッチモード	83
TP モニタ	83
データベースへのアクセス	84
Adabas での Natural	84
その他の Software AG 製品との Natural	85
その他のプロファイルパラメータ	87
セッションの初期化および終了	88
内部使用に予約済みのパラメータ	89
14 Natural パラメータモジュールの使用	91
デフォルトの Natural パラメータモジュール NATPARM の使用	92
新しい Natural パラメータモジュールの作成	92
NTPRM マクロ - Natural パラメータモジュールの作成	93
パラメータモジュールの使用の制限	93
Natural パラメータモジュールでのマクロの使用	94
15 z/OS 環境	97
16 z/OS での Natural	99
Natural サブシステム	100
共有ニュークリアス	100
TP モニタインターフェイス	100
データベース管理システムへのインターフェイス	101
z/OS 環境下のバッチモードの Natural	101
z/OS 環境のサーバーとしての Natural	101
17 z/OS 環境での認可サービスマネージャ	103
ASM の概要	104
ASM システム要件	105
ASM の操作	106
18 z/OS および z/VSE 環境下の Natural 共有ニュークリアス	111
環境に依存しないニュークリアス	112
共有ニュークリアスの作成	116
共有ニュークリアスのインストール	116
ニュークリアスへのサブ製品のリンク	116
単一環境の共有ニュークリアス	117
環境依存のニュークリアス	118
スタティックにリンクされた Natural 以外のプログラム	118
ダイナミックに呼び出された Natural 以外のプログラム	119
19 Natural ロールサーバーの機能	121
Natural ロールサーバーの概要	122
単一の z/OS システムにおけるロールサーバー	122
並列 Sysplex におけるロールサーバー	124
ロールファイルおよび LRB	125
20 Natural ロールサーバーの操作	129
ロールサーバーシステム要件	130
ロールファイルのフォーマット	132
ロールサーバーの開始	135

ロールサーバーメッセージ、コンディションコード、およびアベンドコード	138
ロールサーバー要求のリターンコードおよび理由コード	139
ロールサーバーの操作	140
ロールサーバーのパフォーマンスチューニング	141
ロールサーバーのユーザー出口	142
21 z/VSE 環境	145
22 z/VSE 環境下の Natural	147
Natural サブシステム	148
Natural 共有ニュークリアス	148
TP モニタインターフェイス	148
データベース管理システムへのインターフェイス	148
z/VSE 環境下のバッチモードの Natural	149
23 z/OS および z/VSE 環境下の Natural 共有ニュークリアス	151
24 VM/CMS 環境	153
25 VM/CMS 環境の Natural	155
Natural からの CP および CMS コマンドの発行	156
CMS プログラムスタックの読み込み	156
ハードコピー機能	157
Natural への修正の適用	157
CMS 環境下のバッチモードの Natural	157
TCP/IP 通信の使用	158
Rexx からの Natural サブプログラムの呼び出し	158
26 出力ファイルおよびワークファイルのサポート	161
出力ファイルおよびワークファイルの定義	162
アクセスメソッド STD	162
アクセスメソッド CMS	162
27 BS2000/OSD 環境	165
関連トピック	166
BS2000/OSD 専用の他の Natural 機能	167
28 BS2000/OSD 環境での Natural 共有ニュークリアス	169
Natural 共有ニュークリアスの使用ルール	170
29 Natural ロードプールの更新	173
前提条件/制限	174
手順	174
PREFRESH プログラムのキーワードパラメータ	175
30 メッセージ処理の最適化	179
画面出力処理	180
画面内容の復元	180
31 Natural でサポートされる Siemens 端末タイプ	181
9748 タイプ	182
975n シリーズ	182
9763M タイプ	183
32 9750 デバイスのファンクションキーサポート	185
キー割り当て	186

キー割り当てのモード	186
33 共通メモリプール	189
グローバル共通メモリプール	190
ローカル共通メモリプール	194
34 Natural アプリケーションでのダイナミック再ロード可能 3GL プログラムの呼び出し	203
ストレージ割り当てルール	204
スレッド作成ルール	204
アドレスモードの依存関係	204
35 出力ファイル/ワークファイルサーバー NATPWSV2	207
セットアップ	208
オペレーション	209
36 RPC サーバフロントエンド	211
セットアップ	212
37 バッチモードでの Natural	217
38 z/OS 環境下のバッチモードの Natural	219
Natural z/OS バッチモードインターフェイスに関する全般的な情報	220
Natural z/OS 生成パラメータ	220
z/OS バッチモードの Natural で使用されるデータセット	223
39 z/VSE 環境下のバッチモードの Natural	229
NATVSE - Natural z/VSE バッチモードインターフェイス	230
NTVSE マクロ - z/VSE 環境下の Natural の生成パラメータ	230
z/VSE バッチモードセッションで使用される Natural データセット	236
z/VSE ライブラリメンバに対する NATVSE 出力およびワークファイルサポート	242
NATVSE ダイナミックワークファイルアロケーション (DYNALLOC) のサポート	243
z/VSE 環境下の Natural のデバッグ機能	247
NATVSE アテンションの中断	251
40 CMS 環境下のバッチモードの Natural	253
CMS 環境下のバッチモードによる Natural の実行	254
41 BS2000/OSD 環境下のバッチモードの Natural	257
BS2000/OSD バッチモードセッションで使用される Natural データセット	258
キーワードパラメータ	262
BS2000/OSD ジョブ変数	271
42 バッチモードの Natural (すべての環境)	273
Adabas データセット	274
ソートデータセット	274
バッチモード環境のサブタスクセッションサポート	274
43 Natural バッファプール	281
44 Natural バッファプール - 全般	283
Natural バッファプールの基本動作	284
バッファプールのモニタとメンテナンス	290
Natural グローバルバッファプール	293
45 z/OS 環境下の Natural グローバルバッファプール	295

Natural グローバルバッファプールの使用	296
Natural グローバルバッファプールのオペレーティング	296
NATGBPvr 実行ジョブの例	298
ローカリゼーション	300
46 z/VSE 環境下の Natural グローバルバッファプール	301
Natural グローバルバッファプールの使用	302
Natural グローバルバッファプールのオペレーティング	303
NATGBPvr 実行ジョブの例	304
ローカリゼーション	306
47 z/OS および z/VSE 環境下の共通 Natural GBP オペレーティング機能	309
グローバルバッファプールマネージャパラメータモジュール	310
グローバルバッファプールのオペレーティング機能	310
グローバルバッファプールの機能パラメータ	312
NATBUFFER 指定の例	317
48 BS2000/OSD 環境下の Natural グローバルバッファプール	319
BS2000/OSD 環境での Natural グローバルバッファプールの使用	320
BS2000/OSD 環境下のグローバルバッファプールの設定	320
BS2000/OSD 環境下のグローバルバッファプールの管理	321
49 Natural スワッププール	323
50 Natural スワッププールの目的	325
Natural スワッププールの目的	326
Natural スワッププールを使用する利点	326
スワッププールの構造	327
51 Natural スワッププールの操作	329
ユーザーが Natural への途中で、セッションが開始されていない場合	330
ユーザーが Natural から戻ってくる場合	330
52 Natural スワッププールの初期化	333
スワッププールの初期化の制御	334
スワッププール初期化パラメータ	335
53 スワッププールのダイナミック再構成	337
スワッププールのダイナミック再構成の要件	338
統計テーブル	338
スワッププールの再構成プラステーブル	338
スワッププールの再構成マイナステーブル	339
スワッププール再構成用のパラメータ	339
スワッププール再構成の必要性チェック	340
スワッププールのダイナミック再構成の流れ	340
スワッププールのダイナミック再構成の開始	341
54 Natural スワッププールの定義	343
環境固有の要件	344
マクロ NTSWPRM のキーワードパラメータ	344
55 Natural ユーザーエリアサイズの考慮事項	353
MAXSIZE パラメータの使用	354
個別の Natural バッファのサイズの定義	354
可能性のあるエラーメッセージ	354

すべてのバッファの集計サイズの表示	355
最大サイズの計算	355
56 スワッププールデータスペース	357
追加の ESA データスペースの使用	358
ESA データスペーススロットサイズの調整	358
57 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	359
UTM 環境下の Natural グローバルスワッププールの目的	360
UTM 環境下の Natural グローバルスワッププールのインストール	360
UTM 環境下の Natural グローバルスワッププールの開始	361
グローバルスワッププールに関する情報の表示	361
58 UTM 環境下のグローバルスワッププールの終了	363
コンソールコマンドを使用した終了	364
ダンプ付きの異常終了	364
プログラムによる終了	365
59 Natural 3GL CALLNAT インターフェイス	367
60 Natural 3GL CALLNAT インターフェイス - 目的、前提条件、制限	369
3GL CALLNAT インターフェイスの目的	370
前提条件	371
制限	372
61 Natural 3GL CALLNAT インターフェイス - 使用方法、例	375
使用方法	376
環境の例	379
62 Software AG Editor の使用	383
63 エディタワークファイル	385
エディタワークファイルの構造	386
z/OS、z/VSE、および BS2000/OSD でのエディタワークファイル	387
バッチフォーマットユーティリティの使用	388
初期化中のフォーマット	388
z/OS および z/VSE でのエディタワークファイルの管理	388
BS2000/OSD でのエディタワークファイルの管理	389
VM/CMS でのエディタワークファイル	390
Complete/SMARTS でのエディタワークファイル	390
64 エディタバッファプール	391
エディタバッファプールの目的	392
フリーブロックの取得	393
エディタバッファプールの初期化	393
エディタバッファプールの再起動	394
エディタバッファプールパラメータ	394
マルチユーザー環境でのバッファプールの初期化	394
65 Natural Net Data インターフェイス NATNETTO	397
Natural Net Data ドライバの機能説明	398
全体的なメッセージレイアウト	399
ヘッダーのレイアウト	399
フォーマットバッファのレイアウト	403
バリュースワップバッファのレイアウト	408

属性バッファ	409
66 サーバーとしての Natural	411
67 z/OS 環境のサーバーとしての Natural	413
機能	414
サーバー環境での Natural ニュークリアスのインストール	415
サーバー環境における外部データセットでの出力ファイルおよびワークファイ ルの処理	415
68 z/VSE 環境のサーバーとしての Natural	417
機能	418
サーバー環境での Natural ニュークリアスのインストール	419
サーバー環境における外部データセットでの出力ファイルおよびワークファイ ルの処理	419
69 CICS 環境のサーバーとしての Natural	421
機能	422
サーバー環境での Natural CICS インターフェイスのインストール	422
制限	423
70 Natural の実行 - その他のトピック	425
71 非同期処理	427
非同期 Natural セッションの識別	428
非同期 Natural セッションの出力処理	428
予期しない／望ましくない入力の処理	429
その他のプロファイルパラメータの考慮事項	429
72 ダブルバイト文字セット	431
Natural プロファイルパラメータ SOSI	432
出力フォーマット指定	432
DBCS サポートのためのパラメータ定義	432
エディタプロファイルオプション	433
入力データのチェック	433
出力データの調整	434
Natural スタックデータ	434
DBCS 処理用のアプリケーションプログラミングインターフェイス	435
73 入力／出力デバイス	437
端末サポート	438
ライトペンサポート	438
プリンタのサポート	440
74 バックエンドプログラムの呼び出し規則	443
バックエンドプログラムの呼び出し規則 (バッチモード)	444
CICS 固有の考慮事項	445
IMS/TM 固有の考慮事項	445
サンプルバックエンドプログラム	445
75 Natural 31 ビットモードのサポート	447
76 LE サブプログラム	449
IBM LE サブプログラムのサポート	450
Natural による LE サブプログラムサポートの有効化	450
LE ランタイムオプションの引き渡し	451

LE アベンド処理	452
77 外部 SORT	453
外部 SORT のサポート	454
z/OS 固有の考慮事項	454
z/VSE 固有の考慮事項	454
BS2000/OSD 固有の考慮事項	455
索引	457

1 オペレーション

このドキュメントでは、さまざまなオペレーティングシステム上のメインフレーム環境での Natural 操作について説明します。

このドキュメントは次の項目で構成されています。

 Natural の構成	Natural オブジェクトを Natural ニュークリアスにリンクする方法について説明します。 Natural ユーザー出口、出力/ワークファイルへの Natural ユーザーアクセスメソッド、Natural スクラッチパッドファイル、Natural テキストモジュール、Natural コンフィグレーションテーブル、および Natural ストレージ管理について説明します。
 プロファイルパラメータの使用方法	Natural パラメータを設定できるさまざまなレベルの階層構造について概要を説明します。 プロファイルパラメータに値をスタティックに、ダイナミックに、およびランタイム時に割り当てる方法について説明し、使用可能なプロファイルパラメータの概要を機能別に示します。 Natural パラメータモジュールの使用方法について説明します。
 z/OS 環境	z/OS 環境で Natural をオンラインまたはバッチモードで実行する際の特別な考慮事項について概要を説明します。 Natural 認可サービスマネージャ (ASM) の機能と操作について説明します。 Natural 共有ニュークリアスの機能と使用について説明します。 Natural ロールサーバーの機能について説明します。 ロールサーバーのシステム要件、操作、パフォーマンスチューニング、および再起動可能性について説明します。
 z/VSE 環境	z/VSE 環境で Natural をオンラインまたはバッチモードで実行する際の特別な考慮事項について説明します。 Natural 共有ニュークリアスの機能と使用について説明します。
 VM/CMS 環境	Natural からの CP および CMS コマンドの発行、CMS プログラムスタックの読み込み、ハードコピー機能、Natural への修正の適用などについて説明します。 また、CMS 環境で Natural をバッチモードで使用するときの該当ピックへのリンクも示します。 出力ファイルおよびワークファイルを Natural パラメータモジュールで定義する方法について説明します。
 BS2000/OSD 環境	Natural を BS2000/OSD オペレーティングシステム環境で実行する際の特別な考慮事項について説明します。

 バッチモードでの Natural	Natural をバッチ（Adabas データセット、ソートデータセット、バッチ環境のサブタスクセッションサポート）で実行する際、および特に z/OS、z/VSE、VM/CMS、および BS2000/OSD 環境のバッチモードで実行する際の考慮事項について説明します。
 Natural バックアッププール	Natural 管理者が z/OS、z/VSE、および BS2000/OSD オペレーティングシステム環境で使用できるさまざまなストレージ管理機能について説明します。
 Natural スワッププール	TP モニタ CICS または UTM 使用時に利用できる Natural スワッププールについて説明します。
 Natural 3GL CALLNAT インターフェイス	Natural で 3GL プログラムを使用して Natural サブプログラムを呼び出し実行できるようにする Natural 3GL CALLNAT インターフェイスについて説明します。
 Software AG Editor の使用	Software AG Editor の操作方法について説明します。
 Natural Net Data インターフェイス NATNETTO	Natural Net Data インターフェイスとネットデータプロトコル定義について説明します。
 サーバーとしての Natural	Natural をサーバーおよび Natural サーバーモニタとして使用する方法について説明します。
 Natural の実行 - その他のトピック	Natural の実行に関する全般的な情報を示します。非同期処理、ダブルバイト文字セット、入力/出力デバイス、バックエンドプログラムの呼び出し規則、Natural 31 ビットモードのサポート、LE サブプログラム、および外部 SORT について説明します。

関連ドキュメント：

- インストール
- メッセージおよびコード
- TP モニタインターフェイス
- リモートプロシージャコール（RPC）
- ユーティリティ
- Software AG Editor
- Natural Security
- Natural for VSAM
- Natural for DB2
- Natural for DL/I
- Natural for SQL/DS

2 Natural の構成

このドキュメントでは、Natural コンフィグレーションについて説明します。

- **Natural** ニュークリアスへの **Natural** オブジェクトのリンク
- **Natural** ユーザー出口
- 出力/ワークファイルへの **Natural** ユーザーアクセスメソッド
- **Natural** スクラッチパッドファイル
- **Natural** テキストモジュール
- **Natural** コンフィグレーションテーブル
- **Natural** ストレージ管理

『ユーティリティ』ドキュメントの以下のトピックも参照してください。

- SYSCP ユーティリティ - コードページ管理
- SYSEXT - *Natural* アプリケーションプログラミングインターフェイス
- SYSAPI - *Natural* アドオン製品の API

3 Natural ニュークリアスへの Natural オブジェクトの リンク

- 利点 6
- ULDOBJ ユーティリティ 7
- ULDOBJ を使用したオブジェクトモジュールの生成 7
- サブルーチンのリンクに関するその他の考慮事項 9
- オブジェクトモジュール生成のオペレーティングシステム依存性 10
- Natural ニュークリアスへの Natural オブジェクトのリンク例 11

Natural ニュークリアスは、Natural カーネルを構成するサービスプログラム（メモリ管理、文字列操作、オペレーティングシステムインターフェイス、コンパイラ、ランタイム環境など）の集合体です。これは、オペレーティングシステムや TP システムに依存しません。

このドキュメントでは、Natural オブジェクトを Natural ニュークリアスにリンクする利点を説明し、さらに前に進む方法を示します。

以下のトピックについて説明します。

利点

Natural オブジェクトを Natural ニュークリアスにリンクすると、以下の利点があります。

■ パフォーマンスの向上

オブジェクトは、Natural バッファプールからではなく、ニュークリアスから実行されます。これにより、バッファプールのスペースが節約され、データベースコールも減少します。

（Natural カタログ化オブジェクトが Natural ニュークリアスにリンクされていない場合は、それらのオブジェクトは Adabas などのデータベースファイルに保存され、実行するには実際のコードをこのファイルからバッファプールにロードする必要があります）。

■ 整合性

Natural ニュークリアスにリンクされているオブジェクトは常にニュークリアスから実行されるので、オブジェクトの派生元のカタログ化オブジェクトが Natural システムファイルで削除されたり変更されたりしても影響を受けません。したがって、各 TP モニタセッション中に、オブジェクトのステータスは変化しません。ニュークリアスにリンクされているオブジェクトの新しいバージョンを取得するには、ULDOBJ（下記参照）を使用してオブジェクトをアンロードし、新しいバージョンを Natural ニュークリアスにリンクし直し、Natural モジュールを更新します（更新するということは、モジュールの新しいコピーが TP モニタセッションにロードされるということです）。

■ グローバルエラー処理

カタログ化オブジェクトがエラーを処理するために別のプログラムをフェッチ（例えば、Natural システム変数 *ERROR-TA を使用）する場合、エラー処理プログラムをバッファプールにロードできないと、元のエラーが不明になり、後発のエラーが最初のエラーをマスクして混乱することがあります。このような事態を回避するために、ユーザー作成のグローバルエラー処理プログラムをニュークリアスにリンクできます。

ULDOBJ ユーティリティ

Natural カタログ化オブジェクトを Natural ニュークリアスにリンクするには、ULDOBJ ユーティリティを使用します。ULDOBJ ユーティリティを使用して、Natural カタログ化オブジェクトからオブジェクトモジュールを生成し、Natural ワークファイルに書き込みます。次に、生成されたオブジェクトモジュールをリンケージエディタで処理し、Natural ニュークリアスにリンクします。

z/OS および z/VSE 環境：Natural 共有ニュークリアスを使用している場合は、生成されたオブジェクトモジュールはニュークリアスの環境非依存部分にリンクする必要があります。

ULDOBJ を使用したオブジェクトモジュールの生成

▶手順 3.1. ULDOBJ ユーティリティを呼び出すには

- 1 ライブラリ SYSMISC にログオンし、ULDOBJ コマンドを発行します。

```
10:12:19          ***** NATURAL OBJECT MAINTENANCE *****          2005-01-05
User: XYZ          - NATURAL ULDOBJ UTILITY -          Library: SYSMISC
                  Opsys .. z/OS

                  Specify parameters below ....

                  Object ..... _____ (Enter '.' to exit)
                  Library ..... SYSMISC_
                  OP System ... _____
```

2 以下のパラメータを指定し、確定します。

Object	処理されるカタログ化オブジェクトの名前。オブジェクトは、プログラム、サブプログラム、サブルーチン、ヘルプルーチン、またはマップのいずれでもかまいません。
Library	カタログ化オブジェクトを含むライブラリの名前。
OP System	<p>オブジェクトモジュールが生成されるオペレーティングシステムの名前（オブジェクトモジュールが従うルールはオペレーティングシステムごとに異なります）。以下のいずれかのオペレーティングシステムの名前を指定する必要があります。</p> <p>z/OS z/OS システム</p> <p>z/VSE z/VSE システム</p> <p>BS2000 BS2000/OSD システム</p> <p>CMS VM/CMS システム</p>

ULDOBJ ユーティリティに、処理された各オブジェクトについて以下の情報を含むレポートが表示されます。

- オブジェクトタイプ（プログラム、サブプログラム、サブルーチン、ヘルプルーチン、マップ、アダプタ）
- 処理されたカタログ化オブジェクトの名前
- プログラミングモード（S=ストラクチャードモード、R=レポーティングモード）
- カタログ化オブジェクトを含むライブラリの名前
- オブジェクトデッキが生成されたオペレーティングシステムの名前
- カタログ化オブジェクトのサイズと最適化コード（該当する場合）
- カタログ化オブジェクトの Natural バージョンとシステムメンテナンス（SM）レベル
- ユーザー ID および端末 ID も含めて、オブジェクトの最新のカタログに関する統計情報

最初の入力のデータが処理されると、ULDOBJ ユーティリティに別のオブジェクトとライブラリを入力するプロンプトが表示されます。同じ Natural ワークファイル用に複数のオペレーティングシステムのオブジェクトモジュールを生成することは意味がないので、オペレーティングシステムの入力は要求されません。

▶ **手順 3.2. ULDOBJ ユーティリティを終了するには**

- 最後のカタログ化オブジェクトが処理されたら、最初の入力フィールド（Object）に「.」を入力し、Enter キーを押します。

生成されたオブジェクトモジュールは、指定したオペレーティングシステムのフォーマットに従います。これは、非実行コードと以下の要素から成る再配置可能フォーマットです。

- 外部記号ディレクトリ（ESD）

- 再配置ディクショナリ (RLD)
- プログラムに対応する命令とデータのテキスト
- END ステートメント (ロードモジュールのエンドオブモジュールインジケータ)

生成されたオブジェクトモジュールは Natural ワークファイルに書き込まれ、リンケージエディタへの入力として使用されます (オペレーティングシステムによっては、ULDOBJ をバッチモードで使用した方が効果的です)。

コードをロードモジュールとして実行可能にする前に、生成されたオブジェクトモジュールを該当するオペレーティングシステムのリンケージエディタで処理する必要があります (後述の例を参照)。各ロードモジュールは、Natural ニュークリアスにリンクされて Natural コンフィグレーションモジュール NATCONFIG の `NTSTAT` エントリ定義で定義されると有効になります (「[Natural コンフィグレーションテーブル](#)」を参照)。

サブルーチンのリンクに関するその他の考慮事項

カタログ化オブジェクトは、ULDOBJ ユーティリティでアンロードし、Natural ニュークリアスにリンクした後は、Natural システムファイルから削除してかまいません。

ただし、これは「サブルーチン」タイプのオブジェクトには当てはまりません。サブルーチンには以下の2つの名前があります。

- ステートメント `PERFORM` および `DEFINE SUBROUTINE` で指定された名前
- `DEFINE SUBROUTINE` ステートメントを含むオブジェクトの名前

Natural はこれら2つの名前を内部的に関連付けますが、これはカタログ化オブジェクトが Natural システムファイル内に存在している場合のみ可能です。カタログ化オブジェクトが削除されると、この関連付けは失われ、ニュークリアスにリンクされたサブルーチンは実行できなくなります。

オブジェクトモジュール生成のオペレーティングシステム依存性

オブジェクトモジュールは、オペレーティングシステムによって異なる方法で生成されます。以下のような違いがあります。

プラットフォーム:	要件:
z/OS	<p>NAME 制御ステートメントがオブジェクトモジュールの最後のカードとして生成されます。これは置換機能を指定します。次に例を挙げます。</p> <pre>NAME TEST (R)</pre> <p>TEST はカタログ化オブジェクトの名前です。</p>
z/VSE	<p>オブジェクトモジュールのフォーマットは LIBR になります。CATALOG 制御ステートメントがオブジェクトモジュールの最初のカードとして生成され、「/」が最後のカードとして生成されます。次に例を挙げます。</p> <pre>CATALOG TEST.OBJ REPLACE=YES object module ... /+</pre> <p>TEST.OBJ はカタログ化オブジェクトの名前です。</p> <p>LIBR ユーティリティの実行時に、SYSIPT を ULDOBJ ユーティリティによって作成されたワークファイルに割り当てます (ASSIGN SYSIPT=<i>work-file-1</i>)。</p>
BS2000/OSD	<p>オブジェクトモジュールのフォーマットは LMS になります。ADD 制御ステートメントがオブジェクトモジュールの最初のカードとして生成され、END ステートメントが最後のカードとして生成されます。次に例を挙げます。</p> <pre>ADDR >TEST object module ... END</pre> <p>LMS ユーティリティの実行時に、SYSDTA を ULDOBJ ユーティリティによって作成されたワークファイルに割り当てます (SYSDTA=<i>work-file-1</i>)。生成されるファイル名は <i>Nvr</i>.MOD で、<i>vr</i> は現在の Natural のバージョンおよびリリース番号を表します。</p> <p>ユーティリティの実行中に複数のカタログ化オブジェクトがアンロードされる場合は、相互にオブジェクトデッキが追加されます。</p>

Natural ニュークリアスへの Natural オブジェクトのリンク例

例えば、SYSLIB ライブラリ内のオブジェクト LOGPROG および EDITPROG を Natural ニュークリアスにリンクする場合は、次の手順に従います。

1. リンクするカタログ化オブジェクトを指定します。

Object	Library
LOGPROG	SYSLIB
EDITPROG	SYSLIB

2. バッチ Natural ジョブストリームを設定します。z/OS 環境の場合は、以下のカードを含めます。

```
//CMWKF01 DD DSN=ULD.NAT.PGMS,UNIT=SYSDA,DISP=(,KEEP),
//          SPACE=(CYL,(3,1),,RLSE),VOL=SER=VVVVVV,
//          DCB=(RECFM=FB,BLKSIZE=800,LRECL=80)
//CMSYNIN DD *
LOGON SYSMISC
ULDOBJ LOGPROG,SYSLIB,OS
EDITPROG,SYSLIB
.
FIN
/*
```

3. リンテージエディタジョブストリームを設定します。

```
//JOB CARD JOB (ACCTING),CLASS=A,MSGCLASS=X
/*
/* GENERATE OS LOAD MODULE FROM ULDOBJ UTILITY
/*
//LINK1 EXEC PGM=IEWL,PARM='LIST,LET,XREF,NCAL,RENT,REUS'
//SYSLMOD DD DSN=NATURAL.USER.LOAD,DISP=SHR
//SYSUT1 DD UNIT=SYSDA,SPACE=(1024,(200,20))
//SYSPRINT DD SYSOUT=X
//SYSLIN DD DSN=NAT.ULD.PGMS,DISP=OLD,UNIT=SYSDA,VOL=SER=VVVVVV
/*
```

この手順で、ロードモジュール LOGPROG および EDITPROG が NATURAL.USER.LOAD データセットに配置されます。

追加のリンクエディットジョブにより、手順5でニュークリアスにリンクする前に、これらのモジュールをまとめて1つのロードモジュールとしてリンクできます。

```
//JOB CARD JOB (ACCTING),CLASS=A,MSGCLASS=X
/*
/* OPTIONAL JOB TO LINK CATALOGED OBJECTS TOGETHER
/*
//LINK2 EXEC PGM=IEWL,PARM='LIST,LET,XREF,NCAL,RENT,REUS'
//SYSLMOD DD DSN=NATURAL.USER.LOAD,DISP=SHR
//SYSUT1 DD UNIT=SYSDA,SPACE=(1024,(200,20))
//SYSPRINT DD SYSOUT=X
//SYSLIN DD *
INCLUDE SYSLMOD(LOGPROG) LOGON NATURAL PGM
INCLUDE SYSLMOD(EDITPROG) EDITOR NATURAL PGM
NAME XXXXXX(R)
/*
```

- スタティックにリンクされる Natural プログラムを、リンクされる Natural プログラムの NSTATIC テーブルのソースモジュール NATCONFIG で定義します。

```
NTSTAT INPL,TYPE=W
NTSTAT INPLLIB,TYPE=W
NTSTAT AERROR,TYPE=W
NTSTAT LOGPROG <==== your entries
NTSTAT EDITPROG <====
```

TYPE=W は、指定したプログラムへの、通常の参照ではなく「弱い」外部参照が生成されることを示します。

- Natural ニュークリアス用のリンケージエディタジョブストリームを確認し、以下を含めます。

```
/*
/* INCLUDE DDNAME AND DSN OF DATASET WHERE OBJECTS RESIDE
/*
//SYSLMOD DD DSN=NATURAL.USER.LOAD,DISP=SHR
//NATLIB DD DSN=NATURAL.V2.USER.LOAD,DISP=SHR/*
//SYSLIN DD*
...
... INCLUDE MODULES FOR NUCLEUS
...
INCLUDE NATLIB(NATPARM) NATPARM MODULE
INCLUDE SYSLMOD(LOGPROG) LOGON NATURAL PGM
INCLUDE SYSLMOD(EDITPROG) EDITOR NATURAL PGM
...
... INCLUDE ENTRY AND NAME CARDS
...
/*
```

カタログ化プロジェクトを一括してリンクした場合（手順3のオプション）は、個別のロードモジュールの代わりにこのロードモジュールをニュークリアスのリンクに含めます。

4 Natural ユーザー出口

▪ NATUEX1 - 認証制御のユーザー出口	16
▪ NATSREX2 および NATSREX3 - ソート処理のユーザー出口	17
▪ NATUSKnn - ソートキー計算のユーザー出口	18
▪ NATPM - インバーテッド出力のユーザー出口	19
▪ NREXPG - NATRJE のユーザー出口	20
▪ USR0070P - エディタプロファイルのユーザー出口	21
▪ USR2002P - ヘルプウィンドウテキスト文字列のユーザー出口	21
▪ USR2003P - メインメニューのユーザー出口	22

Natural ユーザー出口は、Natural、サブコンポーネント、またはサブ製品によって呼び出されるプログラミングオブジェクトです。通常は、サンプルのユーザー出口がソース形式で提供されています。ユーザー出口に含まれる命令は、ユーザーが記述または調整する必要があります。ユーザー出口の目的は、データの操作または決定を行うことです。ほとんどのユーザー出口は Natural プログラミング言語を利用しています。1つの小さなサブセットは、アセンブラ言語で記述する必要があります。

このドキュメントでは、以下の Natural ユーザー出口について説明します。

その他の Natural ユーザー出口とアプリケーションプログラミングインターフェイスについては、Natural サブコンポーネントまたはサブ製品のドキュメント（『Natural リモートプロシージャコール (RPC)』、『TP モニタインターフェイス』、『ユーティリティ』、アドオン製品など）で関連箇所を参照してください。

NATUEX1 - 認証制御のユーザー出口

ユーザー出口 NATUEX1 は、ユーザーセッションがアクティブ化されるときはいつでも呼び出されます。これを使用して、ユーザーに Natural の使用権限があるかどうかを判定します。この判定に使用されるセキュリティデータは、使用中のセキュリティシステム（RACF や ACF2 など）から取得できます。

NATUEX1 は、標準の呼び出し規則を使用して呼び出されます。

レジスタ	内容
15	NATUEX1 のエントリアドレス
14	Natural のリターンアドレス
13	18 ワードのセーブエリアのアドレス
1	パラメータリストのアドレス

パラメータリストには、以下の 5 つのアドレスが含まれています。

アドレス	Natural システム変数に指定する値を含む 8 バイトのフィールドをポイント
1	*INIT-USER
2	*ETID
3	*INIT-ID
4	*INIT-PROGRAM
5	*USER（このシステム変数は Natural Security ログオン時に上書きされます）

これらの 5 つの値はユーザー出口によって変更できます。

正常終了するには、ユーザー出口がレジスタ 15 に 0 を設定して制御を返す必要があります。レジスタ 15 の値が 0 でない場合は、Natural セッションはレジスタ 15 の値と同じコンディションコードで終了します。

NATUEX1 は、共有ニュークリアスまたは環境非依存ニュークリアスにリンクできます。また、代替パラメータモジュールにリンクしたり、プロファイルパラメータ RCA で実行している場合は独立したモジュールとしてリンクしたりできます。

ユーザー出口の例は、Natural ソースライブラリのメンバ XNATUEX1 として使用できます。

CICS の場合：Natural の『TP モニタインターフェイス』ドキュメントの「NCIUIDEX - ユーザー ID 出口インターフェイス」も参照してください。

NATSREX2 および NATSREX3 - ソート処理のユーザー出口

Natural には、2 つのソート処理用ユーザー出口、NATSREX2 と NATSREX3 があります。

この 2 つのユーザー出口は、Natural 固有のソートプログラムでも、外部ソートプログラムでも使用できます。出口は、ニュークリアスにリンクされてアドレスが解決されると、自動的にアクティブになります。z/OS および z/VSE 環境では、すでに多数の外部 SORT プログラムが複数の出口機能を提供しているため、NATSREX2 出口と NATSREX3 出口は Natural の内部ソートプログラムまたは BS2000/OSD 環境の外部 SORT 専用にしてもかまいません。

NATSREX2 は、Natural がレコードをソートプログラムに渡すときに常に呼び出されます。NATSREX3 は、ソートプログラムがソート実行の終了後にレコードを Natural に渡すときに呼び出されます。提供している例は、SORT に独自の照合順序を設定する方法を示しています。

ユーザー出口がアクティブ化されるときは、以下のレジスタ規則に従います。

レジスタ	内容
15	NATSREX2 および NATSREX3 のエントリアドレス
14	Natural のリターンアドレス
13	18 ワードのセーブエリアのアドレス
1	ソートレコードのアドレス
3	ソートレコードの長さ

ユーザー出口は、Natural レジスタを保護し、Natural に制御を戻すときにそれらのレジスタを復元する必要があります。

ソート出口モジュールは NAT2SORT モジュールにリンクされるため、プログラミングはリエントラントでなければなりません。ソートレコードのフォーマットと構造は変更できません。

NATUSKnn - ソートキー計算のユーザー出口

一部の言語には、ソートプログラムまたはデータベースシステムで正しいアルファベット順にソートできない文字が含まれています。システム関数 SORTKEY を使用すると、このような「正しくソートされない」文字をアルファベット順で「正しくソートされる」別の文字に変換できます。

Natural プログラム内で SORTKEY 関数を使用すると、ユーザー出口 NATUSKnn が呼び出されます。nn は現在の言語コード（システム変数 *LANGUAGE の現在の値）です。

NATUSKnn ユーザー出口は、標準 CALL インターフェイスを提供する任意のプログラミング言語で記述できます。SORTKEY で指定した文字列はユーザー出口に渡されます。ユーザー出口は、この文字列の「正しくソートされない」文字を対応する「正しくソートされる」文字に変換するようにプログラミングされている必要があります。続けて、変換された文字列が Natural プログラムで使用され、さらに処理が実行されます。

変換のために NATUSKnn でコンフィグレーションモジュール NATCONF の変換テーブル NTUTAB1 を使用できます。その場合は、それに応じて NTUTAB1 を調整する必要があります。

NATUSKnn は、標準の呼び出し規則を使用して呼び出されます。

レジスタ	内容
15	NATUSKnn のエントリアドレス
14	Natural のリターンアドレス
13	18 フルワードのセーブエリアのアドレス
1	パラメータリストのアドレス

パラメータリストには、以下のアドレスが含まれています。

オフセット	アドレスの内容
+0	Natural から渡された文字列
+4	文字列の長さ（フルワード）
+8	変換結果の文字列
+12	結果の文字列の長さ（フルワード）
+16	変換テーブル NTUTAB1

NATUSKnn は、レジスタ 14 および 15 以外のすべてのレジスタを保護し、Natural に制御を戻すときにそれらのレジスタを復元する必要があります。

正常終了するには、ユーザー出口がレジスタ 15 にリターンコード 0 を設定して制御を返す必要があります。レジスタ 15 の値が "0" でない場合は、該当する Natural エラーが発行されます。

サンプルのユーザー出口プログラム

以下のサンプルのユーザー出口がソースコード形式で提供されています。

プログラム	機能
NATUSK01	英語に適用され、文字列内のすべての英小文字を大文字に変換します。
NATUSK02	ドイツ語に適用され、異なるソート順序を提供するために、ドイツ語のウムラウト文字 ä、ö、ü、および ß を対応する置換文字 ae、oe、ue、および ss に変換します。

共有ニュークリアスを使用している場合は、NATUSK nn をニュークリアスの環境非依存部分にリンクできません。

また、代替パラメータモジュールにリンクしたり、呼び出される NATUSK nn モジュールの名前をプロファイルパラメータ RCA で指定している場合は独立したモジュールとしてリンクしたりできます。

リンケージ規則およびロード規則については、Natural の『ステートメント』ドキュメントの CALL ステートメントも参照してください。

NATPM - インバーテッド出力のユーザー出口

NATPM モジュールは、逆方向端末をサポートするために使用されます。このモジュールには、出力モード（プロファイルパラメータ PM）が I に設定されているフィールドがある場合に、端末の I/O 時に Natural から呼び出されるフィールドおよび行変換用のユーザー出口ルーチンが含まれます。

PM=I は、逆方向であることを示し、右から左に記述する言語（双方向言語など）をサポートするために使用されます。プロファイルパラメータ PM の説明も参照してください。

NATPM モジュールはソースモジュールとして提供されるので、必要に応じて変更できます。

インバートロジック

Natural は、結果属性が PM=I の各フィールド用と、ハードコピー、追加レポート、およびプライマリバッチ出力を使用して出力される各行用に呼び出されるユーザー出口ルーチンを提供します。この出口は、以下の 3 つのパラメータで呼び出します。

- インバートするソースフィールド
- インバートされたデータを受け取るターゲットフィールド
- ソースおよびターゲットフィールドの長さを示す長さフィールド

このユーザー出口ルーチンは、すべてのユーザーがソースコード形式で入手できるので、PM=I 属性によってトリガされる明示的なフィールド出口として使用してもかまいません。このようにすると、ユーザーは行内容やフィールド内容を確認および変更することができます。

フィールドユーザー出口

NATPM のユーザー出口は、属性 PM=I が設定されているフィールドごとに呼び出されます。

この属性は、Natural プログラムが設定することも、グローバル出力モードが PM=I に設定されたときに数値フィールドに自動設定されるようにすることもできます。出力の生成目的が、端末、ハードコピー、追加レポート、プライマリバッチ出力のいずれであっても同じです。

印刷デバイスの場合は、Natural ではハードウェアによる自動的なインバートは想定せず、完全行のために NATPM を再度呼び出します。この機能は、フィールドのインバートが必要ない国でも、フィールド属性に基づいて Natural とインターフェイスロジックを確立するために使用できます。

NREXPG - NATRJE のユーザー出口

NREXPG は Natural リモートジョブエントリ (NATRJE) のユーザー出口です。ジョブが完了すると、各 JCL カードはオペレーティングシステムにサブミットされる前に出口に渡されます。以下のデータを出口に使用できます。

- サブミットされる JCL カード
- リターンコードフィールド
- 現在実行中の Natural プログラムの名前
- Natural ユーザー ID
- 240 バイトのワークエリア

出口は、各呼び出しの終了後に、以下のいずれかのイベントを示すリターンコードを NATRJE に渡します。

コード	説明
0	サブミット：カードがサブミットされます。出口により、サブミット前にカードが変更されることがあります。
4	終了：カードがサブミットされます。出口は、現在のジョブの以降のカードには無効です。
8	挿入：カードがスキップされます（カードには INSERT 文字のみが含まれているという前提に基づいて）。追加指定のカードはサブミットされます。
10	削除：カードはサブミットされません。
12	現在のジョブがフラッシュされます。

NREXPG というユーザー出口の例は、Natural ソースライブラリのメンバ XNATRJE として使用できます。CSTATIC として指定されたプログラムのルールに従って、出口をアセンブルしリンクすることができます。ただし、NREXPG の CSTATIC エントリは不要です。

USR0070P - エディタプロファイルのユーザー出口

ユーザー出口ルーチン USR0070P を使用して、デフォルトプロファイル SYSTEM の Natural プログラムエディタまたはデータエリアエディタのパラメータ設定を変更できます。

エディタプロファイルの詳細については、『エディタ』ドキュメントの「エディタ-全般的な情報」を参照してください。

USR0070P は、デフォルト設定を受け取るすべてのパラメータのリストを提供します。

このユーザー出口を使用して、エディタプロファイルを FNAT システムファイル、FUSER システムファイル、またはスクラッチパッドファイルに保存するかどうかを指定することもできます。

さらに、USR0070P では DBCS サポートが考慮され、これに対応してエディタプロファイルオプション Editing in Lower Case および Dynamic Conversion of Lower Case が設定されます。

このユーザー出口ルーチンの例は、FNAT システムファイルの SYSEXT ライブラリから、オブジェクトおよびソースの両方の形式で入手できます。使用方法については、テキストメンバ USR0070T を参照してください。

USR2002P - ヘルプウィンドウテキスト文字列のユーザー出口

ユーザー出口ルーチン USR2002P を使用して、カーソルがメッセージ行にあるときにヘルプキーを押すと表示される [Current Natural Message] ウィンドウのテキスト文字列をカスタマイズできます。

オブジェクト USR2002P 自体にも、ウィンドウタイトルや説明テキストなど [Current Natural Message] ウィンドウで使用されるテキスト文字列が含まれています。記述テキストとは、フィールド名の Sh (ショートメッセージ)、Tx (ロングメッセージ)、Ex (説明)、Ac (アクション) などです。

このユーザー出口ルーチンの例は、FNAT システムファイルの SYSEXT ライブラリから、オブジェクトおよびソースの両方の形式で入手できます。使用方法については、テキストメンバ USR2002T を参照してください。

USR2003P - メインメニューのユーザー出口

ユーザー出口ルーチン USR2003P を使用して、Natural メインメニューとサブメニューの以下の設定をカスタマイズできます。

- メッセージ行の位置と色
- PF キー行の位置と色

このユーザー出口ルーチンの例は、FNAT システムファイルの SYSEXT ライブラリから、オブジェクトおよびソースの両方の形式で入手できます。使用方法については、テキストメンバ USR2003T を参照してください。

5 出力／ワークファイルへの Natural ユーザーアクセス メソッド

- NATAMUSR モジュールの説明 24
- NATAMUSR モジュールのインストール 24
- サードパーティ製品の起動 24

このドキュメントでは、Natural 出力／ワークファイルサポートを目的とするサードパーティベンダ製品のインターフェイスである Natural ユーザーアクセスメソッドについて説明します。

以下のトピックについて説明します。

NATAMUSR モジュールの説明

NATAMUSR モジュールは、ソフトウェアベンダに Natural 出力／ワークファイルを処理するための出口インターフェイス（エントリポイント NATAM9EX）を提供します。これは、実際には以下の2つの部分から成ります。

- Natural に付属の Natural ユーザーアクセスメソッドスタブ NATAMUSR
- ソフトウェアベンダ提供の Natural ユーザーアクセスメソッド出口 NATAM9EX

NATAMUSR モジュールのインストール

NATAMUSR モジュール（アクセスメソッド出口付き）は、以下のいずれかの方法でインストールできます。

- Natural ニュークリアスにリンク。
- Natural ドライバにリンク（ドライバとフロントエンドが分割されている場合）。
- 代替 **Natural** パラメータモジュールにリンク（プロファイルパラメータ PARM を使用してロードされる場合）。
- 独立したモジュールとしてリンク。この場合は、以下の Natural プロファイルパラメータが必要です。

```
RCA=(NATAM09),RCALIAS=(NATAM09,xxx),
```

xxx は、ロードライブラリ内のライブラリ内の独立モジュールです。

サードパーティ製品の起動

▶ **手順 5.1. Natural 出力／ワークファイル処理用のサードパーティ製品を起動するには**

- 関連するファイルに AM=USER を指定します（NTPRINT および NETWORK も参照）。

Natural ユーザーアクセスメソッド出口のインストールと、サードパーティ出口ハンドラの詳細については、関連するソフトウェアベンダのドキュメントを参照してください。

6 Natural スクラッチパッドファイル

- Natural スクラッチパッドファイルの目的 26
- スクラッチパッドファイルを定義する方法 26
- スクラッチパッドファイルに保存される内容とファイルサイズの決め方 27
- スクラッチパッドファイルのメンテナンス 28

このドキュメントでは、Natural スクラッチパッドファイルの目的、使用、およびメンテナンスについて説明します。

以下のトピックについて説明します。

Natural スクラッチパッドファイルの目的

内容と機能

スクラッチパッドファイルは、FNAT や FUSER と同様な Natural システムファイルで、ファイルの物理的なレイアウトも同じです。スクラッチパッドファイルを使用すると、保存画面イメージや、Natural のソース、オブジェクト (SAVE、CATALOG、STOW)、エラーメッセージのように明示的に保存されないデータなどを、システムファイル FNAT または FUSER 以外のファイルに保存できるようになります。

必要な場合

FNAT および FUSER とは対照的に、スクラッチパッドファイルは Natural セッションで必須ではありません。

しかし、システムファイルに対して読み取り専用アクセス (プロファイルパラメータ ROSY=0N) で作業している場合は、スクラッチパッドファイルを定義する必要があります。そうしないと、上記のデータを保存することができず、該当するエラーメッセージ (NAT0106) が発行されます。スクラッチパッドファイルは、読み取り専用アクセスから除外されます。

スクラッチパッドファイルを定義する方法

Software AG 製品の他のすべてのシステムファイルと同様に、スクラッチパッドファイルも論理ファイルです。スクラッチパッドファイルの論理ファイル番号は 212 です。

ニーモニック (FNAT、FUSER、FDIC など) はスクラッチパッドファイルにはないため、以下のいずれかの方法で定義する必要があります。

- Natural パラメータモジュール NATPARM の NTLFILE マクロを使用してスタティックに定義
- プロファイルパラメータ LFILE を使用してダイナミックに定義

NTLFILE および LFILE 定義の例：

LFILE パラメータ：

```
LFILE=(212,physical-dbid,physical-fnr,password,cipher-key)
```

NTLFILE マクロ：

```
NTLFILE 212,physical-dbid,physical-fnr,password,cipher-key
```

スクラッチパッドファイルに保存される内容とファイルサイズの決め方

以下のオブジェクトがスクラッチパッドファイルに保存されます。

- レコーディング
- 画面キャプチャ (NATPAGE ユーティリティ)

レコーディングユーティリティと NATPAGE ユーティリティの使用量は事前に計算できないため、関連するストレージ要件を適切に見積もるのはほとんど不可能です。ただし、サイトで必要なスクラッチパッドファイルのサイズは、保存されるレコードのタイプをよく理解することにより見積もることができます。

レコーディング

『ユーティリティ』ドキュメントで説明しているように、レコーディングユーティリティは端末コマンドを使用してアクティブにします。レコーディングは Natural ソースプログラム（または他のオブジェクトタイプ）と同様に保存されます。レコーディングのサイズは、レコーディングセッション中に行われた画面入力の数によります。レコーディングはライブラリ関連のプログラムに類似しています。

現在、スクラッチパッドファイル上のレコーディングを Natural LIST システムコマンドを使用してリストすることはできません。ただし、SYSMAIN を使用して、スクラッチパッドファイルに保存されているレコーディングをリストおよびメンテナンスすることができます。スクラッチパッドファイルの代わりに FNAT/FUSER ファイルにレコーディングを保存するには、プロファイルパラメータ RFILE を設定します。

レコーディングをシステムファイル FNAT または FUSER に保存中の場合は、ユーザーのアプリケーションプログラムで発行されるトランザクションのバックアウト (BT) の影響を受けます (中断されます)。これはレコーディング機能を使用するユーザーが経験する一般的な問題で、スクラッチパッドファイルを使用することで回避できます。

画面キャプチャ - NATPAGE

画面ページングユーティリティ NATPAGE を使用して、画面イメージを（表示された順に）スクラッチパッドファイルに保存できます。NATPAGE は、端末コマンド %P を使用してアクティブにできます。%P が発行された時点から端末コマンド %0 が入力されるまで、エンドユーザーに表示されたすべての画面がスクラッチパッドファイルに保存されます（セッションに定義している場合）。キャプチャされた画面は、端末コマンド %E を使用して表示できます。

各画面イメージについて、ページバッファとページ属性バッファの現在の内容が保存されます。これは、保存されるデータ量がセッションのプロファイルパラメータ PS/LS の設定と画面イメージ数によることを意味します。ユーザーセッション当たり可能な画面数は、プロファイルパラメータ PD によります（デフォルトは 50、有効な値は 0~255）。

ページバッファのサイズは以下の式で計算できます。

PS * LS

ページ属性バッファのサイズはダイナミックに決められます。

スクラッチパッドファイルのメンテナンス

スクラッチパッドファイルは、十分なサイズであれば、メンテナンスは不要です。

- スクラッチパッドファイル上のレコーディングは、ユーティリティ SYSMAIN を使用して、削除、コピー、移動、およびリストすることができます。
- キャプチャされた画面は、端末コマンド %E を使用して削除できます。
- ただし、SETUP レコードと保存画面イメージは Natural では一切メンテナンスできません。

スクラッチパッドファイルのスペースは、Adabas ユーティリティを使用して更新することにより再利用できます。この更新は、スクラッチパッドファイルを使用する後続の Natural セッションに影響を与えないように、非アクティブ時に行います。

7 Natural テキストモジュール

▪ テキストモジュールの機能と使用方法	30
▪ NATTEXT モジュール	30
▪ NATTXT2 モジュール	31
▪ NATTXT3 モジュール	34

このドキュメントでは、Natural テキストモジュール NATTEXT、NATTXT2、および NATTXT3 について説明します。以下のトピックについて説明します。

テキストモジュールの機能と使用方法

すべての Natural キーワード、代替キーワード、および標準出力テキストは、モジュール NATTEXT と NATTXT2 に含まれます。Natural システムコマンドと代替システムコマンドも、キーワードおよび代替キーワードとしてこれらのモジュールに含まれます。Natural エラーメッセージの置換テキストの部分は、モジュール NATTXT3 に含まれます。モジュールは、Natural ソースライブラリにはソース形式で、Natural ロードライブラリにはロードモジュール形式で含まれます。

必要に応じて、これらのモジュールに含まれている Natural キーワード、代替キーワード、およびテキストを変更できます。例えば、Natural セッション終了メッセージを英語から別の言語に変更したり、Natural キーワードを無効にしたり、同義語を追加したりできます。

NATTEXT、NATTXT2、または NATTXT3 モジュールに何らかの変更を行った場合は、変更した各モジュールをアSEMBルし、リンクエディットしてから、実行可能な Natural モジュールに含める必要があります (Natural の『インストール』ドキュメントを参照)。

NATTEXT モジュール

NATTEXT モジュールには、Natural で認識される各キーワードと代替キーワードの NTKEY および NTALT マクロが含まれます。

NATTEXT の変更

 **注意:** NATTEXT モジュールの変更は、極めて重大な理由がある場合のみにすることを強くお勧めします。いったん変更すると、Software AG の担当者が適切にメンテナンスできなくなります。

以下のルールが適用されます。

- NTKEY または NTALT マクロのキーワード値は、現在のキーワード値を目的の値に置き換えることで変更できます。
- キーワードまたは代替キーワードは、キーワード値を文字 "%" に置き換えることで無効にできます。
- モジュール内での各 NTKEY および NTALT マクロの位置は固定で、移動できません。追加の NTKEY および NTALT マクロを挿入することはできません。
- 同義語は、NTSYN マクロを使用して任意のキーワードまたは代替キーワードに割り当てることができます。1 つまたは複数の NTSYN マクロを NTKEY または NTALT マクロの後に挿入できま

す。NTSYN マクロは、同義語として使用される 1つのパラメータ値を含みます。同義語に埋め込み空白が含まれている場合は、値全体を一重引用符で囲む必要があります。

NATTEXT モジュール変更の例

以下の例は、NATTEXT モジュールを変更する方法を示しています。この例では、以下のように設定されます。

- 同義語 RECHERCHE は、キーワード FIND に使用されます。
- 同義語 LISEZ は、キーワード BROWSE に使用されます。
- キーワード GET および HISTOGRAM は無効にされます。

変更前の NATTEXT

```
STATNAM NTKEY FIND
          NTALT BROWSE
          NTALT GET
          NTALT ACCEPT
          NTALT REJECT
          NTALT HISTOGRAM
```

変更後の NATTEXT

```
STATNAM NTKEY FIND
          NTSYN RECHERCHE
          NTALT BROWSE
          NTSYN LISEZ
          NTALT %
          NTALT ACCEPT
          NTALT REJECT
          NTALT %
```

NATTXT2 モジュール

NATTXT2 モジュールには、以下を定義するマクロ NTKEYT、NTALTT、NTSYNT、および NTERMSG が含まれます。

- **標準 Natural 出力テキスト**
- **Natural システムコマンドおよびユーティリティのキーワードと代替キーワード**

■ Natural 終了メッセージとリターンコード

標準 Natural 出力テキスト

NATTXT2 モジュールは、以下の標準 Natural 出力テキストを含みます。各テキストは、言語コードを適宜設定すれば別の言語で表示することもできます（下記参照）。

- 標準出力ページヘッダーで使用されるリテラルの ページ。
- Natural システム変数 *DATG（グレゴリオ暦）で使用される各月の名前、日付編集マスク（L）、および日付編集マスクで使用される各曜日の名前。
- ENTER INPUT DATA メッセージと、エラー番号 1104、1105、および 1106 のスケルトンエラーメッセージ（オンライン入力処理中に使用）。
- システムファイルのオープンエラー（システムファイルからファイルを取得できない）に使用されるメッセージ。このエラーメッセージには、NAT8xxx 形式（xxx は 10 進形式の Adabas レスポンスコード）のエラー番号が Natural によって追加されます。
- テキスト形式で表示される位置情報としてウィンドウで使用される定数 More、Top、および Bottom。
- レポートと 33 を超えるレポートの処理方法を定義するテーブル。

NATTXT2 に含まれる値は、現在のテキストを目的のテキストに置き換えることで変更できます。月名の同義語が 9 文字を超える場合は、最初の 9 文字のみがシステム変数 *DATG によって使用されます。

NATTEXT モジュールで説明しているように、NTSYNT マクロステートメントを追加できます。ただし、NATTXT2 では 2 番目のパラメータを指定できます。このパラメータは省略可能で、同義語に使用される言語インジケータを表します。言語インジケータを指定すると、この同義語に対応する言語で使用したメッセージ出力が生成されます。また、1（デフォルト、英語）以外の言語インジケータを使用してエラーメッセージテキストを Natural システムファイルに保存している場合は、エラーメッセージは対応する言語で返されます。各言語の言語コードについては、プロファイルパラメータ ULANG を参照してください。

Natural システムコマンドおよびユーティリティのキーワードと代替キーワード

NATTXT2 モジュールは、Natural で認識される各キーワードと代替キーワードの NTKEYT および NTALTT マクロを含みます。以下に示す Natural システムコマンドおよびユーティリティ、および該当する場合はコマンドのパラメータとその値を対象とします。これらは、言語コードを適宜設定すれば別の言語で表示することもできます（下記参照）。

- すべての Natural システムコマンド（通常の場合）
- GLOBALS システムコマンド、該当する場合はパラメータとその値
- COMPOPT システムコマンド、該当する場合はパラメータとその値
- パブリックシステムコマンド（これらのシステムコマンドは永続的に有効で、Natural Security や Natural プロファイルパラメータ NC でも無効にすることはできません）

■ Natural ユーティリティ

NTKEYT および NTALTT マクロステートメントは、**NATTEXT** モジュールで説明しているように、NTKEY および NTALT マクロステートメントと同様に使用できます。

NTSYNT マクロステートメントは、「**標準 Natural 出力テキスト**」で説明しているように使用できます。

Natural 終了メッセージとリターンコード

Natural には、NTERMSG マクロで提供される多数の標準セッション終了メッセージ (NAT99...) があります。これらのメッセージはマクロ内で変更 (別の言語に翻訳するなど) できます。ID とテキストの全体の長さは最大 72 文字です。NTERMSG マクロを変更した場合は、Natural パラメータモジュールと環境依存ドライバ (ソースコードで提供されている場合) を再アセンブルしてリンクする必要があります。

メッセージ ID とテキストの他に、各標準終了メッセージには以下のいずれかの Natural システムリターンコードも含まれています。このコードも、NTERMSG マクロ内で定義されます。

コード	説明
0	正常終了
4	実行/コンパイル中にエラーが発生 (バッチモードの場合のみ)
8	サーバーランタイムエラーによる終了
12	セッション初期化エラー
16	アペンドまたはサーバー環境エラーによる異常終了

ユーザー作成の終了メッセージをすべてのリターンコード (1~255) の NATXT2 に追加できます。これらのリターンコードは、TERMINATE ステートメントを使用して発行でき、通常は Natural 終了メッセージ NAT9987 になります。

ユーザー作成の終了メッセージの場合は、対応するリターンコードを2番目のパラメータとして指定する必要があります。

プロファイルパラメータ TS を ON に設定すると、大文字変換テーブル NTUTAB1 (NATCONFIG モジュールで提供) によって終了メッセージは表示前に大文字に変換されます。

TS=ON の他に、メッセージを大文字に変換するためのパラメータが複数の Natural コンポーネントによって提供されています。詳細については、「TS - システムライブラリでプログラムからの出力を変換」で、大文字に変換するための他のパラメータに関する説明を参照してください。

ユーザー終了メッセージの例：

```
NTERMSG 'USR0077 THIS IS A SAMPLE USER MESSAGE FOR RETURN CODE 77',77
```

NATTXT3 モジュール

NATTXT3 モジュールには、Natural エラーメッセージの *:n:* プレースホルダを置き換えるために使用されるテキストの部分を実定義するマクロが含まれています。

各テキストの部分は、さまざまな言語で定義できます。各言語の言語コードについては、ULANG パラメータを参照してください。

テキストの部分は、EBCDIC および Unicode 表記で生成されます。



注意: NATTXT3 モジュールをアセンブルするには、マクロ機能 UPPER と Unicode 文字 (DC CU'unicode text') の定義をサポートするハイレベルアセンブラを使用する必要があります。

例：

Natural エラー NAT0082 (存在しないプログラムを実行しようとした場合) のテキストは、以下ようになります。

```
Invalid command, or :1: :2: does not exist in library.
```

オブジェクト NOTEXIST を実行しようとする、以下の結果となります。

```
NAT0082 Invalid command, or Program NOTEXIST does not exist in library.
```

:2: はオブジェクト名 (NOTEXIST) で置き換えられました。

:1: はテキストの部分 Program で置き換えられました。

テキストの部分は、モジュール NATTXT3 で以下のように宣言されました。

```
=====
*          PROGRAM          0002
*=====
      MSGSDEF  &LC_PGM
              SPACE
*-----
      MSGSLAN 01,Program      1  ENGLISH
      MSGSLAN 02,Programm    2  GERMAN
      MSGSLAN 03,programme   3  FRENCH
      MSGSLAN 04,programma   4  SPANISH
              SPACE
```

*-----

MSGSGEN

さらに MSGSLAN マクロを追加して、その他の言語のテキストの部分の値を入力できます。

8 Natural コンフィグレーションテーブル

▪ NATCONFIG モジュール	38
▪ NATCONFIG で使用されるマクロの概要	39
▪ NTDVCE - 端末デバイス指定テーブル	40
▪ NTSTAT - Natural ニュークリアスにリンクされる Natural オブジェクトの定義	40
▪ NTCPAGE - コードページ定義	41
▪ コードページのサポート	42
▪ サポートされる出力デバイス	42
▪ NTDVCE の指定	44
▪ 変換テーブル	44
▪ 大文字／小文字変換	47
▪ CMULT エントリ	48
▪ 出力変換	48
▪ 入力変換	49
▪ DBCS データのコード変換	49
▪ NTTZ - タイムゾーン定義	50

このドキュメントでは、NATCONFIG モジュールに含まれている Natural コンフィグレーションテーブルの一般的な情報を示します。

以下のトピックについて説明します。

以下の項目も参照してください。

■ 入力/出力デバイス

NATCONFIG モジュール

NATCONFIG モジュールには、Natural コンフィグレーションテーブルが含まれています。

 **注意:** 通常、NATCONFIG のデフォルト指定は変更する必要はないので、変更しないようにしてください。特に、下記のリストでアスタリスク (*) の付いたテーブルについては、変更する場合は事前に Software AG サポートに問い合わせてください。

ほとんどのテーブルには、ダイナミックプロファイルパラメータに加えて、Natural パラメータモジュール NATPARAM に対応するマクロがあります。NATCONFIG テーブルを変更する必要がある場合は、対応するパラメータモジュールマクロまたはダイナミックプロファイルパラメータを使用してテーブルを上書きします。NATCONFIG テーブル自体で変更を行った場合は、今後のシステムメンテナンス (SM) リリースで NATCONFIG の再度の変更と再アSEMBルが必要になることがあります。

NATCONFIG モジュールでは、Natural デフォルトコンフィグレーションテーブルの定義にマクロを使用します。

また、以下のテーブルも使用します。

- デフォルトアテンション ID テーブル。Natural に対して物理端末キーを定義します (*)。
- その他の各種テーブル (*)。

NATCONFIG で使用されるマクロの概要

以下の表に、Natural デフォルトコンフィグレーションテーブルの定義のために NATCONFIG モジュールで使用されるマクロの概要を示します。

マクロ	目的
NTDVCE *	端末タイプのテーブル。使用する端末ドライバを指定します（下記参照）。 既存の NTDVCE マクロは変更しないでください。新しく作成してください。
NTSTAT	Natural ニュークリアスにリンクされる Natural オブジェクトの定義。
NTCPAGE	コードページ定義。
NTTAB	プライマリ出力変換テーブル。
NTTAB1 NTTAB2	セカンダリ出力／入力変換テーブル。
NTUTAB1 NTUTAB2	小文字／大文字の変換用テーブル。これらのテーブルはドイツ語などの個々の文字セットに対して変更する必要があります。
NTTABA1 NTTABA2	EBCDIC 文字／ASCII 文字の変換用テーブル。これらのテーブルはオブジェクトハンドラによって使用されます。
NTTABL	SYS* 変換テーブル。Natural SYS... ライブラリ内のプログラムからの出力を変換します。
NTLANG *	言語変換テーブル。Natural に定義されているすべての使用可能言語コードのリストを含みます。
NTSCTAB	スキャナ文字タイプテーブル。小文字、大文字、数値、特殊文字を判別します（ダイナミックプロファイルパラメータ MASK および SCAN オプションに適用）。
NTTZ	タイムゾーン定義。NTTZマクロにより、ゾーンタイムの指定と夏時間／冬時間の自動切り替えが可能になります。
NTBUFID	このマクロのパラメータ MIN および MAX を使用して、可変バッファのバッファサイズ制限を変更できます。「 バッファ特性のカスタマイズ 」を参照してください。 重要: このマクロの他のパラメータのデフォルト値は変更しないでください。予期しない結果になることがあります。

* このリストでアスタリスク (*) の付いたテーブルについては、変更する場合は事前に Software AG サポートに問い合わせてください。

詳細については、「[変換テーブル](#)」を参照してください。

NTDVCE - 端末デバイス指定テーブル

Natural でサポートされる各端末タイプに、端末コンバータルーチンが用意されています。対応する端末ドライバが、実際の端末 I/O を引き受けます。ドライバは、画面バッファや画面属性バッファから物理的なデータストリームを作成し、端末 I/O バッファに配置します。

また、より高速なテレックス、テレファックス、およびテレテックス通信を TOPCALL システムと行うために、Con-nect にテレックスドライバモジュール NATTLX が用意されています。NATTLX は TOPCALL フルページプロトコルをサポートします。

NTDVCE マクロを使用すると、新しい端末ドライバを Natural に追加し、端末固有の入力/出力または小文字/大文字変換テーブルの変更を指定できます。指定できる他の情報には、フレーム文字、メッセージ行の位置、画面最適化のオン/オフ、IOCB の各種フラグがあります。また、端末指定を、他の変換テーブルを使用して既存のドライバにルーティングしたり、ドライバルーチンにフックしたりできます。

NTDVCE マクロは、Natural コマンド行からの端末コマンド %T= または Natural プログラム内の SET CONTROL 'T=...' ステートメントのどちらかで呼び出されます。Natural セッションの開始時に、変換テーブル NNTAB、NNTAB1、NNTAB2、NTUTAB1、および NTUTAB2 が、NATCONFIG モジュールからユーザーエリアにコピーされ、そこで NTDVCE によって変更されます。

変換テーブルは、同じマクロでダイナミックに、または NATPARAM パラメータモジュール内で変更できることに注意してください。

NTSTAT - Natural ニュークリアスにリンクされる Natural オブジェクトの定義

Natural ニュークリアスにリンクされるオブジェクトは、NTSTAT マクロを使用して指定する必要があります。オブジェクトを検索する場合、指定したライブラリに関係なく、Natural では常にこのリストが最初にスキャンされます。Natural オブジェクトの Natural ニュークリアスへのリンク方法については、「[Natural ニュークリアスへの Natural オブジェクトのリンク](#)」の ULDOBJ コーティリティを参照してください。

パラメータ	説明
ALIAS	コードページの別名。最大長は 32 文字です。このパラメータは任意指定です。一意ではありません。
PHC	プレースホルダ文字。長さ 2 バイトの 16 進数です。このパラメータは任意指定です。

 **注意:** パラメータ CCSID および CCSN は、プラットフォーム固有 (IBM/SNI) で相互排他的です。

例:

```
NTCPAGE IANA=IBM819,CCSID=819,ALIAS='ISO-8859-1',PHC=003F
```

「Unicode／コードページ環境の設定と管理」も参照してください。

コードページのサポート

NTDVCE マクロを使用して、さまざまなコードページを定義し、特定の端末タイプおよび名前に関連付けることができます。その後 **Natural** を `PM=C` で起動すると、すべての端末 I/O が入力時に変換され、出力時に再変換されます。このため、コードページに互換性がある限り、共通データの表現が維持されます。

『ユーティリティ』ドキュメントの「SYSCP ユーティリティ - コードページ管理」も参照してください。

サポートされる出力デバイス

属性制御変数とフォーマットにより、出力デバイスでの特定の表現を生成する属性を定義します。**Natural** は、エンドユーザーがマップやレポートを端末で最適に設計できるように、可能性のある多種多様な属性を提供します。

ただし、**Natural** で使用できるすべての機能をすべての端末がサポートしているわけではありません。このようなデバイスでは、これらの機能はほとんどの場合無視されるか、別の技術を使用してシミュレートされます。基本的に、**IBM** 環境には標準データストリームと拡張データストリームと呼ばれる2つのデータストリーム定義があり、**SNI** 環境には多数のデータストリーム定義があります。

以下の出力デバイスがサポートされています。

- バッチおよび追加レポート用のシーケンシャル出力デバイス
- 行指向オンライン端末

■ ブロックモード指向オンライン端末

バッチおよび追加レポート用のシーケンシャル出力デバイス

出力データには、指定されたプリンタの行送りとページ換えの機能を制御する標準 ASA 制御文字が含まれます。このプリンタは、オンラインまたはバッチスプーリングシステムによってサポートされるコンピュータセンターの中央プリンタか、オンライン端末プリンタとして使用される SCS プリンタです。

この形式で生成されるレポートの印刷には、以下のデバイスを使用できます。

デバイス	タイプ
インパクトプリンタ	標準中央プリンタハードウェア
レーザープリンタ	高速プリンタ、端末プリンタ
デジプリンタ	端末プリンタ
インクジェット	端末プリンタ

行指向オンライン端末

端末の型	説明
TTY	TTY デバイスに送信されるデータは、標準の改ページ文字や改行文字を使用して生成されません。

ブロックモード指向オンライン端末

端末の型	説明
IBM	標準データストリームと拡張データストリームのいずれかまたは両方をサポートするすべてのモデルおよびサイズ
SNI	すべての 9750 および互換モノクロデバイスと、すべての 9763 および互換カラーデバイス
Wang	すべてのモデル
PC	標準データストリームと拡張データストリームのいずれかまたは両方をサポートするすべてのモデルおよびサイズ

NTDVCE の指定

NTDVCE マクロの指定方法と各パラメータの詳細については、NTDVCE マクロ自体を参照してください。

NTDVCE マクロの例：

```
NTDVCE TYP=EBS2,NAME=BS2CHAR,ENTRY=VC3270,WXTRN=OFF,RTAL=5,  
FLAG1=CM3270,TCIO=(X'CO',X'FB',X'6A',X'4F',X'DO',X'FD',  
X'4A',X'BB',X'E0',X'BC',X'5A',X'BD',X'A1',X'FF',X'4F',  
X'5A')
```

このサンプルマクロでは、内部SNIコードページを外部IBMコードページに変換します。これにより、内部的にはSNIコードページで動作するアプリケーションをIBM端末上で開発して、IBMからSNIに移行する際のデータ競合などを回避することができます。

変換テーブル

Natural プログラムによって出力、表示、作成されるすべてのデータは、Natural によって変換されます。これにより、無効な制御文字が原因で端末 I/O エラーが発生したり不要な情報が端末に表示されたりするのを確実に防ぐことができます。

また、ラテン語定義と異なる文字セット（特に、アラビア語、キリル語、ギリシャ語、ヘブライ語）との相互変換機能もあります。

このセクションでは、データがCRT（画面端末）やオンラインおよびバッチスプーリングシステムなどの外部デバイスに書き込まれるときのフィールド変換に関するすべての機能について説明します。

ステートメント INPUT、DISPLAY、PRINT、および WRITE は、CRT、TTY、シーケンシャルファイルなどの外部デバイスに対してデータの読み書きを行います。これらのステートメントはすべて、定数、変数、編集マスク、属性制御変数、フォーマットなどのパラメータを使用して、出力イメージと入力表現を制御します。定数と変数は、出力イメージのそれぞれの値を使用して生成されます。これらの値の表現は、属性制御変数、フォーマット、編集マスク、および変換テーブルによって制御されます。

Natural では、複数の変換テーブルを使用し、代替変換テーブルも使用できます。これらはすべて NATCONFIG に含まれています。

以下のテーブルが用意されています。

マクロ	テーブル
NATSCTU	<p>Unicode 文字に必須のスキナテーブル。Natural ニュークリアスによる使用のために、(提供される ICU バージョンでサポートされる) Unicode 仕様の Unicode 文字のプロパティをマップします。</p> <p>重要: このテーブルは変更しないでください。</p>
NATCPTAB	<p>シングルバイトコードページの変換アクセラレータテーブル (オプション)。</p> <p>テーブルが存在する場合は、コードページ間の変換は ICU 機能呼び出す代わりにこのテーブルを使用して行われるのでより高速に実行されます。</p> <p>提供される NATCPTAB では、以下のコードページがサポートされます。</p> <p>IBM01140 IBM01141 IBM01145 IBM01146 IBM01147 ASCII</p> <p>NTPCNCV マクロを使用して、新しいエントリを追加できます。変換方向ごとに、ソースコードページの IANA 名、ターゲットコードページの IANA 名、およびオプションで空白文字、置換文字、プレースホルダ文字、その後に文字マッピングリストを含むエントリが必要です。</p>
NTSCTAB	<p>Natural マスク定義機能の各出力可能文字のプロパティを定義する SCAN/MASK 文字テーブル。</p> <p>このテーブルは、大文字属性、小文字属性、特殊文字、16 進文字、および数字の定義に使用できます。</p> <p>ユーザーによる変更が可能で、結果を直接 Natural MASK 節で使用できます。</p> <p>このテーブルを変更するには、Natural パラメータモジュールの NTSCTAB マクロ、または対応するダイナミックプロファイルパラメータ SCTAB を使用します。</p> <p>コードページがプロファイルパラメータ CP (CP=ON、CP=AUTO、または CP=<i>code-page</i>) を使用して指定されている場合は変更は無視され、テーブルはセッションの開始時に使用されるコードページに従って ICU によって調整されます。</p>
NTTAB	<p>画面またはプリンタへの出力に使用される標準 (プライマリ) 出力変換テーブル。</p> <p>基本的に、このテーブルは X'40' 以下のスペース文字から疑問符までのすべての文字の変換に使用されます (X'00' は変換されません)。これにより、確実に、すべての端末制御文字が出力前に変換され、制御エスケープシーケンスが画面出力に影響を及ぼしません。画面出力に影響を及ぼす可能性がある特殊文字 (X'FE' および X'FF') は、疑問符に変換されます。</p> <p>他に指定されていない限り、すべての Natural 出力データは NTTAB を使用して変換されます。</p> <p>このテーブルを変更するには、Natural パラメータモジュールの NTTAB マクロ、または対応するダイナミックプロファイルパラメータ TAB を使用します。</p>

マクロ	テーブル
	<p>コードページがプロファイルパラメータ CP (CP=ON、CP=AUTO、または CP=<i>code-page</i>) を使用して指定されている場合は変更は無視され、テーブルはセッションの開始時に使用されるコードページに従って ICU によって調整されます。その後、Natural はコードページを使用して実行されていますが、結果の出力データからの不正な出力不能文字を回避するために、このテーブルを使用した変換は継続されます。</p>
NTTAB1	<p>Natural パラメータ PM が C に設定されている場合に使用されるセカンダリ文字セットの代替 (セカンダリ) 出力変換テーブル。</p> <p>重要な点は、可能なすべての端末制御文字の変換です。PM=C を指定している場合は、すべての Natural 出力データは NNTAB1 を使用して変換されます。NTTAB1 の可能なアプリケーションは、プリンタ制御用のエスケープシーケンスの変換を回避します。</p> <p>このテーブルを変更するには、Natural パラメータモジュールの NNTAB1 マクロ、または対応するダイナミックプロファイルパラメータ TAB1 を使用します。</p> <p>コードページがプロファイルパラメータ CP (CP=ON、CP=AUTO、または CP=<i>code-page</i>) を使用して指定されている場合は変更は無視され、テーブルは使用されません。</p>
NTTAB2	<p>Natural パラメータ PM が "C" に設定されている場合に使用されるセカンダリ入力変換テーブル。PM=C を指定している場合は、すべての Natural 入力データは NNTAB2 を使用して変換されます。異なる言語またはコードページ間の変換は、このテーブルと NNTAB1 を使用して実行できます。</p> <p>このテーブルを変更するには、Natural パラメータモジュールの NNTAB2 マクロ、または対応するダイナミックプロファイルパラメータ TAB2 を使用します。</p> <p>コードページがプロファイルパラメータ CP (CP=ON、CP=AUTO、または CP=<i>code-page</i>) を使用して指定されている場合は変更は無視され、テーブルは使用されません。</p>
NTTABS	<p>このテーブルは、Natural 変数名に使用できるすべての有効な文字を定義し、Natural 構文プロセッサで使用されます。</p> <p>また、Natural 変数名の先頭位置に使用できるすべての有効な文字も定義します。</p> <p>さらに、変数がグローバル変数、非データベース変数、またはソースコード変数であるかどうかも定義します。</p> <p>コードページがプロファイルパラメータ CP (CP=ON、CP=AUTO、または CP=<i>code-page</i>) を使用して指定されている場合は、テーブルはセッションの開始時に使用されるコードページに従って ICU によって調整されます。</p>
NTUTAB1	<p>小文字から大文字への入力変換用のサンプルユーザー固有変換テーブル。</p> <p>このテーブルでは、ステートメント EXAMINE TRANSLATE INTO UPPER CASE で指定された変換も実行します。</p> <p>このテーブルを変更するには、Natural パラメータモジュールの NTUTAB1 マクロ、または対応するダイナミックプロファイルパラメータ UTAB1 を使用します。</p> <p>コードページがプロファイルパラメータ CP (CP=ON、CP=AUTO、または CP=<i>code-page</i>) を使用して指定されている場合は変更は無視され、テーブルは使用されません。</p>

マクロ	テーブル
NTUTAB2	<p>ステートメント EXAMINE TRANSLATEINTO LOWER CASE で指定された変換を実行するサンプルユーザー固有変換テーブル。</p> <p>このテーブルを変更するには、Natural パラメータモジュールの NTUTAB2 マクロ、または対応するプロファイルパラメータ UTAB2 を使用します。</p> <p>コードページがプロファイルパラメータ CP (CP=ON、CP=AUTO、または CP=code-page) を使用して指定されている場合は変更は無視され、テーブルは使用されません。</p>
NTLANG	<p>システム変数 *LANGUAGE の言語コードへの言語番号の割り当てを定義する言語コードテーブル。</p>
NTTABL	<p>Natural プロファイルパラメータ TS で制御される SYS* 出力変換テーブル。TS=ON の場合は、このテーブルを使用して、Natural SYS* ライブラリにあるプログラムで生成される出力をラテン文字の小文字から大文字に変換します（変更可能フィールドを除く）。</p> <p>このテーブルを使用すると、ラテン語系の国ですべての大文字と小文字を使用できるようになります。ただし、小文字が固有のアルファベットで置き換えられている国でも、これらのアプリケーションを使用できます。</p> <p>このテーブルを変更するには、Natural パラメータモジュールの NNTTABL マクロ、または対応するダイナミックプロファイルパラメータ TABL を使用します。</p> <p>Natural が MBCS コードページ (CP='IBM-939' など) を使用して実行されている場合、テーブルは使用されませんが、変換は現在のロケール設定に従って ICU によって実行されます。</p>
WRDFCUC1 WRDFCUC2 WRDFCSP2	<p>ダブルバイト文字とラテン文字の相互変換に使用される DBCS 変換テーブル。</p> <p>重要: これらのテーブルは、極東諸国などで使用するよう明示的にアクティブにする必要があります。</p>

大文字／小文字変換

変更可能フィールドおよび入力フィールドには、大文字／小文字変換を指定できます。通常、小文字変換とは、データが入力されたとおりに取得され、変換は行われなことを意味します。これにより、例えば、バッチモードで 16 進数データを変換せずに読み取ることができます。

大文字／小文字変換を指定するには、以下のような方法があります。

LC=OFF	<p>小文字変換はオフに切り替えられ、大文字変換がグローバルに有効になります。</p> <p>このプロファイルパラメータは、Natural パラメータモジュールで指定することも、ダイナミックパラメータとして指定することもできます（セッションパラメータ LC の機能はまったく異なります）。</p>
--------	--

%U	<p>大文字変換がグローバルにオンになります。</p> <p>フィールドレベルでは、属性 AD=T または AD=W を指定できます。これらの属性は、グローバルな大文字変換が非アクティブな場合 (LC=ON,%L) にのみ有効になります。Natural プログラムからフィールドレベルで変換を制御できます。</p>
EXAMINE TRANSLATE	<p>大文字／小文字変換は、EXAMINE TRANSLATE ステートメントでも実行できます。</p> <p>デフォルトでは、EXAMINE TRANSLATE は、変換テーブル NTUTAB1 を使用して大文字に、変換テーブル NTUTAB2 を使用して小文字に変換します。</p>

CMULT エントリ

CMULT エントリの使用はお勧めできません。代わりに EXAMINE TRANSLATE ステートメント（上記参照）を使用してください。

出力変換

すべてのフィールド（可能な編集マスクによってフォーマットされた後）、AL パラメータ値、NL パラメータ値、充填文字などは、変換テーブルを使用して変換されます。これにより、確実に、Natural によって明示的に生成されていない埋め込み制御情報でデータがフロントエンド出力デバイスに送信されることがなくなります。これは、内部属性と同じ 16 進数情報が含まれている場合でも、フィールドを表示デバイスに送信できることを意味します。これらの属性は出力操作の前に変換されるため、確実に画面レイアウトは出力ステートメントで定義されたとおりになります。

使用可能な変換テーブルは複数あります。変換テーブルが明示的に定義されていない場合は、プライマリ変換テーブル NTTAB が使用されます。

PM=C が指定されている場合は、セカンダリ変換テーブル NTTAB1 が使用されます。変更可能フィールドでは、PM=C は受信データが再変換されることも意味します。つまり、出力用に交換され、入力用に再変換されます。

この変換テーブルロジックにより、例えば、アラビア数字をラテン数字に変換することが可能です。アラビア数字は、端末ハードウェアでの 16 進数表現が通常のアラビア数字と異なります。そのため、出力時にはラテン数字が等価のアラビア数字に変換され、入力時にはアラビア数字が等価のアラビア数字に変換されます。

ラテン文字の小文字と大文字を使用する Natural システムアプリケーションについては特別な考慮が必要です。特に、アラビア語、ギリシャ語、キリル語などをサポートする端末では、小文字のラテン文字ではなく固有の文字が表示されるようにハードウェアを切り替えられます。

ただし、ラテン文字の小文字は、キリル語などで表示された場合は判読しづらくなります。そのため、Natural をパラメータ TS=ON（システム出力変換）を指定して使用します。TS=ON を指

定すると、「SYS*」ライブラリ（SYSTEMライブラリを除く）とすべてのNaturalシステムコマンドが3番目の変換テーブルNTTABLを使用して変換されます。デフォルトでは、この変換テーブルによって、すべての小文字のラテン文字が大文字に変換されます。このように処理されるのは出力データのみです。したがって、Naturalエディタやシステムアプリケーションでも固有の文字セットでデータ入力できます。

ただし、固有の文字セットで入力されたデータをNaturalユーティリティを使用して表示する場合は、例えばキリル語表現のデータでも大文字変換されます。結果は再度読み取り不可になります。そのため、すべてのNaturalシステムユーティリティでは、固有の文字セットで入力されたデータを含むフィールドにフォーマットPM=Cを使用できます。この場合は、NTTABL変換テーブルもセカンダリ変換テーブルNTTAB1も使用されません。データはプライマリ変換テーブルNTTABで変換されるだけです。

詳細については、『パラメータリファレンス』ドキュメントでプロファイルパラメータPMとTSを参照してください。

入力変換

変換テーブルNTUTAB1を使用して、小文字から大文字への変換を制御できます。この使用が原因で、この文字のステータスを1ビットのみで制御する単純なロジックで設定されない特殊文字を使用している国では問題が発生する場合があります。これは特に、ドイツ語のウムラウト文字やデンマーク語の特殊文字に当てはまります。そのような場合、変換を正常に行うには、NTUTAB1テーブルをカスタマイズして、各文字に対応する小文字／大文字を指定します。

大文字変換(%U)とPM=Cを指定している場合は、最初に大文字変換(NTUTAB1を使用)、次にセカンダリ入力変換(NTTAB2を使用)が実行されます。

DBCS データのコード変換

ダブルバイト文字セット(DBCS)データを処理できるように、ダブルバイト文字をラテン文字に変換するユーザーアプリケーションプログラミングインターフェイスUSR4213Nが提供されています。「[ダブルバイト文字セット](#)」を参照してください。

NTTZ - タイムゾーン定義

以下では次のトピックについて説明します。

- [NTTZ マクロ](#)
- [NTTZ マクロ構文](#)
- [NTTZ マクロパラメータ](#)
- [NTTZ マクロの制限](#)
- [NTTZ マクロの例](#)

NTTZ マクロ

NTTZ マクロにより、ゾーンタイムの指定と夏時間／冬時間の自動切り替えが可能になります。

タイム定義はシステム管理者が決定し、ユーザーは Natural プロファイルパラメータ `TD=zonename` を使用してこれらの定義を参照できます。このパラメータを使用することで、タイムゾーンが異なる各国のユーザーがそれぞれのローカルタイムを選択することができます。

最小限、NTTZ マクロを使用して、タイムゾーンの時差を定義できます。また、夏時間／冬時間の自動切り替えを、決まった日付で、または「4月の最初の日曜日」のようにより柔軟な方法で指定できます。夏時間／冬時間の自動切り替えは Natural セッションの実行中に処理され、ユーザーの操作は必要ありません。NTTZ マクロ定義の定義済みサンプルが NATCONFIG に付属しています。

夏時間／冬時間の自動切り替えの参照ポイントは現在のマシンタイムで、UTC (GMT) タイムです。現在のマシンタイムに基づいて現在のローカルタイムが決まります。夏時間／冬時間の自動切り替えは 2002 年から 2041 年までです。



注意:

1. 夏時間／冬時間の自動切り替えは現在のマシンタイムに基づいて行われるため、Natural プロファイルパラメータ `DD` および `YD` は切り替えに影響しません。`DD` または `YD` を `TD=zonename` と同時に使用しないでください。
2. `TD=zonename` とユーザー出口 `CMCOTIME` (マシンタイムの上書き) の同時使用はお勧めできません。マシンタイム (TOD クロック) を変更すると、`TD=zonename` で呼び出された自動切り替えが予期しない結果になることがあります。

NTTZ マクロ構文

NTTZ マクロの構文は以下のとおりです。

```
NTTZ ZONE=time-zone-name, TDON=+/-hh:mm:ss,
      [TDOFF=+/-hh:mm:ss, SWTON=hh:mm:ss,
      SWTOFF=hh:mm:ss,
      DSTON=( [{FIRST | SECOND | THIRD | FOURTH | LAST},
               {MONDAY | ... | SUNDAY},
               {AFTER | BEFORE | IN}],
               {JANUARY | ... | DECEMBER},
               [, day-number]),
      DSTOFF=( [{FIRST | SECOND | THIRD | FOURTH | LAST},
                {MONDAY | ... | SUNDAY},
                {AFTER | BEFORE | IN}],
                {JANUARY | ... | DECEMBER}
                [, day-number]) ] ]
```

NTTZ マクロパラメータ

NTTZ マクロパラメータ	説明
< <i>time +/- hh:mm:ss</i> >	基本フォーマットは<{+/-} hh:mm:ss>で、範囲は00:00:00~23:59:59です。省略形も使用でき、<hh:mm>または単に<hh>と指定します。プラス記号(+)はデフォルトで想定され、マイナス記号(-)はパラメータ TDON または TDOFF で必要な場合があります(下記参照)。
<i>time-zone-name</i>	Software AG またはユーザーによって定義されるタイムゾーン名で、TD パラメータで参照できます。最初に検出された名前が選択されます。タイムゾーン名の最大長は32文字で、首都名などの適切なユーザー定義名を指定できます。
TDON	ローカルの夏時間と UTC タイム(旧 GMT) との差を表します。このパラメータはパラメータ SWTON に対応します。 TDON パラメータのみを定義している場合は、ユーザーにはローカルタイムがユーザーのゾーンタイムで表示され、夏時間/冬時間の自動切り替えはありません。
TDOFF	ローカルのゾーンタイムと UTC タイム(旧 GMT) との差を表します。このパラメータはパラメータ SWTOFF に対応します。
SWTON	夏時間をオンにする UTC タイムポイントを表します。
SWTOFF	夏時間をオフにする UTC タイムポイントを表します。
<i>day-number</i>	各月の有効な日にち。 <i>day-number</i> のデフォルト値は1です。

NTTZ マクロの制限

- LAST には BEFORE または IN が必要です。
- IN を指定しない場合は、*day number* を指定する必要はありません。



注意: タイム切り替えのための一意の参照ポイントは NNTZ マクロパラメータ SWTON および SWTOFF で UTC タイムで指定し、曜日名と日にちは NNTZ マクロパラメータ DSTON および DSTOFF でローカルタイムで指定します。

NTTZ マクロの例

西ヨーロッパの夏時間切り替え：

```
NTTZ ZONE=MEZ,  
    TDON=2, TDOFF=(+01,00,00), SWTON=(01,00,00), SWTOFF=(01,00,00),  
    DSTON=(LAST,SUNDAY,IN,MARCH),  
    DSTOFF=(LAST,SUNDAY,IN,OCTOBER)
```

他のタイムゾーン（北米、南米、アジアなど）の例は、Software AG 提供の [NATCONFIG](#) にあります。

9 Natural ストレージ管理

▪ スレッドおよび非スレッド環境	54
▪ バッファタイプ	54
▪ 固定バッファ	55
▪ 可変バッファ	55
▪ バッファ特性のカスタマイズ	56

このドキュメントでは、Naturalでメインストレージを割り当て使用方法について説明します。Naturalニュークリアスコンポーネントによって要求されるストレージ領域は「バッファ」と呼ばれます。

以下のトピックについて説明します。

スレッドおよび非スレッド環境

ストレージ環境には、以下の2つのタイプがあります。

- スレッドストレージ環境（代表的なマルチユーザー環境、例えば CICS）
- 非スレッドストレージ環境（代表的なシングルユーザー環境、例えば バッチ）

スレッド環境では、「スレッド」と呼ばれるストレージの大きい領域が、あらかじめセッションに割り当てられます。スレッドサイズは、システム管理者が事前に定義する必要があります。セッション中に、各バッファ割り当て要求 (getmain) は、Natural 自体によってスレッド内で満たされます。バッファ要求の解放 (freemain) によるフリースペースは再使用できます。

特定のイベント（端末I/Oや長い待機など）の発生時に、スレッドストレージを圧縮し、外部ストレージ（スワッププールやロールファイル）にロールアウトすることができます。解放されたスレッドは、別の Natural セッションで再使用できます。中断されたセッションが再開されるときには、セッションが外部ストレージからフリースレッドに再度ロールインされます。

圧縮されたスレッドストレージが保存されるスワッププールまたはロールファイル上の場所は、「スロット」と呼ばれます。スロットサイズは固定長で、システム管理者が定義します。スロットには、最も大きい圧縮スレッドストレージを収容できる大きさが必要です。最悪の場合は、スレッドサイズと同じになります。

非スレッド環境では、すべてのストレージ要求がオペレーティング（サブ）システムに直接渡されます。ロールアウト／ロールインは実行されません。つまり、セッション用のバッファは、明示的に解放されない限り、セッション終了まで保持されます。

バッファタイプ

バッファには、以下の3つのタイプがあります。

- 固定バッファ
- 可変バッファ
- 物理バッファ

固定バッファと可変バッファには、すべての環境に共通なレイアウトの 32 バイトの接頭辞が付きます。バッファの接頭辞は、バッファ名で始まり、その後5つのバッファ長フィールド

(total、used low-end、max. used、used high-end、max. used high-end) が続きます。使用された長さのフィールドは、バッファを所有しているコンポーネントによって保持され、スレッド圧縮に使用されます。各バッファは、一意の ID 番号 (1~255) を持ち、1 回だけ存在可能です。一部のバッファはセッションの初期化時に割り当てられ、その他のバッファは要求があったときに割り当てられます。システムコマンド `BUS` を使用して、現在割り当てられているすべての固定および可変バッファに関する情報を表示できます。バッファの特性は、ソースモジュール `NATCONFIG` で定義され、特別な場合のみカスタマイズできます (後述の「[バッファ特性のカスタマイズ](#)」を参照)。一部のバッファのサイズはプロファイルパラメータで指定できます。このようなバッファの完全なリストについては、プロファイルパラメータ `DS` を参照してください。

物理バッファは、スレッド外に割り当てられます。物理バッファは、バッファ接頭辞を持たず、一意でもありません。これらは、特別な場合に一時的に使用されます。物理バッファは、次の端末 I/O 時に自動的に解放されます。プロファイルパラメータ `WPSIZE` を使用して、物理バッファ用のワークプールを定義できます。

固定バッファ

スレッド環境では、固定バッファはスレッドのローエンドからのみ割り当てられます。可変バッファとは対照的に、固定バッファはスレッドと相対的に移動することはできず、サイズの増減もできません。

可変バッファ

スレッド環境では、可変バッファはスレッドのハイエンドからのみ割り当てられます。スレッドに余分のスペースがなくなると、可変バッファは一時的にスレッド外に割り当てられます。スレッドの圧縮時に、使用されたすべてのバッファ領域はスレッドに圧縮されます。スレッド内に収容できない場合は、セッションが異常終了します。このような事態は、特に、大きいダイナミック変数を使用している場合に発生します。

スレッドの圧縮後、可変バッファがスレッド内またはスレッド外の別の場所に移動されていることがあります。可変バッファのサイズは、要求時に所有コンポーネントによって増減できます。一部の可変バッファは、スレッド圧縮時に自動的に縮小または解放されるように定義できます。

スレッド外に割り当てられるストレージの総容量を、プロファイルパラメータ `OVSIZE` で制限できます。

バッファ特性のカスタマイズ

すべてのバッファは、ソースモジュール `NATCONFIG` に `NTBUFID` マクロ定義によって定義されています。

注意: 以下で説明する最小および最大バッファサイズ以外のバッファ特性は変更しないでください。予期しない結果になることがあります。

バッファサイズ制限は、`NTBUFID` マクロのパラメータ `MIN` および `MAX` で変更できます。これは、可変バッファ (`TYPE=VAR`) の場合のみです。すべてのバッファの制限は、デフォルト (0~2097151KB) または対応するプロファイルパラメータの制限で定義されます。詳細については、プロファイルパラメータ `DS` を参照してください。Natural パラメータモジュール (`NATPARM`) のバッファサイズプロファイルパラメータの制限は、`NTBUFID` のパラメータ `MIN` および `MAX` の影響を受けませんが、ダイナミックプロファイルバッファサイズパラメータの制限は `MIN` および `MAX` によって上書きされます。

`MAX` パラメータに値 (KB 単位) を設定すると、セッション実行中に、このバッファのサイズは設定した値を超えることができなくなります。これにより、このバッファ以上のバッファストレージが要求されると、ランタイムエラーが発生することがあります。

`MIN` パラメータに値 (KB 単位) を設定すると、セッション実行中に、このバッファのサイズは設定した値を下回ることができなくなります。例えば、`3GL CALLNAT` インターフェイス (`NAT3GCAN`) の場合は、バッファの最小値の設定は、以下のバッファに有効です。これらのバッファのサイズは、`3GL` プログラムによって呼び出される下位レベルの Natural プログラムでは大きくできないからです。

<code>DATSIZE</code>	データエリア
<code>GLBT00L</code>	ユーティリティ GDA
<code>GLBUSER</code>	ユーザー GDA
<code>GLBSYS</code>	システム GDA
<code>AIVDAT</code>	AIV エリア
<code>CONTEXT</code>	コンテキスト変数

バッファ特性定義の例：

```
DATSIZE NTBUFID ID=GETMDATA,TYPE=VAR+INI,MAX=512
```

バッファサイズに影響するプロファイルパラメータの詳細については、「[バッファサイズ](#)」を参照してください。

10 プロファイルパラメータの使用方法

このセクションでは、メインフレーム環境で Natural プロファイルパラメータを使用する際の基本とルールについて説明します。

- **Natural** パラメータ階層 Natural パラメータを設定できるさまざまなレベルの階層構造について概要を説明します。各種の例が用意されています。
- パラメータ値の割り当て プロファイルパラメータに値をスタティックに、ダイナミックに、およびランタイム時に割り当てる方法について説明します。
- プロファイルパラメータの概要 使用可能なプロファイルパラメータについて機能別に概要を説明します。
- **Natural** パラメータモジュールの使用 Natural パラメータモジュールのアセンブル、NATPARM デフォルト Natural パラメータモジュールの使用、NTPRM マクロの作成、および Natural パラメータモジュールで使用されるオプションのマクロについて説明します。

各プロファイルパラメータの詳細については、『プロファイルパラメータ』ドキュメントを参照してください。

11 Natural パラメータ階層

▪ Natural パラメータ階層の概要	62
▪ パラメータ使用の一般ルール	62
▪ Natural 標準パラメータモジュール	63
▪ 代替パラメータモジュール	63
▪ 事前定義されたダイナミックパラメータ設定	64
▪ 事前定義されたユーザーパラメータプロファイル	64
▪ ダイナミックパラメータ入力	64
▪ Natural Security 定義	65
▪ セッションパラメータ設定	65
▪ プログラム／ステートメントレベル設定	65
▪ 開発環境設定	66
▪ さまざまなパラメータ文字列の例	66

このドキュメントでは、Natural プロファイルパラメータを設定できるさまざまなレベルの階層構造について説明します。各種の例が用意されています。

以下のトピックについて説明します。

各プロファイルパラメータの詳細については、『パラメータリファレンス』ドキュメントを参照してください。

Natural パラメータ階層の概要

Natural プロファイルパラメータは、Natural ユーザーの操作環境の外観とレスポンスに影響します。これらのパラメータは、下の表に示されているように、異なる階層構造レベルで設定されます（優先度は上の方が高くなります）。

レベル	簡単な説明／詳細な説明の参照
セッション中	<ul style="list-style-type: none">■ 開発環境設定■ プログラム／ステートメントレベル設定■ セッションパラメータ設定■ Natural Security 定義
セッション開始時にダイナミック	<ul style="list-style-type: none">■ ダイナミックパラメータ入力■ 事前定義されたユーザーパラメータプロファイル■ 事前定義されたダイナミックパラメータ設定
スタティック	<ul style="list-style-type: none">■ 代替パラメータモジュール■ Natural 標準パラメータモジュール

階層構造の各レベルについて、優先度の低い順に参照先セクションで説明します。

パラメータ使用の一般ルール

以下の一般ルールが適用されます。

- 下のレベルで定義されたパラメータ値は、上のレベルで設定された値によって上書きされます（例外：PROFILE、SYS、DYNPARM、および値を追加することで機能する一部のパラメータ）。
- セッション開始時のダイナミックパラメータには優先順位があり、左から右に評価されます。

例：

```
ESIZE=20,DATSIZE=60,ESIZE=100
```

結果値は ESIZE=100 です。

- 下のレベルでは使用可能でも上のレベルでは定義できないパラメータもあります。

Natural 標準パラメータモジュール

Natural パラメータは、Natural ニュークリアスにリンクする標準（デフォルト）パラメータモジュールで定義されます。このモジュールは、Natural パラメータ階層の最下位レベルを構成します。

特殊なケース：

Natural 共有ニュークリアスを使用する場合は、このパラメータモジュールを環境依存ニュークリアスモジュールにリンクする必要があります。これで、このパラメータモジュールは、2 番目の階層レベルを構成し、共有ニュークリアスにリンクするパラメータモジュールのすべてのパラメータを上書きします（存在する場合）。例外：共有ニュークリアスの CSTATIC サブプログラム。「[スタティックにリンクされた Natural 以外のプログラム](#)」を参照してください。

代替パラメータモジュール

Natural 管理者は、Natural 標準パラメータモジュールに加えて、任意の数の追加（代替）パラメータモジュールを定義できます。このようなモジュールは TP またはオペレーティングシステムライブラリに保存され、Natural の起動時にパラメータ PARM によって代替パラメータモジュールとして使用できます。

これらのパラメータにより、標準パラメータモジュールのパラメータは完全に上書きされます。

例外：CSTATIC エントリ。「[スタティックにリンクされた Natural 以外のプログラム](#)」を参照してください。

-  **重要:** PARM はダイナミックパラメータ文字列の最初のパラメータである必要があります。そうでない場合は、ダイナミックパラメータ文字列に入力済みのすべてのパラメータ設定が代替パラメータモジュールによって上書きされます。

マクロ NTUSER を使用して、代替パラメータモジュールの使用を特定のユーザーまたは複数のユーザーに制限できます。

事前定義されたダイナミックパラメータ設定

アセンブラマクロ NTSYS を使用して、Natural パラメータモジュールに指定するパラメータセットを事前定義できます。これらのセットは、対応するパラメータモジュールがアクティブであれば、Natural の呼び出し時にその名前指定できます。

事前定義されたパラメータセットは、呼び出されると、その位置にダイナミックに入力されたパラメータと同様に動作します。

プロファイルパラメータ SYS も参照してください。

事前定義されたユーザーパラメータプロファイル

Natural ユーティリティ SYSPARM を使用して、システムファイルに保存される個別のプロファイルを作成できます。各プロファイルには一意の文字名が付きます。このプロファイル内の任意のダイナミック Natural パラメータに値を設定できます。

ユーティリティ SYSPARM で作成したプロファイルは、Natural の呼び出し時にパラメータ PROFILE を使用してアクティブにします。

プロファイルパラメータ USER を使用して、プロファイルの使用を特定のユーザーまたは複数のユーザーに制限できます。

事前定義されたパラメータプロファイルは、呼び出されると、その位置にダイナミックに入力されたパラメータと同様に動作します。

ダイナミックパラメータ入力

ほとんどすべてのパラメータは、Natural の起動時にダイナミックに上書きできます。ダイナミックパラメータは、厳密に順次に評価されます。

ただし、この一般的な上書き機能は、プロファイルパラメータ DYNPARM を使用して（プロファイル内など、ダイナミックのみ）、全般的にまたは特定のパラメータに制限できます。

パラメータモジュール NATPARM のマクロ NTDYNP を使用して、アナログ設定を行うことができます。ただし、これにより、プロファイルパラメータ DYNPARM は使用できなくなります。

ファイル CMPRMIN を使用して、z/OS、BS2000/OSD、および z/VSE 環境でバッチモードで、または TSO、TIAM、CMS や IMS/TM 上の BMP などのバッチ的なシステムで、ダイナミックパラメータを定義できます。

このメソッドの長所は、Natural 設定を変更する場合に JCL を修正する必要がないことです。また、パラメータ文字列の長さ制限（z/OS で 100 文字など）を超えることができます。

Natural Security 定義

ライブラリ、ファイル、およびコマンドの保護とは別に、Natural Security を使用して特定のセッション関連のプロファイルパラメータを設定できます。定義はユーザーの現在のライブラリに適用されます。

ユーザーはプライベートまたはデフォルトのライブラリの設定も定義できます。

現在のセキュリティ設定（セッションパラメータ）を表示するには、Natural システムコマンド PROFILE を使用します。

Natural Security パラメータ定義は通常のプロファイルパラメータの後に評価されるので、通常のプロファイルパラメータを上書きできます。

セッションパラメータ設定

Natural システムコマンド GLOBALS を使用して、またはレポーティングモードで Natural ステートメント SET GLOBALS を使用して、Natural セッション内およびセッション中に特定のセッション関連のプロファイルパラメータを表示および設定（変更）できます。

これらの定義は、コマンドモードに、および現在のセッション中に実行されるすべてのプログラムに適用されます。

「[パラメータ値のランタイム割り当て用のセッションパラメータ](#)」または「SET GLOBALS」も参照してください。

プログラム／ステートメントレベル設定

Natural ステートメント FORMAT をプログラムで使用して、そのプログラムに有効なパラメータ値を設定できます。

また、端末コマンドを使用してステートメントレベルで特定のパラメータを設定することもできます。

開発環境設定

Natural のメインメニューの [Development Environment Settings] オプションを使用して、サブメニューを呼び出し、Natural 開発環境のモニタリングおよび設定に使用可能なツールを選択できます。

さまざまなパラメータ文字列の例

以下の各例は、次のパラメータ設定に基づいています。

パラメータ	パラメータモジュール、共有ニュークリアス (特殊なケース)	パラメータモジュール フロントエンド	代替パラメータモジュール ALTPARM	ユーザープロファイル MYPROF
DATSIZE	32 (デフォルト)	40	50	60
DSIZE	4	6	2 (デフォルト)	指定されていない
ESIZE	20	28 (デフォルト) NTSYS A : 40 NTSYS B : 50	NTSYS A : 60	80

以下に、さまざまなダイナミックパラメータ文字列の結果の例を示します。

例 1：ダイナミックパラメータなし

結果値	ソース
DATSIZE 40	フロントエンドモジュール
DSIZE 6	フロントエンドモジュール
ESIZE 28	フロントエンドモジュール
その他：デフォルト	フロントエンドモジュール

例 2：PARM=ALTPARM

結果値	ソース
DATSIZE 50	ALTPARM
その他：デフォルト	ALTPARM

例 3：SYS=A

結果値	ソース
DATSIZE 40	フロントエンドモジュール
DSIZE 6	フロントエンドモジュール
ESIZE 40	NTSYS フロントエンドモジュール

例 4：PARM=ALTPARM,SYS=A

結果値	ソース
DATSIZE 50	ALTPARM
DSIZE 2	ALTPARM
ESIZE 60	NTSYS、ALTPARM

例 5：PARM=ALTPARM,SYS=B

結果値	ソース
エラー	ALTPARM には NTSYS B 指定は含まれない

例 6：SYS=A,PROFILE=MYPROF

結果値	ソース
DATSIZE 60	MYPROF
DSIZE 6	フロントエンドモジュール
ESIZE 80	MYPROF

例 7 : SYS=A, PROFILE=MYPROF, ESIZE=100

結果値	ソース
DATSIZE 60	MYPROF
DSIZE 6	フロントエンドモジュール
ESIZE 100	ダイナミックパラメータ

例 8 : PROFILE=MYPROF, SYS=A

結果値	ソース
DATSIZE 60	MYPROF
DSIZE 6	フロントエンドモジュール
ESIZE 40	NTSYS フロントエンドモジュール

例 9 : DSIZE=8, SYS=A, PROFILE=MYPROF, PARM=ALTPARM

結果値	ソース
DATSIZE 50	ALTPARM
その他：デフォルト	ALTPARM

12 パラメータ値の割り当て

- パラメータ値の割り当てのソース 70
- パラメータ値のスタティックな割り当て 71
- パラメータ値のダイナミックな割り当て 72
- パラメータ値のランタイム割り当て用のセッションパラメータ 74

パラメータ値の割り当て

このドキュメントでは、プロファイルパラメータに値をスタティックに、ダイナミックに、およびランタイム時に割り当てる方法について説明します。

以下のトピックについて説明します。

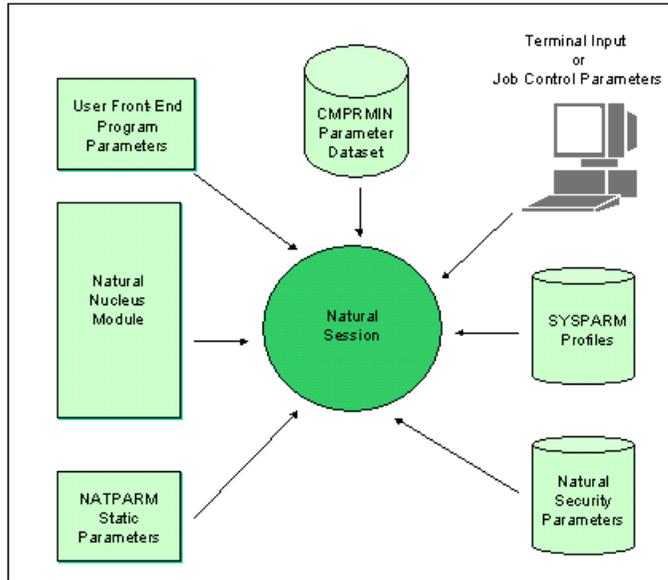
各プロファイルパラメータの詳細については、『パラメータリファレンス』ドキュメントを参照してください。

パラメータ値の割り当てのソース

プロファイルパラメータの値は以下の3つのソースから取得されます。

1. スタティックな割り当て
マクロ `NTPRM` および Natural パラメータソースモジュール (`NATPARM`) 内の他のマクロによって指定されたプロファイルパラメータ。これらのマクロはその後、Natural ニュークリアスでアセンブルおよびリンクされます。指定されないすべてのパラメータには、デフォルト値が割り当てられます。
2. ダイナミックな割り当て
Natural セッションの実行に対して指定されたパラメータ。これらのパラメータはスタティックな割り当てより優先され、現在の Natural セッションに有効です。ダイナミックパラメータは、フロントエンドプログラム、パラメータデータセット (`CMPRMN`)、セッション初期化 JCL、端末入力、または Natural Security から渡すことができます。また、特定のパラメータを Natural プログラムステートメントで上書きすることもできます。
3. セッションパラメータ
現在の Natural セッション内でシステムコマンド `GLOBALS` (またはレポーティングモードで `SET GLOBALS` ステートメント) によって指定されたパラメータ。このパラメータは、スタティックおよびダイナミックな割り当てを上書きします。

Natural パラメータ割り当ての図：



パラメータ値のスタティックな割り当て

Natural パラメータモジュール **NATPARM** を使用して、すべての Natural 環境のプロファイルパラメータをスタティックに割り当てます。

パラメータモジュールで、マクロ **NTPRM** および他のいくつかのマクロを使用して、パラメータを指定します。

パラメータモジュールで行われたすべてのパラメータ設定（パラメータ **CSTATIC** を除く）は、Natural セッションの開始時に動的に上書きできます。

技術的な理由から、一部のプロファイルパラメータには、対応するマクロがパラメータモジュールでのスタティックな割り当てに使用されます。そのため、スタティックな指定と動的な指定の構文は若干異なり、一般形式は以下のようになります。

スタティック：	MACRO-NAME KEYWORD1=value1,KEYWORD2=value2,...
動的：	PARAMETER-NAME=(KEYWORD1=value1,KEYWORD2=value2,...)

例：

- パラメータモジュール内のマクロ：NTSORT WRKSIZE=500,EXT=0N
- 同等の動的プロファイルパラメータ：SORT=(WRKSIZE=500,EXT=0N)

プロファイルパラメータに対応するマクロがある場合は、パラメータの説明に示されています。

パラメータ値の割り当て

スタティクな割り当ての詳細については、「[Naturalパラメータモジュールの使用](#)」を参照してください。

一部の Natural サブ製品（Natural for DL/I や Natural for DB2 など）では、追加のパラメータモジュールが使用されます。これらのモジュールについては、該当のサブ製品のドキュメントを参照してください。

パラメータ値のダイナミックな割り当て

Natural セッションの開始時にプロファイルパラメータをダイナミックに指定し、単一の Natural セッション中に Natural パラメータモジュール `NATPARM` の各プロファイルパラメータ設定を上書きできます。

例：

```
NUCNAME='NATNUC#5',IM=D,INTENS=1,DU=OFF,FUSER=(10,32),PROGRAM=' ',  
WORK=((1),AM=STD,DEST=WORK1,OPEN=INIT),PS=60,LS=120
```

Natural パラメータモジュールでスタティクにのみ指定できる `CSTATIC` を除いて、すべてのプロファイルパラメータはダイナミックに指定できます。

ダイナミックパラメータの割り当ては、コンマまたは空白で区切ります。ダイナミックパラメータの値に英数字以外の文字や特殊文字が含まれる場合は、値を一重引用符で囲んで指定する必要があります。 `NATCONFIG` の文字テーブルマクロ `NTSCTAB` で定義される特殊文字については、「[Natural コンフィグレーションテーブル](#)」を参照してください。

ダイナミックパラメータの使用は、マクロ `NTDYNP` または対応するダイナミックプロファイルパラメータ `DYNPARM` で有効または無効にできます。

一連のダイナミックパラメータをより簡単に指定するには、プロファイルパラメータ `PROFILE` または `SYS` を使用します。また、Natural Security では多数のダイナミックパラメータを設定できます。

ダイナミックパラメータ内にコメント文字列を挿入できます。コメントは「`*`」で開始し「`*/`」で終了します。コメント文字列の終了デリミタがない場合は、セッション初期化時にエラーメッセージが発行されます。

例：

```
PARM=MYPARMS /* my comment */ ADANAME=ADALNKR,PROFILE=MYPROF
```

ダイナミックパラメータ設定は、セッション開始時に Natural に渡されます。パラメータ値を Natural に渡すメソッドは環境によって異なります。

バッチモードの z/OS の例：

- 値は、Natural を開始する EXEC ジョブ制御文内の PARM キーワードで指定されます。
- また、ダイナミックパラメータはデータセット **CMPRMIN** で指定できます。
- さらに、制御を Natural に渡すフロントエンドプログラムを、z/OS 標準に従ってセッションのダイナミックパラメータで作成できます。

z/VSSE 環境でのダイナミックパラメータの指定

ダイナミックパラメータは、以下のように、JCL EXEC ステートメントで PARM 指定を使用して直接渡すことができます。

```
// EXEC NATBATCH,PARM='dynamic parameters...',SIZE=...
```

または、以下のように、PARM='SYSRDR' を指定して、Natural が SYSRDR からダイナミックパラメータを読み取るようにすることができます。

```
// EXEC NATBATCH,SIZE=...,PARM='SYSRDR'  
dynamic parameters  
...  
/* END OF DYNAMIC PARAMETERS
```

PARM キーワードが JCL EXEC ステートメントで指定されていない場合は、以下のように、互換性保持のために JCL OPTION ステートメントの SYSPARM パラメータがチェックされます。

```
...  
// OPTION SYSPARM='SYSRDR'  
// EXEC NATBATCH,SIZE=...  
dynamic parameters  
...  
/* END OF DYNAMIC PARAMETERS
```

パラメータ値のランタイム割り当て用のセッションパラメータ

一部のプロファイルパラメータには、対応するセッションパラメータを使用して、ランタイム時に Natural セッション内で値を割り当てることができます。セッションパラメータ値により、プロファイルパラメータ値が上書きされます。

プロファイルパラメータに対応するセッションパラメータがある場合は、プロファイルパラメータの説明に示されています。

セッションパラメータはシステムコマンド GLOBALS で指定されます。セッションパラメータの詳細については、『パラメータリファレンス』ドキュメントを参照してください。システムコマンドの詳細については、『コマンドリファレンス』ドキュメントを参照してください。

例：

```
GLOBALS SA=ON,IM=D
```

レポートモードでは、セッションパラメータをプログラムの SET GLOBALS ステートメントで指定することもできます。

一部のプロファイルパラメータは、端末コマンドを使用して Natural セッション内で上書きすることもできます。プロファイルパラメータに対応する端末コマンドがある場合は、プロファイルパラメータの説明に示されています。端末コマンドについては、『端末コマンド』ドキュメントを参照してください。

例：

```
SET CONTROL 'T=3279'
```

プロファイルパラメータ TTYPE の値が上書きされます。

13 プロファイルパラメータの概要

▪ システムファイル	76
▪ バッファサイズ	77
▪ 外部サブプログラム	77
▪ 出力レポートおよびワークファイル	78
▪ 日付／時刻設定	78
▪ 制限	79
▪ 文字割り当て	79
▪ 端末通信	80
▪ バッファプール	80
▪ 変換テーブル	81
▪ コードページおよび Unicode のサポート	81
▪ プロファイルパラメータの使用	81
▪ コンパイラオプション	82
▪ デバッグ	82
▪ バッチモード	83
▪ TP モニタ	83
▪ データベースへのアクセス	84
▪ Adabas での Natural	84
▪ その他の Software AG 製品との Natural	85
▪ その他のプロファイルパラメータ	87
▪ セッションの初期化および終了	88
▪ 内部使用に予約済みのパラメータ	89

このドキュメントでは、Natural管理者がサイトに適用するパラメータを決定できるように、使用可能なプロファイルパラメータの概要を説明します。パラメータは機能別に分類されています。

各プロファイルパラメータの詳細については、『パラメータリファレンス』ドキュメントを参照してください。

システムファイル

Natural システムファイルは、さまざまなデータおよびプログラムの保存に使用されます。『Natural システムアーキテクチャ』ドキュメントの「Natural システムファイル」を参照してください。

次のプロファイルパラメータは、すべてのシステムファイルに適用されます。

パラメータ	簡単な説明
DBID	Natural システムファイルのデフォルトのデータベース ID
FNR	Natural システムファイルのデフォルトのファイル番号
SYSPSW	Natural システムファイルのデフォルトのパスワード
SYSCIP	Natural システムファイルのデフォルトのサイファキー
ROSY	システムファイルへの読み取り専用アクセス (FNAT、FUSER、および FSEC のみ)

次のパラメータを使用して、各システムファイルのデフォルト値を上書きできます。

パラメータ	簡単な説明
FNAT	システムプログラム用の Natural システムファイル
FUSER	ユーザープログラム用の Natural システムファイル
FDIC	Predict システムファイル
FSEC	Natural Security システムファイル
FSP00L	Natural Advanced Facilities スプールファイル

バッファサイズ

Naturalでは、いくつかのバッファエリアを使用してプログラムおよびデータを保存します。場合によっては、バッファ効率を最大限に得るためにこれらのエリアの1つまたは複数のサイズを調整する必要があります。指定したスペースを使用できない場合は、要求されたバッファのサイズは0に設定されます。

パラメータ	簡単な説明
DATSIZE	ローカルデータ用のバッファのサイズ
DS	ストレージバッファのサイズ
DSIZE	デバッグバッファエリアのサイズ
ESIZE	ユーザーバッファ拡張エリアのサイズ
ISIZE	初期化バッファのサイズ
MONSIZE	SYSTP モニタバッファのサイズ
RDCSIZE	Natural Data Collector 用のバッファのサイズ
RJESIZE	NATRJE バッファの初期サイズ
RUNSIZE	ランタイムバッファのサイズ
WPSIZE	Natural ワークプールのサイズ

外部サブプログラム

次のパラメータは、Natural 以外のプログラムのダイナミックなロードおよび削除に影響します。

パラメータ	簡単な説明
CDYNAM	Natural 以外のプログラムのダイナミックなロード
CSTATIC	Natural にスタティックにリンクされるプログラム
DELETE	ダイナミックにロードされた Natural 以外のプログラムの削除
LIBNAM	外部プログラムロードライブラリの名前 (BS2000/OSD、z/OS、TSO のみ)
RCA	スタティックな Natural 以外のプログラムのアドレス解決
RCALIAS	Natural 以外のプログラムの外部名定義

出力レポートおよびワークファイル

次のパラメータは、Naturalレポートの作成中に使用されるさまざまな標準属性を制御します。

パラメータ	簡単な説明
DL	出力の表示長
EJ	ページ換え
FAMSTD	出力／ワークファイルのアクセスメソッド割り当ての上書き
HCAM	ハードコピーのアクセスメソッド
HCDEST	ハードコピーの出力先
INTENS	高輝度フィールドの出力
LS	Natural レコードの行サイズ
MAINPR	デフォルト出力レポート番号の上書き
MP	レポートの最大ページ数
PCNTRL	出力制御文字
PM	出力モード
PRINT	プリンタの割り当て
PS	Natural レポートのページサイズ
SF	フィールド間の空白
TQ	引用符の変換
TS	システムライブラリでプログラムからの出力を変換
WORK	ワークファイルの割り当て
ZP	ゼロ出力

日付／時刻設定

次のパラメータは、Natural による日付および時刻の値の処理と Natural で使用される内部日付／時刻に影響します。

パラメータ	簡単な説明
DD	日差
DFOUT	出力の日付フォーマット
DFSTACK	スタックの日付フォーマット
DFTITLE	デフォルトページタイトルの日付フォーマット
DTFORM	日付フォーマット

パラメータ	簡単な説明
TD	時差
YD	年差
YSLW	年スライディングウィンドウ

制限

次のパラメータを使用して、単一のプログラムが内部リソースを過剰に消費するのを防ぐことができます。

パラメータ	簡単な説明
LE	処理ループの制限超過時の処理
LT	処理ループの制限
MADIO	画面 I/O 処理間の最大 DBMS 呼び出し
MAXCL	プログラム呼び出しの最大数
MT	最大 CPU 時間
PD	NATPAGE により取得されるページ数

文字割り当て

次のパラメータを使用して、デフォルトの文字割り当てを変更できます。

パラメータ	簡単な説明
CVMIN	入力時の制御変数の修正
FC	INPUT ステートメント用の充填文字
FCDP	ダイナミック保護入力フィールドの充填文字
CF	端末コマンドの文字
DC	小数点文字
HI	ヘルプ文字
IA	INPUT 割り当て文字
ID	INPUT 区切り文字

端末通信

次のパラメータは、ビデオ端末上の Natural の使用に影響します。

パラメータ	簡単な説明
ATTN	アテンションキー割り込みのサポート
CLEAR	NEXT モードでの CLEAR キーの処理
DSC	データストリーム圧縮 (3270 タイプ端末)
ESCAPE	端末コマンド %% および % の無視
IKEY	PA キーおよび PF キーの処理
IM	入力モード
KEY	PA、PF、CLEAR キーへの値割り当て
LC	小文字から大文字への変換
ML	メッセージ行の位置
RM	修正したフィールドの再転送
SA	サウンド端末アラーム
TTYTYPE	端末タイプ

バッファプール

次のパラメータは、Natural バッファプールに影響します。

パラメータ	簡単な説明
BPCSIZE	Natural バッファプールのキャッシュサイズ
BPC64	Natural バッファプールのキャッシュサイズ
BPI	バッファプールの初期化
BPLIST	Natural バッファプールのプリロードリストの名前
BPMETH	バッファプールスペースの検索アルゴリズム
BPNAME	Natural グローバルバッファプールの名前
BPPROP	グローバルバッファプールの反映
BPSFI	バッファプールでの最初のオブジェクトの検索
BPSIZE	Natural ローカルバッファプールのサイズ
BPTXT	Natural バッファプール内のテキストセグメントのサイズ

変換テーブル

次のパラメータを使用して、Naturalで使用されるさまざまな文字変換テーブルを上書きできます。

パラメータ	簡単な説明
CCTAB	プリンタエスケープシーケンス制御文字
CP	コードページ
SCTAB	スキャナ文字タイプテーブル
TAB	標準出力変換テーブル
TABA1	EBCDIC から ASCII への変換テーブル
TABA2	ASCII から EBCDIC への変換テーブル
TABL	"SYS" ライブラリからの出力用変換テーブル
TAB1	代替出力変換テーブル
TAB2	代替入力変換テーブル
UTAB1	小文字から大文字への変換テーブル
UTAB2	大文字から小文字への変換テーブル

コードページおよび Unicode のサポート

コードページおよび Unicode のサポートに使用できるプロファイルパラメータの概要については、『Unicode およびコードページのサポート』ドキュメントの「Unicode / コードページ環境の設定と管理」の「プロファイルパラメータ」を参照してください。

プロファイルパラメータの使用

次のパラメータは、Natural プロファイルパラメータの使用に影響します。

パラメータ	簡単な説明
DYNPARM	ダイナミックパラメータの使用の制御
PARM	代替パラメータモジュール
PLOG	ダイナミックパラメータのロギング
PROFILE	ダイナミックパラメータプロファイルのアクティブ化
SYS	一連のダイナミックプロファイルパラメータのアクティブ化

パラメータ	簡単な説明
USER	プロファイルパラメータの使用の制限

コンパイラオプション

以下のパラメータを使用して、Natural コンパイラを制御できます。

パラメータ	簡単な説明
CMPO	コンパイルオプション
FS	ユーザー定義変数のデフォルトのフォーマット/長さ設定
SM	ストラクチャードモードでのプログラミング
XREF	クロスリファレンス機能の使用

デバッグ

次のパラメータは、デバッグ目的で使用できます。

パラメータ	簡単な説明
CANCEL	ダンプ付きのセッションのキャンセル
DU	ダンプ生成
ETRACE	外部トレース機能
ITRACE	内部トレース機能
RELO	ストレージスレッドの再配置
TRACE	トレース対象のコンポーネントの定義

バッチモード

次のパラメータは、Natural がバッチモードで使用されている場合に適用されます。

パラメータ	簡単な説明
CC	バッチモードでのエラー処理
CPOBJIN	バッチ入力ファイルのコードページ
CPPRINT	バッチ出力ファイルのコードページ
CPSYNIN	コマンド用のバッチ入力ファイルのコードページ
ECHO	入力データの出力制御
OBJIN	Natural 入力ファイルとしての CMOBJIN の使用
READER	入力用システム論理ユニット (z/VSE のみ)

TP モニタ

次のパラメータは、Natural が TP モニタ (Com-plete、CICS、CMS、IMS/TM、UTM) で使用されている場合に適用されます。

パラメータ	簡単な説明
ASYNNAM	非同期処理の出力システム ID (UTM)
OUTDEST	非同期処理の出力先 (CICS、Com-plete、UTM)
PSEUDO	擬似会話型モード (CICS)
SENDER	非同期処理の画面出力先 (CICS、Com-plete、IMS/TM、UTM)
SKEY	ストレージ保護キー
SUBSID	サブシステム ID

データベースへのアクセス

次のパラメータは、Natural によるデータベースアクセスの処理に影響します。

パラメータ	簡単な説明
DB	データベースタイプおよびオプション
DBCLOSE	セッション終了時のデータベースのクローズ
DBOPEN	ETID なしのデータベースのオープン
DBROLL	ロールアウト前のデータベースコール
DBUPD	データベースの更新
ENDBT	セッション終了時の BACKOUT TRANSACTION の発行
ET	END/BACKOUT TRANSACTION ステートメントの実行
ETDB	トランザクションデータ用のデータベース
ETEOP	プログラム終了時の END TRANSACTION の発行
ETIO	端末 I/O での END TRANSACTION の発行
LFIL	論理ファイルのダイナミックな指定
OPRB	データベースのオープン/クローズ処理
RCFIND	FIND ステートメントに対するレスポンスコード 113 の処理
RCGET	GET ステートメントに対するレスポンスコード 113 の処理
TF	データベース ID/ファイル番号の変換
UDB	ユーザーデータベース ID

Adabas での Natural

次のパラメータは、Natural が Adabas で使用されている場合に適用されます。

パラメータ	簡単な説明
ADANAME	Adabas リンクルーチンの名前
ADAMODE	Adabas インターフェイスモード
ADAPRM	Review/DB サポート
ADASBV	Adabas セキュリティバイバリュウ
ETID	Adabas ユーザー ID
RI	リリース ISN
WH	ホールド状態でのレコードの待機

その他の Software AG 製品との Natural

Adabas テキストの取得

パラメータ	簡単な説明
TSIZE	Adabas テキスト取得用のバッファエリアのサイズ

Con-nect

パラメータ	簡単な説明
CSIZE	Con-nect バッファエリアのサイズ

EntireX Broker

パラメータ	簡単な説明
BSIZE	EntireX Broker バッファのサイズ

Entire DB

パラメータ	簡単な説明
ZSIZE	Entire DB バッファエリアのサイズ

Entire System Server

パラメータ	簡単な説明
ASIZE	Entire System Server 補助バッファ

Entire Transaction Propagator

次のパラメータは、Entire Transaction Propagator を使用している場合に適用されます。

パラメータ	簡単な説明
ETPSIZE	Entire Transaction Propagator バッファのサイズ

Natural Advanced Facilities

次のパラメータは、Natural Advanced Facilities を使用している場合に適用されます。

パラメータ	簡単な説明
NAFSIZE	Natural Advanced Facilities 用のバッファのサイズ
NAFUPF	Natural Advanced Facilities ユーザープロファイル

Natural Connection

次のパラメータは、Natural Connection を使用している場合に適用されます。

パラメータ	簡単な説明
PC	パーソナルコンピュータのアクセスメソッドの制御 (Natural Connection)
XSIZE	ユーザーサブシステム用のバッファのサイズ

Natural データベースインターフェイス

次のパラメータは、以下のデータベースインターフェイスを使用している場合に適用されます。

パラメータ	簡単な説明
DB2SIZE	Natural DB2 または SQL/DS インターフェイス用のバッファエリアのサイズ
DLISIZE	Natural DL/I インターフェイス用のバッファエリアのサイズ
VSIZE	Natural VSAM インターフェイス用のバッファエリアのサイズ

Natural Expert

パラメータ	簡単な説明
EXCSIZE	Natural Expert C インターフェイス用のバッファのサイズ
EXRSIZE	Natural Expert 規則表用のバッファのサイズ

Natural Optimizer Compiler

次のパラメータは、Natural Optimizer Compiler を使用している場合に適用されます。

パラメータ	簡単な説明
OPT	Natural Optimizer Compiler の制御

Natural ワークステーションインターフェイス

次のパラメータは、Natural ワークステーションインターフェイスを使用している場合に適用されます。

パラメータ	簡単な説明
WSISIZE	Natural ワークステーションインターフェイス用のバッファ

Software AG Editor

次のパラメータは、Software AG Editor を使用している場合に適用されます。

パラメータ	簡単な説明
EDPSIZE	Software AG Editor 補助バッファプールのサイズ
SSIZE	Software AG Editor 用のバッファのサイズ

その他のプロファイルパラメータ

パラメータ	簡単な説明
CM	コマンドモード
CPCVERR	変換エラー
ETA	エラートランザクションプログラム
MAXROLL	ロールアウトの前の CMROLL 呼び出しの数
MSGSF	システムエラーメッセージのフル表示

パラメータ	簡単な説明
NC	Natural システムコマンドの使用
OPF	ヘルプルーチンによる保護フィールドの上書き
POS22	POS システム関数用のバージョン 2.2 アルゴリズム
RDCEXIT	Natural Data Collector ユーザー出口の定義
RECAT	ダイナミックな再カタログ
REINP	不正データに対する内部 REINPUT ステートメントの発行
RFILE	レコーディング用ファイル
RPC	リモートプロシージャコール設定
SI	ダブルバイト文字セット用のシフトインコード
SL	ソース行の長さ
S0	ダブルバイト文字セット用のシフトアウトコード
SORT	ソートプログラムの制御
SYNERR	構文エラーの制御
ULANG	ユーザー言語
ZD	ゼロ除算チェック

セッションの初期化および終了

次のパラメータは、Natural セッションの初期化および終了に影響します。

パラメータ	簡単な説明
AUTO	自動ログオン
ENDMSG	セッション終了メッセージの表示
IMSG	セッション初期化エラーメッセージ
ITERM	初期化エラー時のセッション終了
MENU	メニューモード
NUCNAME	共有ニュークリアスの名前
PROGRAM	Natural セッション後に制御を受け取るプログラム
STACK	スタックへのデータ/コマンドの配置
STEPLIB	追加の Steplib ライブラリ

内部使用に予約済みのパラメータ

 **注意:** 以下のパラメータの値は変更しないでください。

次のパラメータは、Natural が内部で使用するため予約されています。

パラメータ	簡単な説明
ASPSIZE	(内部使用)
CFWSIZE	(内部使用)
LOG	(内部使用)
NISN	(内部使用)
RDNODE	(内部使用)
RDPORT	(内部使用)
TPF	(内部使用)
USERBUF	(内部使用)

14 Natural パラメータモジュールの使用

- デフォルトの Natural パラメータモジュール NATPARM の使用 92
- 新しい Natural パラメータモジュールの作成 92
- NTPRM マクロ - Natural パラメータモジュールの作成 93
- パラメータモジュールの使用の制限 93
- Natural パラメータモジュールでのマクロの使用 94

このドキュメントでは、Naturalパラメータモジュールのアセンブル方法について説明します。

以下のトピックについて説明します。

各プロファイルパラメータの詳細については、『パラメータリファレンス』ドキュメントを参照してください。

デフォルトの Natural パラメータモジュール NATPARM の使用

デフォルトの Natural パラメータモジュール NATPARM には、ほとんどの環境で使用できる事前定義された一連のパラメータが含まれています。モジュールはソース形式で提供され、必要に応じて変更できます。

新しい Natural パラメータモジュールの作成

デフォルトの Natural パラメータモジュールを使用または変更する代わりに、1つまたは複数の代替の Natural パラメータモジュールをさまざまな目的で作成し、Natural プロファイルパラメータ PARM を使用して必要に応じてロードすることができます。このパラメータモジュールの使用は特定のユーザーに制限できます（「[パラメータモジュールの使用の制限](#)」を参照）。

▶手順 14.1. 新しい（代替の）Natural パラメータモジュールを作成するには

- 1 マクロ `NTPRM` をアセンブルします（以下の「[アセンブルマクロコーディング規則](#)」も参照）。
- 2 1つまたは複数のオプションのパラメータマクロを追加します（[下記参照](#)）。

複数のパラメータマクロを指定する場合は、`NTPRM` マクロを最初に指定する必要があります。`NTPRM` マクロの後の他のマクロは、任意の順序で指定できます。

 **注意:** オペレーションのバッチおよび TP モード用に個別のパラメータモジュールを作成する必要はありません。Natural を実行する環境に適用されないパラメータは無視されません。

NTPRM マクロ - Natural パラメータモジュールの作成

Natural パラメータモジュールを作成するには、NTPRM マクロをアセンブルする必要があります。

通常は、NTPRM マクロのプロファイルパラメータのデフォルト値を使用できます。デフォルト値が要件に合わない場合は、独自の値で上書きできます。

個別のプロファイルパラメータの詳細については、『パラメータリファレンス』ドキュメントを参照してください。

NTPRM の構文

このマクロの構文は以下のとおりです。

```
NTPRM parameter=value,...
```

アセンブルマクロコーディング規則

パラメータ値を変更する際は、アセンブルマクロコーディング規則に従う必要があります。以下に例を示します。

- 最初のエントリは列 2 以降で開始する必要があり、列 71 を超えることはできません。
- 他の行に継続する場合は、最後のエントリの後にコンマを置き、空白以外の文字を列 72 に挿入し、列 16 で始まる次の行でエントリを続けます。
- パラメータとその値は常に同じ行に入力する必要があります。

パラメータモジュールの使用の制限

マクロ NTUSER をパラメータモジュールに追加して、その使用を特定のユーザーに制限できます。

▶手順 14.2. パラメータモジュールの使用を制限するには

- 1 マクロ NTUSER をパラメータモジュールに追加します。
- 2 このマクロで、そのパラメータモジュールの使用を有効にするユーザーの ID を定義します。

これらのユーザーのみが、プロファイルパラメータ PARM を使用してそのパラメータモジュールの名前を指定できます。

Natural パラメータモジュールでのマクロの使用

Natural パラメータモジュールには、先頭にマクロ `NTPRM` があります。これ以外に、以下のマクロを任意の順序で指定できます。

マクロ	機能
NTALIAS	Natural 以外のプログラムの外部名
NTBPI	バッファプールの初期化
NTCCTAB	プリンタエスケープシーケンス定義
NTCFICU	さまざまな Unicode 設定に対して Unicode サポートを有効にします。
NTCMP0	コンパイルオプション
NTCSTAT	Natural にスタティックにリンクされているプログラム
NTDB	データベースタイプおよびオプション
NTDS	ストレージバッファのサイズ
NTDYNP	ダイナミックパラメータの使用の制御
NTEDBP	Software AG エディタバッファプールの定義
NTFILE	NTLFILE、「古いNTFILE マクロの構文」を参照
NTLFILE	論理ファイルの指定
NTOPRB	データベースのオープン/クローズ処理
NTOPT	Natural Optimizer Compiler の制御
NTPRINT	出力ファイルの割り当て
NTPRM	Natural パラメータモジュールの作成
NTRPC	リモートプロシージャコールの処理
NTSCTAB	スキャナ文字
NTSORT	ソートプログラムの制御
NTSYS	一連のダイナミックプロファイルパラメータの定義およびアクティブ化
NTTAB	標準の出力文字変換
NTTABA1	EBCDIC から ASCII への変換
NTTABA2	ASCII から EBCDIC への変換
NTTABL	「SYS」ライブラリ出力変換
NTTAB1	代替出力変換
NTTAB2	代替入力変換
NTTF	データベース ID/ファイル番号の変換
NTTRACE	トレース対象のコンポーネントの定義
NTUSER	プロファイルパラメータ文字列およびモジュールの使用の制限
NTUTAB1	小文字/大文字の変換

マクロ	機能
NTUTAB2	大文字／小文字の変換
NETWORK	ワークファイルの割り当て
NTXML	PARSE XML および REQUEST DOCUMENT ステートメントのアクティブ化

15 z/OS 環境

このドキュメントでは、z/OSオペレーティングシステム環境でのNaturalについて説明します。

- **z/OS での Natural** z/OS 環境で Natural をオンラインまたはバッチモードで実行する際の特別な考慮事項について概要を説明します。
- **認可サービスマネージャ** z/OS 環境で使用可能な認可サービスマネージャ (ASM) の機能と操作について説明します。
- **z/OS および z/VSE 環境下の Natural 共有ニュークリアス** Natural 共有ニュークリアスの機能と使用について説明します。
- **Natural ロールサーバーの機能** Natural ロールサーバーの一般的な機能と、単一 z/OS システムおよび並列 Sysplex 環境での使用について説明します。
- **Natural ロールサーバーの操作** ロールサーバーのシステム要件、操作、パフォーマンスチューニング、および再起動可能性について説明します。



注意: Naturalセッションランタイム中のロールサーバー使用時にNaturalが受け取るコードは、対応するTPインターフェイスで出力されます（「CICS環境でのNatural」または「IMS/TM環境のNatural」を参照）。これらのコードのリストについては、『メッセージおよびコード』ドキュメントの「ロールサーバー要求のリターンコードおよび理由コード」を参照してください。

16 z/OS での Natural

▪ Natural サブシステム	100
▪ 共有ニュークリアス	100
▪ TP モニタインターフェイス	100
▪ データベース管理システムへのインターフェイス	101
▪ z/OS 環境下のバッチモードの Natural	101
▪ z/OS 環境のサーバーとしての Natural	101

このドキュメントでは、Natural を z/OS 環境で実行する際の特別な考慮事項について概要を説明します。

Natural サブシステム

z/OS 環境の Natural サブシステムは、以下のコンポーネントで構成されます。

- 1 つまたは複数のグローバルバッファプール
- 認可サービスマネージャ
- ロールサーバー

Natural サブシステムは、Natural プロファイルパラメータ SUBSID および上記コンポーネントの対応する起動パラメータによって識別されます。デフォルトのサブシステム名は NAT4 です。

Natural サブシステム技術により、複数のロールサーバーを同時に使用し、複数の独立した一連のグローバルバッファプールを作成することができます。互いに完全に独立した複数の Natural ランタイム環境の作成が可能です。

共有ニュークリアス

Natural 共有ニュークリアスの利点については、「[z/OS および z/VSE 環境下の Natural 共有ニュークリアス](#)」を参照してください。

TP モニタインターフェイス

z/OS 環境の Natural で使用できる TP モニタインターフェイスについては、以下のセクションを参照してください。

- *Com-plete/SMARTS 環境での Natural*
- *CICS 環境での Natural - 概要*
- *TSO 環境での Natural*
- *IMS/TM 環境の Natural - 概要*

これらのセクションは、Natural の『TP モニタインターフェイス』ドキュメントにあります。

データベース管理システムへのインターフェイス

Software AG のデータベース管理システムである Adabas を除き、データベース対話が必要なすべての操作は対応する Natural インターフェイスモジュールで実行されます。

z/OS 環境の Natural で使用できるデータベースのインターフェイスについては、以下のドキュメントを参照してください。

- *Natural for DB/2*
- *Natural for VSAM*
- *Natural for DL/I*

z/OS 環境下のバッチモードの Natural

「[バッチモードの Natural \(すべての環境\)](#)」および「[z/OS 環境下のバッチモードの Natural](#)」を参照してください。

z/OS 環境のサーバーとしての Natural

Natural は、プログラミング言語である以外に、クライアント/サーバー環境でサーバーとして機能できます。詳細については、「[z/OS 環境のサーバーとしての Natural](#)」を参照してください。

17 z/OS 環境での認可サービスマネージャ

■ ASM の概要	104
■ ASM システム要件	105
■ ASM の操作	106

このドキュメントでは、z/OS 環境の Natural で使用可能な認可サービスマネージャ（ASM）の機能と操作について説明します。

以下のトピックについて説明します。

ASM の概要

認可サービスマネージャ（ASM）は、認可されたオペレーティングシステム機能を Natural で使用できるようにします。これらの機能には、SMF レコードの書き込みや、カップリング機能（CF）経由の並列 Sysplex 通信などがあります。ASM は、PC ルーチン経由で機能を提供し、独自のアドレススペースで実行されます。

以下の認可された機能が提供されます。

- Natural バッファプール管理メッセージの通知
- システムキー内のグローバルバッファプールへの書き込みアクセス
- SMF レコードの書き込み
- セッション情報プール（SIP）への Natural セッション情報の保存

最初の3つの機能は常に使用可能です。SIP はオプションで、起動パラメータ経由で使用可能になります。ASM の開始方法については、「[ASM の開始](#)」を参照してください。

以下の場合、ASM を使用する必要があります。

- Natural プロファイルパラメータ BPPROP が、PLEX、GLOBAL、または GPLEX に設定されている（バッファプールの反映を使用）。
- Natural グローバルバッファプールがシステムキー内で割り当てられている。「[Natural GBP オペレーティングプログラムのインストール](#)」を参照してください。
- CICS 環境で Natural が、並列 Sysplex で使用されている（SIP 機能が必要）。
- IMS/TM 環境で Natural が、端末指向の非会話型モードで使用されている（SIP 機能を使用）。
- IMS/TM 環境で Natural に、SMF レコードを書き込むアカウントिंग機能が使用されている。

セッション情報プール（SIP）には、Natural セッション情報レコードが保持されます。端末指向の非会話型モードでは、端末 I/O 後に Natural セッションを続行するために、NCI および NII インターフェイスにこれらのレコードが必要です。並列 Sysplex で実行している場合は、SIP が CF で作成され、データスペースが中間バッファとして使用されて、CF への不要なアクセスが回避されます。そうでない場合は、SIP はデータスペースで作成されます。

ASM が並列 Sysplex で使用される場合は、参加する各 z/OS イメージで1つの ASM インスタンスを開始する必要があります。

Natural/CICS に関する注意事項：CICS システムリカバリテーブルには、z/OS システムアベンドコード 0D6 が含まれている必要があります。

ASM システム要件

- APF 認可
- システムリンケージインデックス
- CF 構造
- XCF シグナルパス

APF 認可

IEWL パラメータ AC(1) を指定して、モジュール NATASM vr (vr =バージョンとリリース番号) および NATBPMGR を Authorized Program Facility (APF) ライブラリにリンクします。「z/OS 環境での Natural のインストール手順」を参照してください。

システムリンケージインデックス (システム LX)

ASM で 1 つのシステムリンケージインデックス (システム LX) を確保するため、ライブラリ SYS1.PARMLIB のメンバ IEASYS xx に十分大きい値の NSYSLX があることをチェックしてください。

 **注意:** ASM を終了すると、システム LX が使用されているので、アドレススペース ID は使用できなくなります。これは、次回の IPL で再度使用可能になります。

CF 構造

CF 構造は、SIP を並列 Sysplex で実行する場合にのみ使用されます。必要なスペースは、以下の式で計算できます。

$$30 \text{ KB} + (\text{SIP slot size in bytes} + 165) * (\text{number of SIP slots} + 8)$$

512 バイトの 500 の SIP スロットごとに、以下のように定義します。

```
STRUCTURE NAME(NATASM) SIZE(380) PREFLIST(CF1)
```

XCF シグナルパス

バッファプール管理メッセージを並列 Sysplex で伝達するために、XCF シグナルサービスが使用されます。メッセージの長さは、最小が 64 バイトで最大が 2048 バイトです。メッセージが送信される頻度は、Natural オブジェクトを操作（システムコマンド CATALOG、STOW、または DELETE を使用）する頻度によって異なります。

ASM の操作

以下について説明します。

- ASM の開始
- ASM メッセージ、コンディションコード、およびアベンドコード
- ASM オペレータコマンド

ASM の開始

ASM をバッチジョブまたは開始タスクとして開始するには、モジュール NATASMvr を実行します。vr は現在の Natural のバージョンおよびリリース番号です。JCL EXEC ステートメントで、PARM に以下のパラメータを指定します。

```
subsystem-id,XCF-group-name,CF-structure-name,number-of-SIP-slots,SIP-slot-size,message-case
```

すべて位置指定パラメータであり、コンマで区切る必要があります。詳細については、以下の表を参照してください。

パラメータ	指定可能な値	デフォルト	コメント
subsystem-id	空白以外の 4 バイトの文字列	NATv	指定値は、Natural プロファイルパラメータ SUBSID の値と一致する必要があります（v はバージョン）。 注意: CICS 環境の Natural の場合は、適切なサブシステム ID を設定するには、NCMDIR マクロの CICSPLX パラメータを参照してください。
XCF-group-name	任意の有効な XCF グループ名	なし	シグナルサービス用の XCF グループ名。

パラメータ	指定可能な値	デフォルト	コメント
<i>CF-structure-name</i>	任意の有効な CF 構造名	なし	オプション。SIP が使用される場合にのみ必要です。SIP 機能に使用される CF 構造の名前です。
<i>number-of-SIP-slots</i>	1 - 32767	なし	オプション。SIP が使用される場合にのみ必要です。CF 構造がまだ割り当てられていない場合に割り当てるスロットの数です。値を省略するか、0 を指定すると、構造全体で保持可能な数のスロットが使用されます。
<i>SIP-slot-size</i>	256、512、1024、2048、4096	1024	CF 構造がすでに割り当てられている場合は、指定した値は無視されます。
<i>message-case</i>	UCTRAN または空白	空白	認可サービスマネージャですべてのメッセージを大文字で発行する場合は、UCTRAN を指定します。

例：

```
//ASM EXEC PGM=NATASMvr,PARM='NATv,NATXCF,CFSIP,1500,512'
```

サブシステム ID は NATv です。バッファプール通信のメッセージグループは NATXCF です。セッション情報プールの構造は CFSIP です。1500 の SIP スロットが使用され、各スロットのサイズは 512 バイトです。

```
//ASM EXEC PGM=NATASMvr,PARM='NATv,NATXCF,CFSIP'
```

上記と似ていますが、SIP スロットについて次の点が異なります。

ASM は、CFSIP 構造で保持可能な数の SIP スロットを使用します。スロットの各サイズは 1024 バイトです。

```
//ASM EXEC PGM=NATASMvr,PARM='NATv,NATXCF,,500,512'
```

SIP サービスは、カップリング機能を使用しませんが、ストレージに 500 の SIP スロットを構築します。各スロットのサイズは 512 バイトです。

```
//ASM EXEC PGM=NATASMvr,PARM='NATv,NATXCF'
```

SIP サービスは使用できません。

ASM メッセージ、コンディションコード、およびアベンドコード

ASM は、WTO マクロ (ROUTCDE=11) を使用して、情報およびエラーメッセージを JESMSG LG に書き込みます。メッセージの先頭には、メッセージ ID と ASM のジョブ名が付きます。例えば、以下のようになります。

```
ASM0005 FBASMvr
```

この例では、認可サービスマネージャバージョン *vrs* (*vr* の *v* はバージョン、*r* はリリース、*s* はシステムメンテナンスレベル) は有効です。

以下のコンディションコードが使用されます。

コンディションコード	説明
0	正常終了
12	不適切なパラメータの入力
16	ランタイムエラーの発生
20	サブタスクの失敗
24	アベンドの発生
>100	ワーキングストレージの割り当ての失敗

以下のユーザーアベンドコードが使用されます。

アベンドコード	理由	コメント
U0100	IXCJOIN が失敗しました。	アベンドレジスタ 14 に理由コードが含まれます。
U0101	IXCQUERY が失敗しました。	アベンドレジスタ 14 に理由コードが含まれます。
U0103	アクティブメンバリストがいっぱいです。	Software AG サポートに連絡してください。
U0104	IXCMSGI が失敗しました。	アベンドレジスタ 14 に理由コードが含まれます。
U0105	メッセージ出口がパージタスク要求ブロックを取得できませんでした。	Software AG サポートに連絡してください。
U0106	IXLCONN 用のワークスペースを取得できませんでした。	Software AG サポートに連絡してください。
U02xx	DSPSERV CREATE が失敗しました。	xx が理由コードです。
U03xx	ALESERV ADD が失敗しました。	xx が理由コードです。
U04xx	ALESERV ADD が失敗しました。	xx が理由コードです。
U05xx	IXLCONN が失敗しました。	xx が理由コードです。
U06xx	IXLLIST WRITE が失敗しました。	xx が理由コードです。

理由コードの詳細については、IBM の『MVS プログラミング：シスプレックス・サービス解説書』を参照してください。エラーが環境固有で、理由が不明な場合は、Software AG サポートにお問い合わせください。

ASM オペレータコマンド

以下のコマンドは、MODIFY コマンドを使用して ASM に渡すことができます。

コマンド	説明
TERM	ASM を終了します。
TRSTART	デバッグ機能です。Software AG から推奨された場合にのみ使用します。トレースタスクを有効にします。GTF が開始され、ユーザーレコード 202 に対して有効になっている場合は、トレースレコードが GTF に書き込まれます。
TRSTOP	トレースタスクを無効にします。
SNAP	デバッグ機能です。ASM のアドレススペースが SYSUDUMP にダンプされます。
VLIST	ASM にリンクされているモジュールの名前、バージョン、およびアセンブリ時間を表示します。

SIP サービスのリターンコードおよび理由コードのリストについては、『メッセージおよびコード』ドキュメントの「SIP サービスのリターンコードおよび理由コード」を参照してください。

18 z/OS および z/VSE 環境下の Natural 共有ニュークリ アス

- 環境に依存しないニュークリアス 112
- 共有ニュークリアスの作成 116
- 共有ニュークリアスのインストール 116
- ニュークリアスへのサブ製品のリンク 116
- 単一環境の共有ニュークリアス 117
- 環境依存のニュークリアス 118
- スタティックにリンクされた Natural 以外のプログラム 118
- ダイナミックに呼び出された Natural 以外のプログラム 119

このドキュメントでは、z/OS および z/VSE 環境での Natural 共有ニュークリアスについて説明します。

環境に依存しないニュークリアス

Natural は、環境非依存ニュークリアスと環境依存ニュークリアスの2つの機能部分に分割できます。

共有ニュークリアスの環境非依存部分は、オペレーティングシステムの以下の共有エリアに配置されます。

- z/OS 環境の場合：リンクパック記憶域 (LPA) または拡張リンクパック記憶域 (ELPA)
- z/VSE 環境の場合：共用仮想領域 (SVA)

オペレーティングシステムのこれらの特殊エリアから実行することで、非依存ニュークリアスには、CICS、Com-plete、TSO、バッチモードなどの同じオペレーティングシステム内の複数のアドレススペース (つまり、リージョンやパーティション) から一般にアクセス (共有) できます。

共有ニュークリアスのコンポーネント

以下の基本モジュールをまとめてリンクして、非依存の Natural (共有) ニュークリアスを構築できます。

モジュール	機能
NATSTUB	Natural スタブモジュール。
NATURAL	Natural コンパイラおよびランタイム。
NATCONFIG	Natural コンフィグレーションモジュール。
NATCMOD	C ルーチンのバンドルモジュール (サーバー呼び出し)。
NATBPMGR	Natural バッファプールマネージャ
NAT2SORT	すべてのシステム用の Natural ソート (Natural の内部または外部いずれかのソートプログラムを使用する場合)。NAT2SORT をロードライブラリに含めることもできます。これにより、ランタイムにダイナミックにロードできるようになります。このためには、NAT2SORT をプロファイルパラメータ RCA で指定する必要があります。
NATRPCvr または NTRPC	Natural RPC ランタイム。 注意: このモジュールの複数のバージョンが使用可能な場合は、使用可能な Natural RPC のバージョンについて、Natural の最新の『リリースノート』を参照してください。このモジュールの単一のバージョンのみが使用可能な場合は、実際のモジュール名の『インストール』ドキュメントで、共有ニュークリアスをリンクするインストールジョブを参照してください。

モジュール	機能
NATEDIT	Natural プログラムエディタおよびマップエディタ。
NATTEXT	Natural 構文。
NATTXT2	Natural キーワード。
NATTXT3	Natural エラーメッセージの置換部分。
NATPM	Natural 出力モード。
INPL	INPL モジュール。
NATEDT	Software AG Editor モジュール。
NATLAST	最後の INCLUDE。

端末ドライバおよびバッチモードモジュール

以下のモジュールはオプションです。

モジュール	機能
NATTTY	Natural テレタイプサポート
NAT3270	3270 端末サポート
NAT3279	3279 端末サポート
NATWEB	Web I/O 端末コンバータ (『Unicode およびコードページのサポート』ドキュメントの「Natural アプリケーションの入力/出力処理」の「Web I/O インターフェイス」参照)
NATBTCH	Natural バッチモジュール

Unicode およびコードページのサポートに必要なモジュール

Unicode およびコードページのサポートにリンクするメインフレーム固有モジュールのリストについては、『Unicode およびコードページのサポート』ドキュメントの「Unicode/コードページ環境の設定と管理」の「ICU ライブラリ」を参照してください。

REQUEST DOCUMENT および PARSE XML ステートメントサポートに必要なモジュール

モジュール	機能
NATXML	ニュークリアスルーチンモジュール

詳細については、Natural の『インストール』ドキュメントの「REQUEST DOCUMENT および PARSE XML ステートメントサポートのインストール」を参照してください。

追加モジュールのリンク

追加モジュールのリンクが必要になる場合があります。例えば、すべての Natural リージョンで使用される共通のユーザー出口やユーザー定義プログラムなどです。リンクされたモジュールのエントリ名は、CMSTUB である必要があります。

共有ニュークリアスの利点

共有ニュークリアスには、以下の利点があります。

- 仮想ストレージの負荷が軽減される。
- システム内にニュークリアスのコピーは1つしか存在しないため、ページングアクティビティが減少する。
- Zap は 1 回しか適用されないため、メンテナンスの手間が少なくなる。

Natural の環境非依存部分を取り除いて共有ニュークリアスに置くことにより、環境依存ニュークリアスのサイズを大幅に縮小できます。これは、環境依存部分のみがバッチまたは TP モニタアドレススペースにロードされ、共有ニュークリアスはオペレーティングシステムのリンクパック記憶域からアクセスされるためです。

Natural バッチジョブで必要とされるストレージが削減されるため、ページングが減少し、オペレーティングシステムの全体的なパフォーマンスが向上します。共有ニュークリアスに同時アクセスするバッチジョブが増えると、削減量も大きくなります。

バッチ環境の場合と同様、オンライン環境で実行される Natural は同じ共有ニュークリアスにもアクセスできます。複数の CICS リージョンで Natural を実行するような実稼働環境では、仮想ストレージを大幅に節約できます。

また、Natural ニュークリアスに修正を適用するときにも利点があります。これらの Zap を共有ニュークリアスに適用する必要があるのは1回だけで、その後は複数の環境（CICS、Com-plete、TSO、バッチなど）からアクセスされるためです。

Natural for VSAM、Natural for DB2、Natural for DL/I、Natural Advanced Facilities などの製品を使用している場合は、これらの製品はすべて共有ニュークリアスから実行できるため、追加の利点もあります。これらの製品をインストールするときは、これらの製品に固有の INCLUDE ステートメントを共有ニュークリアスのリンクエディットに配置するだけで済みます。

管理面

LPA/ELPA または SVA にインストールされたモジュールでは、Zap をオンラインで適用できません。LPA/ELPA または SVA は書き込み保護されているためです。z/OS 環境では、オペレータコマンド SETPROG を使用して、共有ニュークリアスの新しいコピーを LPA/ELPA にロードできます。ただし、特定の環境で修正をテストするには、サイトのニーズに応じて調整できる以下のメソッドのいずれかを使用することをお勧めします。

環境	要件
バッチモード	STEPLIB 連結に追加するロードライブラリに、共有ニュークリアスをリンクエディットします。オペレーティングシステムは、共有エリアのコピーの代わりに共有ニュークリアスのこのコピーにアクセスします。
CICS	<p>CICS 起動手順で DFHRPL 連結に追加するロードライブラリに、共有ニュークリアスをリンクエディットします。これにより、CICS は、共有エリアからではなく DFHRPL ライブラリから共有ニュークリアスをロードできるようになります。</p> <p>共有ニュークリアスが SHR=NO を読み取って CICS が共有エリアの代わりに DFHRPL ライブラリにアクセスするように、ALT (代替ロードテーブル) エントリを変更する必要があります。</p> <p>CICS バージョン 3.3.0 以降を使用している場合は、ALT がいないため、代わりに共有ニュークリアスの PPT エントリにこの変更を行います。</p> <p>このプログラム定義に、z/OS では USELPACOPY (NO) を、z/VSE では USESVACOPY (NO) を指定します。</p>
Com-plete/TPF	共有ニュークリアスを、Com-plete ユーザープログラムライブラリにリンクし、Com-plete SYSPARM の RESIDENTPAGE プログラムのリストに追加するか、RESIDENTPAGE としてダイナミックにロードします。
TSO	TSO 環境の Natural 用の TP 依存ニュークリアスを含む同じロードライブラリに、共有ニュークリアスをリンクエディットします。CLIST が実行されると、オペレーティングシステムは、共有エリアのコピーの代わりに共有ニュークリアスのこのコピーにアクセスします。
IMS/TM	IMS/TM アプリケーションリージョンの実行手順で STEPLIB 連結に追加するロードライブラリに、共有ニュークリアスをリンクエディットします。Natural が開始されると、オペレーティングシステムは、共有エリアからではなく STEPLIB から共有ニュークリアスにアクセスします。

共有ニュークリアスの作成

共有ニュークリアスは、SMA ジョブ NATI060 でリンケージエディタにより基本 Natural インストールのオプション部分として作成されます。

リンケージエディタ INCLUDE ステートメントを共有ニュークリアス用に設定する場合は、必ず Natural の『インストール』ドキュメントのインストール手順に従ってください。

よくある間違いは、共有ニュークリアスまたは非共有ニュークリアスへのリンクエディット INCLUDE ステートメントを省略または追加することです。これにより、Natural セッションを開始しようとすると、予期しない結果になることがあります。この問題が発生する場合は、インストール手順を見直し、必要であれば Software AG サポートにお問い合わせください。

共有ニュークリアスのインストール

共有ニュークリアスのインストールについては、Natural の『インストール』ドキュメントを参照してください。さまざまな Natural TP モニタインターフェイスのインストールに関するセクションがあります。TP モニタインターフェイスについては、『TP モニタインターフェイス』ドキュメントを参照してください。一般に注意する必要がある点は以下のとおりです。

- 共有ニュークリアスは、追加のリンクステップによって作成されます。このリンクのターゲットライブラリは、オペレーティングシステムローダーが実行可能モジュールを検索する任意のライブラリでかまいません。テスト目的では、最初に共有ニュークリアスを STEPLIB（または SEARCH チェーン）内の1つのライブラリにリンクしてから、実稼働用の LPA（または SVA）ライブラリにリンクするほうが簡単な場合があります。混乱を避けるために、STEPLIB ライブラリ内のモジュールは、LPA ライブラリにリンクするときに削除する必要があります。
- 使用する共有ニュークリアスの名前は、環境依存部分をインストールするときに Natural パラメータモジュールのプロファイルパラメータ NUCNAME で指定します。NUCNAME は、プライマリパラメータ入力ではダイナミックパラメータとして指定できますが、PROFILE または SYS パラメータ文字列ではできません。

ニュークリアスへのサブ製品のリンク

ほとんどの Software AG サブ製品は、環境非依存 Natural ニュークリアスか環境依存部分にリンクできます。サブ製品のインストール手順を参照してください。

ただし、以下の Natural サブ製品は、環境依存部分にリンクする必要があり、共有ニュークリアスにリンクすることはできません。

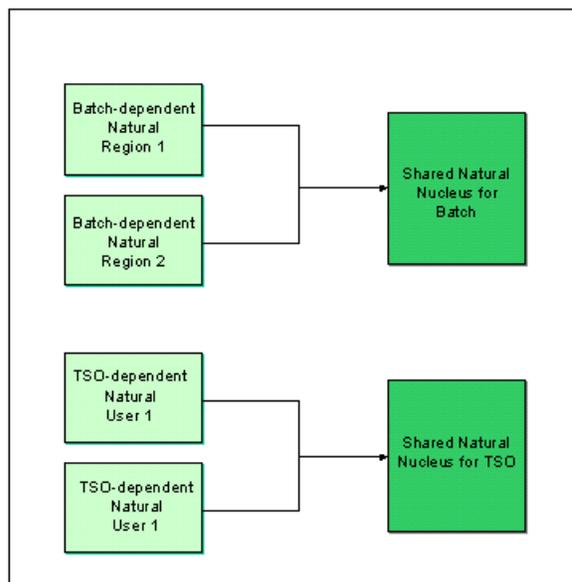
- Adabas リンクルーチン (ADALNK または ADAUSER)

他にいくつかの製品で、個別の部分を共有ニュークリアスおよび環境依存部分にリンクする必要があります。これについては、それぞれのサブ製品で詳細に記載されています。

単一環境の共有ニュークリアス

Natural の一部のサブ製品では、Natural ニュークリアスに TP 固有のモジュールが含まれている必要があります。この場合は、各オペレーティング環境に 1 つの単一環境の共有ニュークリアスを作成する必要があります（例えば、バッチモードに 1 つ、TSO に 1 つ、CICS に 1 つ）。これには、すべてのバッチリジョンやすべての TSO ユーザーが独自の Natural ニュークリアスを共有するという利点があります。

以下の図に、この状況の例を示します。



このコンセプトの単一環境の共有ニュークリアスは必ずインストールされるため、パラメータ SHARED-NUC が Y に設定されていれば、Software AG の System Maintenance Aid (SMA) でこのタイプの共有ニュークリアスが生成されます。

単一環境の共有ニュークリアスがすべて同じであり、TP モニタ固有のモジュールが含まれていない場合は、単一環境の共有ニュークリアスから複数環境の共有ニュークリアスに移行できます。

環境依存のニュークリアス

Natural 共有ニュークリアスの環境非依存部分の他に、単一の Natural リージョンはそれぞれ 1 つまたは複数の環境依存モジュールを実行します。環境依存モジュールは、実際の環境（Complete、CICS、IMS/TM、TSO、またはバッチモード）に応じて異なります。環境依存部分は、セッションの開始時に制御を受け取り、Natural ニュークリアスがリンクされているかどうかをチェックします。チェックされていない場合は、共有ニュークリアスがロードまたは検出され、通信が確立されます。

以下のモジュールは、各環境に固有の Natural の依存部分を構築するために、まとめてリンクする必要があります。

- Natural 環境固有インターフェイス（つまり、NCFNUC、NATCICS、NATIMS、NATTSO、または NATOS/NATVSE）と他のインターフェイス関連のモジュール
- 環境固有の Natural パラメータモジュール NATPARM
- マクロ NTINV により内部 CSTATIC リストに定義されたエントリを持つ、または Natural パラメータモジュールで CSTATIC として指定された、Natural サブ製品モジュール
- Natural パラメータモジュールで CSTATIC として定義された、Natural 以外のプログラム
- Complete、TSO、またはバッチモード用のワークファイルおよび出力ファイルモジュール

スタティックにリンクされた Natural 以外のプログラム

Natural パラメータモジュール NATPARM には、スタティックにリンクする Natural 以外のプログラムのリストが含まれています。このリストは、マクロ NTINV で定義された内部部分と CSTATIC パラメータで定義された外部部分で構成されます。リストの各エントリは、プログラム名と、対応するモジュールを Natural パラメータモジュールにリンクして解決する必要がある V 定数で構成されます。

内部リストは、非依存ニュークリアスの NATCONFIG モジュールに常に存在し、パラメータモジュールが非依存ニュークリアスにリンクされていない場合に使用されます。非依存ニュークリアスにスタティックにリンクされた Natural 以外のプログラムがある場合は、これらすべてのプログラムが定義されたパラメータモジュールもリンクされている必要があります。

オプションで、代替パラメータモジュールを PARM パラメータで指定できます。代替パラメータモジュールは、リンクされたパラメータモジュールよりも優先されます。セッション初期化時に、スタティックにリンクされたプログラムのリストが最大 3 つマージされます。このマージの基本リストは、実際のパラメータモジュールのリストです。これは、そのエントリのみが使用されることを意味します。このリストで解決されない V 定数は、代替パラメータモジュールが使用される場合は、環境パラメータモジュールによって解決が試されます。環境パラメータモジュールで解決されない V 定数は、環境非依存ニュークリアスによって解決が試されます。

Natural以外のプログラムが非依存ニュークリアスにスタティックにリンクされている場合は、これは、非依存ニュークリアスにリンクされたパラメータモジュールおよびセッションに実際に使用されるパラメータモジュールで指定される必要があります。

また、スタティックにリンクするように定義された Natural 以外のプログラムを、Natural セッションの初期化中に "ダイナミック" にリンクすることも可能です。詳細については、RCA プロファイルパラメータの説明を参照してください。

ダイナミックに呼び出された Natural 以外のプログラム

Natural 以外のプログラムを、スタティックにリンクする代わりに、Natural CALL ステートメントを使用して実行時にダイナミックに呼び出すこともできます。ただしこの場合は、プログラムを、スタティックにリンクされたと定義することはできません。

CALL ステートメントが実行されると、Natural は、ダイナミックなロードを試み、環境（サブ）システム（例えば、CICS 環境の EXEC CICS LINK）を利用して、オペレーションを呼び出します。

19 Natural ロールサーバーの機能

▪ Natural ロールサーバーの概要	122
▪ 単一の z/OS システムにおけるロールサーバー	122
▪ 並列 Sysplex におけるロールサーバー	124
▪ ロールファイルおよび LRB	125

このドキュメントでは、以下のトピックについて説明します。

「[Natural ロールサーバーの操作](#)」も参照してください。

Natural ロールサーバーの概要

Natural は、Natural ロールサーバーを使用することにより、アドレススペースが複数の z/OS イメージ（並列 Sysplex）に配置されている CICS や IMS/TM などの複数のアドレススペースシステムで実行できます。もちろん、単一の z/OS システムを実行している場合でも、ロールサーバーを使用できます。

Natural は、端末 I/O の実行時にアプリケーションのコンテキストデータ（スレッド）を保存する必要があります。端末 I/O が開始される前に、スレッドがロールサーバーに提供され、ローカルロールバッファ内かロールファイル内に保持されます。端末 I/O が完了すると、Natural はロールサーバーにスレッドを要求し、アプリケーションを続行します。並列 Sysplex では、ロールサーバーはスレッド（ロールファイルディレクトリ）に関する情報をカップリング機能のデータ構造に保持します。このため、Natural アプリケーションは、異なる時間に異なる z/OS システムで実行することができます。スレッドを、1つのシステム上のロールサーバーに提供し、他のシステムから戻すように要求できます。

ロールサーバーは独自のアドレススペースで実行されます。サービスは PC ルーチンとして提供されます。並列 Sysplex では、参加する各 z/OS イメージでロールサーバーの1つのインスタンスを開始する必要があります。

適用されるロールサーバー Zap のリストは、Natural コマンド DUMP ZAPS で表示できます。また、適用される Zap のリストは、ロールサーバーの起動中に JESMSG LG に書き込まれます。

CICS 環境での Natural に関する注意事項：CICS システムリカバリテーブルには、z/OS システムアベンドコード OD6 が含まれている必要があります。

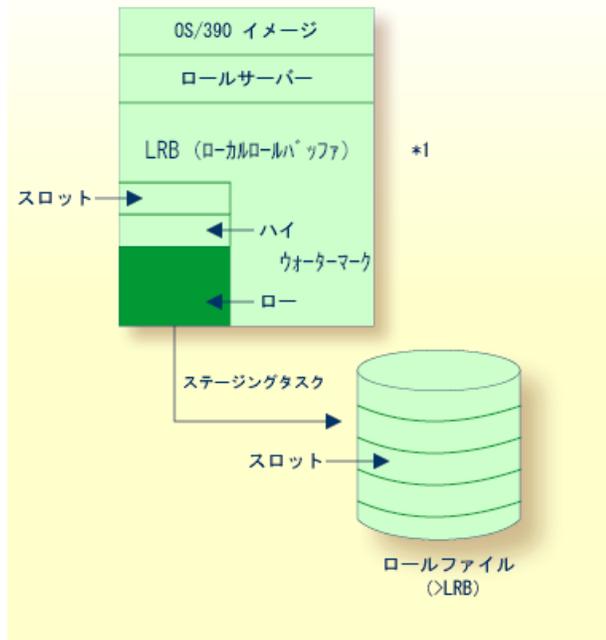
単一の z/OS システムにおけるロールサーバー

ロールサーバーは、書き込み要求でスレッドを受け取ると（端末出力の前）、ローカルロールバッファ（LRB）に十分なスペースがあるかどうかを確認します。ある場合は、スレッドが LRB にコピーされます。ない場合は、スレッドがロールファイルに書き込まれます。また、スレッドは、LRB スロットサイズより大きい場合も、ロールファイルに書き込まれます。スレッドがロールファイルのスロットサイズより大きい場合は、スレッドに対応するために、追加のオーバーフロースロットが割り当てられます。オーバーフロースロットの割り当ては、Natural セッションが最初に割り当てられたロールファイルに制限されます。必要なオーバーフロースロットを割り当てられるフリースペースがロールファイルにない場合は、エラーが生成され、Natural セッションの要求は異常終了します。オーバーフロースロットは、小さなスレッドで後続の書き込み要求によって暗黙的に解放されます。

ロールサーバーは、スレッドの読み込み要求を受け取ると（端末入力の後）、LRB でスレッドを検索します。見つかった場合は、スレッドが LRB から要求者のアドレススペースにコピーされます。見つからない場合は、スレッドがロールファイルから読み込まれて要求者のアドレススペースにコピーされます。

システムパフォーマンスが良好で LRB に常に十分なスペースがあることを確実にするために、「最高／最低値」があります。LRB の最高値に達すると、ステージングタスクが有効になり、最低値に達するまで LRB コンテンツがロールファイルにコピーされます。そのため、最高値と最低値をどこに設置するかは、パフォーマンスチューニングにおいて重要です。パフォーマンスチューニングの詳細については、「ロールサーバーのパフォーマンスチューニング」を参照してください。

単一の z/OS システムにおけるロールサーバーの図：



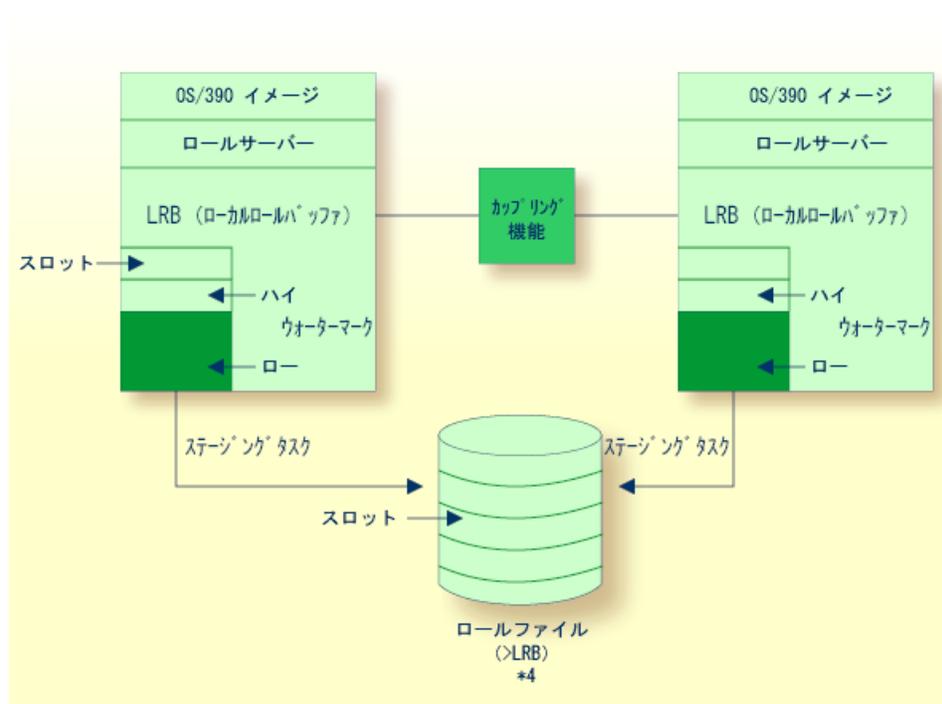
並列 Sysplex におけるロールサーバー

並列 Sysplex では、参加する z/OS イメージ内のロールサーバーはカップリング機能 (CF) の XCF シグナルサービス経由で通信し、ロールファイルディレクトリは XES データ構造に存在します。

ロールサーバーは、書き込み要求でスレッドを受け取ると (端末出力の前)、ローカルロールバッファ (LRB) に十分なスペースがあるかどうかを確認します。十分なスペースがある場合は、スレッドが LRB にコピーされて LRB からロールファイルに非同期的に書き込まれます。LRB に十分なスペースがない場合は、スレッドがロールファイルに直接書き込まれます。CF 構造内のロールファイルディレクトリは、それに従って更新されます。スレッドオーバーフローは、「[単一の z/OS システムにおけるロールサーバー](#)」に記述されているように処理されます。

ロールサーバーは、スレッドの読み込み要求を受け取ると (端末入力の後)、最後の書き込み要求が同じ z/OS イメージで発行された場合は、スレッドを LRB から要求者のアドレススペースに直接コピーします。最後の書き込み要求が同じ z/OS イメージから発行されていない場合は、スレッドがロールファイルから読み込まれて要求者のアドレススペースにコピーされます。

並列 Sysplex におけるロールサーバーの図：



ロールファイルおよび LRB

ロールファイルは、ディレクトリと固定長スロットに論理的に細分化された BDAM ファイルです。スロットサイズは、ロールファイルフォーマットルーチン NATRSRFI のパラメータで決まります。スロットは、予想される最大の圧縮 Natural スレッドより大きい必要があります。

ロールファイルディレクトリには、アクティブな Natural セッションごとに1つのエントリと、最後の書き込み要求のタイムスタンプが含まれます。単一の z/OS システムでは、ディレクトリはロールサーバーのアドレススペースに存在します。並列 Sysplex では、ディレクトリはカップリング機能に存在します。ディレクトリは、ロールサーバーが終了するときリソースの割り当てを解除するときのみロールファイルに書き戻されます。「[Natural ロールサーバーの操作](#)」の DEAL および TERM コマンドを参照してください。

ローカルロールバッファは、データスペースまたは z/OS メモリオブジェクトに含まれ、固定長スロットに細分化されています。LRB スロットはロールファイルスロットより小さくてもかまいません。スレッドが LRB スロットサイズより大きい場合は、スレッドはロールファイルに直接書き込まれます。LRB スロットの数とサイズは、ロールサーバーの起動パラメータで決まります。これらは、システムパフォーマンスにおいて重要な要素です。

ロールサーバーは、最大5つの異なるロールファイルで実行できます。これらのロールファイルはそれぞれ1つのローカルロールバッファに論理的に接続されます。ロールファイルが5つある場合は、対応する LRB が5つあります。各ロールファイルは、専用の読み取り、書き込

み、およびステージングタスクによってアクセスされます。このため、複数チャネルの複数ディスクにロールファイルが作成された場合は、ロールファイルに同時にアクセスできます。

Natural ユーザーは、以下のアルゴリズムに応じてロールファイルに割り当てられます。

```
RN := (first four bytes of roll-server-user-id interpreted as positive integer)
modulo number of roll files + 1
ALLOCNUM := 0
FOR I = RN TO number of allocated roll files
  IF ROLLFILE(I) is not full THEN
    ALLOCNUM := I
    ESCAPE BOTTOM
  END-IF
END-FOR
IF ALLOCNUM = 0 THEN
  FOR I = 1 TO RN-1
    IF ROLLFILE(I) is not full
      ALLOCNUM := I
      ESCAPE BOTTOM
    END-IF
  END-FOR
END-IF
IF ALLOCNUM = 0 THEN
  Return 'roll file full' error
ELSE
  Allocate userid to roll file number ALLOCNUM
END-IF
```

roll-server-user-id は、Natural インターフェイスによって提供される 16 バイトの一意な文字列です。詳細については、このドキュメント内の対応する TP モニタインターフェイスのセクションを参照してください。

例：

- ロールファイルが 5 つあり、*roll-server-user-id* は UF です。ロールファイル 2 はいっぱいですが、ロールファイル 3 には使用可能な空きスロットがあります。

```
E4C64040 - 80000000 = 64C64040
64C64040 modulo 5 = 1
```

- 最初にロールファイル 2 で空きスロットがチェックされます。チェックに失敗し、次にロールファイル 3 がチェックされて空きスロットが見つかります。
- そのため、ユーザー UF はロールファイル 3 に割り当てられます。

このアルゴリズムでユーザー ID がロールファイル間で均等に分散されない場合は、ロールサーバーのユーザー出口 NATRSU14 が役立ちます。これは、ロールサーバーユーザー ID の最初の 8 バイトにはサーバー名が挿入されるため、サーバー環境に特に関連します（「[z/OS 環境のサー](#)

バーとしての*Natural*」を参照)。ユーザー出口の詳細については、「*Natural* ロールサーバーの操作」の「*NATRSU14* ユーザー出口」を参照してください。

ユーザー ID が均等に分散されていることを確認するには、Natural コマンド SYSTP、選択 "R" を使用して、ロールサーバー統計を表示します。

20 Natural ロールサーバーの操作

■ ロールサーバーシステム要件	130
■ ロールファイルのフォーマット	132
■ ロールサーバーの開始	135
■ ロールサーバーメッセージ、コンディションコード、およびアベンドコード	138
■ ロールサーバー要求のリターンコードおよび理由コード	139
■ ロールサーバーの操作	140
■ ロールサーバーのパフォーマンスチューニング	141
■ ロールサーバーのユーザー出口	142

このドキュメントでは、以下のトピックについて説明します。

「[Natural ロールサーバーの機能](#)」も参照してください。

ロールサーバーシステム要件

このセクションでは、ロールサーバーシステム要件について説明します。

以下のトピックについて説明します。

- APF 認可
- システムリンケージインデックス
- 仮想ストレージ
- CF 構造
- XCF シグナルパス

APF 認可

IEWL パラメータ AC(1) を指定して、モジュール NATRSMvr (vr=バージョンとリリース番号) を Authorized Program Facility (APF) ライブラリにリンクします。「[z/OS 環境での Natural のインストール手順](#)」を参照してください。

システムリンケージインデックス

ロールサーバーで1つのシステムリンケージインデックス (システム LX) を確保するため、ライブラリ SYS1.PARMLIB のメンバ IEASYSxx に十分大きい値の NSYSLX があることをチェックしてください。

ロールサーバーが終了すると、システム LX が使用されているので、アドレススペース ID は使用できなくなります。これは、次の IPL で再度使用可能になります。

これを回避するには、DEAL オペレータコマンドを使用してロールサーバーを非アクティブにし、後で再起動します。

確保されたシステム LX は、次の IPL まで、ロールサーバーが再起動されるたびに再使用されます。

仮想ストレージ

ストレージ	サイズ
ECSA	84 バイト
プライベートプログラムストレージ	30 KB 以上
固定サブプールストレージ (ELSQA を含む)	10 KB 未満、50 KB 以上
LRB ディレクトリ	32 + (64 * LRB スロット数)
ロールファイルごとに 100 スロット	4 KB 以上
各追加ロールファイル	30 KB 以上
ワーキングストレージ	ロードに応じて約 1 MB 以上

CF 構造

必要なスペースは、以下の式で計算します。

$$24 \text{ KB} + (\text{RFN} + 1) * 1 \text{ KB} + (\text{RFS} + 8) * 160 \text{ bytes}$$

RFN はロールファイル数、RFS はすべてのロールファイル内のロールファイルスロット合計数です。

例：

- それぞれ 1000 スロットを持つ 5 つのロールファイルがあります。

$$24 \text{ KB} + 6 \text{ KB} + 5008 * 160 \text{ bytes} = 24 \text{ KB} + 6 \text{ KB} + 783 \text{ KB} = 813 \text{ KB}$$

- この結果、CF 構造は 820 KB で定義する必要があります。

```
STRUCTURE NAME(NATROLLS) SIZE(820) PREFLIST(CF1)
```

XCF シグナルパス

並列 Sysplex では、ロールサーバーは XCF シグナルサービス経由で通信します。デフォルトの XCF グループ名として、CF 構造名の左端の 8 バイトが使用されます。

独自の XCF グループ名を指定する場合は、NATRSU24 ユーザー出口を使用します。このユーザー出口の詳細については、「[NATRSU24 ユーザー出口](#)」を参照してください。

ロールファイルのフォーマット

ロールファイルをフォーマットするには、次の手順に従います。

1. ロールファイルを固定レコードフォーマットの物理シーケンシャルデータセットとして割り当てます。
2. これを、モジュール NATRSRFI を使用してフォーマットします。

フォーマット中に、ロールファイルはデバイス依存のブロックサイズで BDAM フォーマットに変換されます。

 **注意:** 以前のバージョンの既存のロールファイルを使用する予定の場合は、NATRSRFI RESET 機能を実行するだけで十分です。

フォーマットするには、以下のパラメータ文字列を DD 名 RFIPARMS に入力するか JCL EXEC ステートメントで PARM として入力します。

```
function,dd-name,slot-size,number-of-slots
```

すべてのパラメータは位置指定です。詳細については、以下の表を参照してください。

パラメータ	説明	
<i>function</i>	FORMAT	ロールファイルをフォーマットします。
	RESET	すべてのロールファイルスロットがリセットされます (空きスロットとしてマークされます)。このパラメータ値は、ロールファイルがすでにフォーマットされている場合にのみ使用できます。 他に使用できるパラメータは <i>dd-name</i> のみです。
	LIST	ロールファイルに含まれるセッションIDとその最後のアクティビティのリストを出力します。 他に使用できるパラメータは <i>dd-name</i> のみです。
<i>dd-name</i>	ロールファイルが指定されている DD ステートメントの名前。	
<i>slot-size</i>	ロールファイルスロットのサイズ (バイト単位)。このサイズは、使用ブロックサイズの倍数に切り上げられます。 最初は、Natural スレッドのサイズと等しいスロットサイズを使用することをお勧めします。次に、ロールサーバー統計を確認します。スレッドサイズの最大のおカレンスも表示されます。必要であれば、この値を使用してスロットサイズを減らします。	
<i>number-of-slots</i>	割り当て対象のロールファイルスロットの数。この数は、同時にアクティブになれるユーザーの最大値です。	

パラメータ	説明
	このパラメータはオプションです。省略すると、ロールファイル全体が割り当てどおりにフォーマットされます。

ロールファイル用のシリンダに必要なディスクスペース（DD ステートメントの SPACE パラメータ）を計算するには、以下の式を使用します。

$$\text{number-of-cylinders} = \text{ceiling} (\text{number-of-slots} * \text{slot-size} / 30 * \text{block-size})$$

トラックの場合は、以下の指揮を使用します。

$$\text{number-of-tracks} = \text{ceiling} (\text{number-of-slots} * \text{slot-size} / 2 * \text{block-size})$$

使用されるブロックサイズは以下のとおりです。

3380 DASD : 23476

3390 DASD : 27998

9345 DASD : 22928

また、ロールファイルディレクトリヘッダ（40 バイト）用のスペースと、各ロールファイルスロット用の1つのディレクトリエントリ（24 バイト）が必要です。このため、1つの追加のブロックが、3380 上で約 976 スロット、3390 上で 1164 スロット、9345 DASD 上で 953 スロット必要です。

NATRSRFI 出力

DD ステートメントで DD 名 RFIPRINT が指定されると、NATRSRFI の出力がこのデータセットにダイレクトされます。RFIPRINT が省略されると、出力は、WTO マクロ（ROUTDCE=11）を使用して JESMSG LG に書き込まれます。RFIPRINT は LIST 機能用に指定する必要があるので注意してください。

NATRSRFI コンディションコードおよびアベンドコード：

以下のコンディションコードが使用されます。

0	正常終了。
4	フォーマットされたスロットの数が、要求された数を下回っています。
20	パラメータエラー。

以下のユーザーアベンドコードが使用されます。

アベンドコード	原因
U0100	RFIPARMS または RFIPRINT の OPEN に失敗しました。
U0101	ロールファイルの OPEN に失敗しました。

例 1 :

```
//FBRUNRFI JOB (FB,218),FB,CLASS=K,MSGCLASS=X,NOTIFY=FB
//FORMAT EXEC PGM=NATRSRFI
//STEPLIB DD DISP=SHR,DSN=NATURAL.NATvr.LOAD
//RF1 DD DISP=SHR,DSN=FB.SYSF.ROLLF1
//RF2 DD DISP=SHR,DSN=FB.SYSF.ROLLF2
//RFIPARMS DD *
FORMAT,RF1,200000,1000
FORMAT,RF2,200000
/*

Exerpt from resulting JESMSGLG:
+FBRUNRFI: FORMAT,RF1,200000,1000
+FBRUNRFI: RF1: FB.SYSF.ROLLF1
+FBRUNRFI: Creation date: 2001/06/13 Volume: ADA002(3390)
IEC031I D37-04,IFG0554P,FBRUNRFI,FORMAT,RF1,305B,ADA002,FB.SYSF.ROLLF1
+FBRUNRFI: Not enough space for 1000 slots.
+FBRUNRFI: 60 Blocks written. Block size is 27998.
+FBRUNRFI: 1 Directory block.
+FBRUNRFI: 8 Blocks per slot. Slot size is 223984.
+FBRUNRFI: 7 Slots initialized. Roll file version vrs.
+FBRUNRFI: 3 Blocks unused.
+FBRUNRFI: FORMAT,RF2,200000
+FBRUNRFI: RF2: FB.SYSF.ROLLF2
+FBRUNRFI: Creation date: 2001/06/08 Volume: USRF08(3380)
IEC031I D37-04,IFG0554P,FBRUNRFI,FORMAT,RF2,020F,USRF08,FB.SYSF.ROLLF2
+FBRUNRFI: 60 Blocks written. Block size is 23476.
+FBRUNRFI: 1 Directory block.
+FBRUNRFI: 9 Blocks per slot. Slot size is 211284.
+FBRUNRFI: 6 Slots initialized. Roll file version vrs.
+FBRUNRFI: 5 Blocks unused.
```

vrs はバージョン、リリース、システムメンテナンスレベルです。

例 2 :

```
//FBRUNRFI JOB (FB,218),FB,CLASS=K,MSGCLASS=X,NOTIFY=FB
//FORMAT EXEC PGM=NATRSRFI,PARM='FORMAT,RF1,200000'
//STEPLIB DD DISP=SHR,DSN=NATURAL.NATvr.LOAD
//RF1 DD DISP=SHR,DSN=FB.SYSF.ROLLF1
//RFIPRINT DD SYSOUT=X
```

結果の RFIPRINT :

```
Natural Roll Server - Roll File Utility Version
vrs
FORMAT,RF1,200000
RF1: FB.SYSF.ROLLF1
Creation date: YYYY/MM/DD Volume: ADA002(3390)
60 Blocks written. Block size is 27998.
1 Directory block.
8 Blocks per slot. Slot size is 223984.
7 Slots initialized. Roll file version vrs.
3 Blocks unused.
```

ロールファイルのフォーマットまたはリセットに関する注意事項

- RFIPARMS で複数のパラメータ行を指定することにより、複数のロールファイルを1回でフォーマットまたはリセットできます。
- ロールサーバーがアクティブなときにロールファイルをフォーマットまたはリセットすることはできません。
- ロールファイルが並列 Sysplex でフォーマットされている場合は、以下のように SETXCF オペレータコマンドを使用してロールサーバーのカップリング機能構造をクリアする必要もあります。

```
SETXCF FORCE,STR,STRNAME=NATROLL1
```

ロールサーバーの開始

ロールサーバーは、モジュール NATRSMvr を実行して、バッチジョブまたは開始タスクとして開始します。ロールファイルは、DD ステートメントとして ddnameROLLF1~ROLLF5 で定義する必要があります。

JCL EXEC ステートメントで、PARM に以下のパラメータを指定します。

subsystem-id, number-of-roll-files, number-of-LRB-slots, LRB-slot-size, CF-structure-name, low-water-mark, high-water-mark, non-activity-time, timeout-check-time, message-case

すべて位置指定パラメータであり、コンマで区切る必要があります。詳細については、以下の表を参照してください。

パラメータ	指定可能な値	デフォルト	コメント
<i>subsystem-id</i>	空白以外の4バイトの文字列	NATv	指定値は、Natural プロファイルパラメータ SUBSID の値と一致する必要があります (v = vバージョン番号)。 注意: CICS 環境の Natural の場合は、適切なサブシステム ID を設定するには、NCMDIR マクロの ROLLSRV パラメータを参照してください。
<i>number-of-roll-files</i>	0 - 5	1	並列 Sysplex 以外では、ロールサーバーは、ストレージ内のローカルロールバッファのみを使用してロールファイルなしで動作できます。
<i>number-of-LRB-slots</i>	1 - 32767	なし	LRB スロット数にスロットサイズを掛けた値は、2 GB を超えることはできません。 同じ数の LRB スロットが、各 LRB (つまり、使用される各ロールファイル) に割り当てられます。LRB スロットの合計数は、以下の式で計算します。 <i>number-of-roll-files * number-of-LRB-slots</i>
<i>LRB-slot-size</i>	任意の数値	ロールファイルのスロットサイズ	バイト数の値。 ロールファイルを使用しない場合は、このパラメータを指定する必要があります。 ロールファイルを使用する場合は、このパラメータは無視され、代わりにロールファイルのスロットサイズが使用されます。
<i>CF-structure-name</i>	任意の有効な構造名	なし	16 文字未満の文字を指定すると、空白が追加されます。 このパラメータは (並列 Sysplex で) カップリング機能を使用する場合にのみ指定します。
<i>low-water-mark</i>	1 - 9	7	最低値を 10 % 刻みの LRB スロット数で指定します。 このパラメータは並列 Sysplex では無視されます。
<i>high-water-mark</i>	1 - 10	8	最低値の場合と同様です。 "10" の値を指定すると、ステージタスクが有効になりません。"10" を指定するのは、LRB が、同時にアク

パラメータ	指定可能な値	デフォルト	コメント
			<p>タイプになっているすべての Natural セッションに対応できる十分な大きさの場合のみにすることをお勧めします。</p> <p>このパラメータは並列 Sysplex では無視されます。</p>
<i>non-activity-time</i>	1 - 999999	なし	<p>セッションが非アクティブになってからロールファイルから削除されるまでの時間数。</p> <p>この時間を超えると、セッションはスケジュールされた次のタイムアウトチェック時に削除されます。</p> <p>このパラメータを省略すると、タイムアウトチェックは実行されません。</p> <p>このパラメータを変更するには、オペレータコマンド TIMEOUT を使用します (以下参照)。</p>
<i>timeout-check-time</i>	0000 - 2359	なし	<p>タイムアウトチェックが実行される時刻。</p> <p>セッションは、先行するパラメータで指定された非アクティビティ時間より長く非アクティブになっていると削除されます。</p> <p>このパラメータを省略すると、タイムアウトチェックはスケジュールされません。</p> <p>このパラメータを変更するには、オペレータコマンド TIMEOUT を使用します (以下参照)。</p>
<i>message-case</i>	UCTRAN または空白	空白	<p>ロールサーバーですべてのメッセージを大文字で発行する場合は、UCTRAN を指定します。</p>



注意: ローカルロールバッファは、「バーの上」のメモリオブジェクトに配置されます。十分なメモリが「バーの上」に割り当てられるように、EXEC ステートメントで MEMLIMIT パラメータを使用します。

ロールサーバーをバッチジョブとして開始する例

```
// EXEC PGM=NATRSMvr,PARM='NAvr,,1000'
//ROLLF1 DD DSN=SYSF.ROLLFILE
```

サブシステムIDは、NA *vr*で、1つのロールファイルが使用され（デフォルト）、ローカルロールバッファには1000のロットが含まれます。使用されるロットサイズはロールファイルのロットサイズと同じです。最低値は70%（デフォルト）で、最高値は80%（デフォルト）です。

```
// EXEC PGM=NATRSMvr,PARM=',5,1000,150000,NATROLL1',MEMLIMIT=800M
//ROLLF1 DD DSN=DASD1.ROLLFILE
//ROLLF2 DD DSN=DASD2.ROLLFILE
//ROLLF3 DD DSN=DASD3.ROLLFILE
//ROLLF4 DD DSN=DASD4.ROLLFILE
//ROLLF5 DD DSN=DASD5.ROLLFILE
```

サブシステムIDはNAT*v*（デフォルト）で、5つのロールファイルが使用され、5つのローカルロールバッファにはそれぞれ1000のロットが含まれます。LRBロットサイズは150000バイトです。ロールファイルディレクトリはカップリング機能構造NATROLL1にあります。すべてのスレッドがロールファイルに書き込まれるため、最低値と最高値は無視されます（「[Natural ロールサーバーの機能](#)」を参照）。このジョブはz/OS向けであるため、MEMLIMITオプションはローカルロールバッファ用に800MBを指定します。

 **注意:** ロールサーバーは、以下の場合は開始されません。

- 別のロールサーバーが同じ *subsystem-id* で実行されている。
- 別のロールサーバーが、JCL で指定されているロールファイルにアクセスしている。
- ロールファイルが、CF構造をリセットせずにSETXCF FORCEコマンドを使用して再フォーマットされている。

ロールサーバーメッセージ、コンディションコード、およびアベンドコード

ロールサーバーは、WTOマクロ（ROUTCDE=11）を使用して、情報およびエラーメッセージをJESMSG LGに書き込みます。メッセージの先頭には、メッセージIDとロールサーバーのジョブ名が付きます。例えば、以下のようになります。

```
RSM0019 FBRSMvrs: Roll Server Version vrs is active
```

メッセージについては、『メッセージおよびコード』ドキュメントの「ロールサーバーメッセージ」を参照してください。

ロールサーバーにより開始されたタスクのコンディションコード

以下のコンディションコードが使用されます。

0	正常終了
12	不適切なパラメータの入力
16	ランタイムエラー
20	アベンドの発生
>100	初期化エラー

ユーザーアベンドコード

XCF または XES サービス呼び出しにより予期しないリターンコードが発行されると、ダンプ付きのアベンドが強制されます。アベンドレジスタのレジスタ14に理由コードが含まれます。理由の詳細説明については、IBM の『MVS プログラミング：シスプレックス・サービス解説書』を参照してください。エラーが環境固有ではない場合は、ダンプを Software AG サポートに送信してください。

以下のユーザーアベンドコードが使用されます。

アベンドコード	原因
U0200	IXLCONN が失敗
U0201	IXLFORCE が失敗
U0202	IXLLIST が失敗
U0203	IXLDISC が失敗
U0204	IXCLEAVE が失敗
U0301	IXLLIST が失敗
U0302	IXCMGO が失敗
U0401	IXLLIST が失敗
U0501	IXLLIST が失敗

ロールサーバー要求のリターンコードおよび理由コード

ロールサーバー要求のリターンコードおよび理由コードは、Natural がロールサーバーの PC サービスルーチンから受け取るコードです。これらは、それぞれの TP インターフェイス（Natural CICS または Natural IMS インターフェイス）により報告されます。これらのコードのリストについては、『メッセージおよびコード』ドキュメントの「ロールサーバー要求のリターンコードおよび理由コード」を参照してください。

ロールサーバーの操作

以下のコマンドは、MODIFY オペレータコマンドを使用してロールサーバーに渡すことができます。

コマンド	説明	
DEAL	ロールサーバーが停止しましたが、アドレススペースが終了していません。ロールファイルディレクトリと変更されたすべての LRB スロットはロールファイルに書き込まれます。割り当て解除ステータスの場合は、ロールサーバーを新しいパラメータと古いアドレススペース ID で再起動できます。ロールファイルは、割り当て解除ステータスで再フォーマットできます。ただし、これを行うと、現在アクティブな Natural セッションを再度開始することはできなくなります。	
DIAGNOSE	デバッグ機能です。Software AG から推奨された場合にのみ使用します。 このコマンドには機能はありません。将来的には、特別な Zap と組み合わせて必要に応じて特定の顧客の問題を診断するのに利用する予定です。	
SNAP	デバッグ機能です。ロールサーバーのアドレススペースが SYSUDUMP にダンプされます。	
START,parmstring	ロールサーバーを、指定されたパラメータで再度アクティブにします。このコマンドは、非アクティブなステータスでのみ使用できます。	
STATS	ロールサーバー統計を JESMSGLG に書き込みます。	
TERM	ロールサーバーを停止します。ロールファイルディレクトリと変更されたすべての LRB スロットはロールファイルに書き込まれ、アドレススペースは終了されます。アドレススペース ID は次回の IPL まで使用できなくなります。	
TIMEOUT	NAT <i>nnn</i>	非アクティビティ時間のパラメータを指定または置き換えます。
	TOC <i>hhmm</i>	タイムアウトチェックの時刻を指定または置き換えます。
	OFF	タイムアウトチェックを無効にします。
	ON	タイムアウトチェックを再度有効にします。
	NOW	直ちにタイムアウトチェックを開始します。通常のタイムアウトチェックのスケジュール（指定されている場合）は有効なままになります。
	? (または指定なし)	現在のタイムアウト設定を表示します。疑問符 (?) は任意であり、省略することができます。
TRSTART	デバッグ機能です。Software AG から推奨された場合にのみ使用します。トレースタスクを有効にします。General Trace Facility (GTF) が開始され、タイプ 200 のユーザーレコードに対して有効になっている場合は、トレースレコードが GTF に書き込まれます。	
TRSTOP	トレースタスクを無効にします。	

ロールサーバーのパフォーマンスチューニング

原則として、ロールサーバーのパフォーマンスチューニングでは、ロールサーバーのディスク優先度は Natural が実行されるアドレススペースよりも高くします。

パフォーマンスの弱点がどこにあるかを特定するには、SYSTP ユーティリティの *Natural* サブシステムおよびロールサーバー情報機能を使用して、システムパフォーマンスを分析します。

ロールサーバー統計を確認するときは、以下の値に特に注意します。

■ 直接書き込みの数

"直接書き込み"とは、受け取られた Natural スレッドがロールファイルに直接書き込まれたことを意味します。

考えられる理由は以下の2つがあります。

1. 使用できる LRB スロットがありません。LRB を増やしてください。
2. 圧縮スレッドが単一の LRB スロットより大きいです。LRB スロットサイズを増やしてください。

■ 直接読み込みの数

「直接読み込み」とは、要求されたスレッドが LRB になかったため、ロールファイルから直接読み込む必要があったことを意味します。

単一の z/OS システムで読み込み合計数に対する直接読み込みの割合が非常に高い場合は、LRB が小さすぎます (LRB を増やしてください)。

並列 Sysplex z/OS システムで読み込み合計数に対する直接読み込みの割合が非常に高い場合は、システム間アクティビティが多数あることを示していることもあります。これは、Natural セッションが存続期間中に非常に頻繁に z/OS イメージを変更していることを意味します。

■ ステージ待機の数 (単一の z/OS 環境内)

「ステージ待機」とは、ステージタスクが LRB スロットをロールファイルに書き込み終わるまで書き込み要求が待機しなければならない状況のことです。書き込み要求合計数に対するステージ待機の割合が非常に高い場合は、最高値および最低値の設定が不適切であるか、ロールファイルデバイス/ロールファイルチャンネルにボトルネックがあります。

ストレステストの経験に基づき、以下をお勧めします。

LRB スロット数に対するアクティブユーザーの最大数の割合が非常に低い場合は、最高値を増やします。そうでない場合は、最高値を減らします。

最高値と最低値の差は、3 (30 %) 以下にします。

理想的には、LRB スロット数が同時ユーザーの最大数より確実に大きい場合は、最高値を 10 に設定します。

ロールサーバーのユーザー出口

ロールサーバーには、2つのユーザー出口があります。

- NATRSU14

- NATRSU24

これらのサンプルのソースモジュールが用意されています。

NATRSU14 ユーザー出口

使用するロールファイル数を指定します。

エン트리呼び出し規則：

- レジスタ 1 は、以下の DSECT で記述されているパラメータリストをアドレスします。

```

PLIST      DSECT
PLRSVER    DS CL4    Roll server version (= 'vrs')
PLNRF      DS H      Number of roll files
PLUID      DS CL16   Userid
PLTNUM1    DS H      Total number of slots Roll file 1
PLUSNUM1   DS H      Number of slots in use Roll file 1
PLTNUM2    DS H      Total number of slots Roll file 2
PLUSNUM2   DS H      Number of slots in use Roll file 2
PLTNUM3    DS H      Total number of slots Roll file 3
PLUSNUM3   DS H      Number of slots in use Roll file 3
PLTNUM4    DS H      Total number of slots Roll file 4
PLUSNUM4   DS H      Number of slots in use Roll file 4
PLTNUM5    DS H      Total number of slots Roll file 5
PLUSNUM5   DS H      Number of slots in use Roll file 5
PLISTL    EQU *-PLIST

```

vrs は、現在の Natural バージョン、リリース、およびシステムメンテナンスレベルを示します。

- レジスタ 13 は、36 フルワードセーブエリアをポイントします。
- レジスタ 14 には、リターンアドレスが含まれます。
- レジスタ 15 には、NATRSU14 のエントリアドレスが含まれます。

リターン呼び出し規則：

- レジスタ 15 には、バイナリフォーマットのロールファイルの数が含まれます。



注意: アクセスレジスタがこのユーザー出口内で変更された場合は、これらのアクセスレジスタはリターン時に保存および復元される必要があります。このユーザー出口は、PSW キー 8 を使用してプライマリアドレッシングモードで呼び出されます。これはクロスメモリモードで実行されるため、SVC 13 以外の SVC は使用できません。

NATRSU24 ユーザー出口

使用する XCF グループ名を指定します。

エントリ呼び出し規則：

- レジスタ 1 は、グループ名が生成される 8 バイトのエリアをポイントします。
- レジスタ 13 は、18 フルワードセーブエリアをポイントします。
- レジスタ 14 には、リターンアドレスが含まれます。
- レジスタ 15 には、NATRSU24 のエントリアドレスが含まれます。

デフォルトのグループ名として、ロールサーバーは CF 構造名の左端の 8 バイトを使用します。

このユーザー出口は、プライマリモード、PSW キー 8、およびタスクモードで呼び出されます。

21 z/VSE 環境

このドキュメントでは、z/VSE オペレーティングシステム環境での Natural について説明します。

- **z/VSE 環境下の Natural** z/VSE 環境で Natural をオンラインまたはバッチモードで実行する際の特別な考慮事項について説明します。
- **z/VSE 環境での Natural 共有ニュークリアス** Natural 共有ニュークリアスの機能と使用方法について説明します。

22 z/VSE 環境下の Natural

▪ Natural サブシステム	148
▪ Natural 共有ニュークリアス	148
▪ TP モニタインターフェイス	148
▪ データベース管理システムへのインターフェイス	148
▪ z/VSE 環境下のバッチモードの Natural	149

このドキュメントでは、Natural を z/VSE 環境で実行する際の特別な考慮事項について概要を説明します。

Natural サブシステム

z/VSE 環境の Natural サブシステムは、以下のコンポーネントで構成されます。

- 1 つまたは複数の [グローバルバッファプール](#)

Natural サブシステムは、Natural プロファイルパラメータ SUBSID および上記コンポーネントの対応する起動パラメータによって識別されます。デフォルトのサブシステム名は NATV です。V は製品のバージョン番号です。

Natural 共有ニュークリアス

Natural 共有ニュークリアスの利点については、「[z/OS および z/VSE 環境下の Natural 共有ニュークリアス](#)」を参照してください。

TP モニタインターフェイス

z/VSE 環境の Natural で使用できる TP モニタインターフェイスについては、以下のセクションを参照してください。

- Com-plete/SMARTS 環境での Natural
- CICS 環境での Natural - 概要

これらのセクションは、Natural の『TP モニタインターフェイス』ドキュメントにあります。

データベース管理システムへのインターフェイス

Software AG のデータベース管理システムである Adabas を除き、データベース対話が必要なすべての操作は対応する Natural インターフェイスモジュールで実行されます。

z/VSE 環境の Natural で使用できるデータベースのインターフェイスについては、以下のドキュメントを参照してください。

- Natural for DB/2
- Natural for VSAM

- Natural for DL/I

z/VSE 環境下のバッチモードの Natural

「バッチモードの *Natural* (すべての環境)」および「z/VSE 環境下のバッチモードの *Natural*」を参照してください。

23 z/OS および z/VSE 環境下の Natural 共有ニュークリ アス

共有ニュークリアスの機能および使用は、オペレーティングシステムが z/OS でも z/VSE でもほとんど同じです。

重複した記述を防ぐため、z/OS 環境の対応するドキュメントを参照してください。以下のトピックについて説明しています。

- [環境に依存しないニュークリアス](#)
- [共有ニュークリアスの作成](#)
- [共有ニュークリアスのインストール](#)
- [ニュークリアスへのサブ製品のリンク](#)
- [単一環境の共有ニュークリアス](#)
- [環境依存のニュークリアス](#)
- [スタティックにリンクされた *Natural* 以外のプログラム](#)
- [ダイナミックに呼び出された *Natural* 以外のプログラム](#)

24 VM/CMS 環境

このドキュメントでは、VM/CMS 環境の Natural について説明します。

- **VM/CMS 環境の Natural** Natural からの CP および CMS コマンドの発行、CMS プログラムスタックの読み込み、ハードコピー機能、Natural への修正の適用などについて説明します。また、CMS 環境で Natural をバッチモードで使用するときの該当トピックへのリンクも示します。
- **出力ファイルおよびワークファイルのサポート** 出力ファイルおよびワークファイルを Natural パラメータモジュールで定義する方法について説明します。

25 VM/CMS 環境の Natural

▪ Natural からの CP および CMS コマンドの発行	156
▪ CMS プログラムスタックの読み込み	156
▪ ハードコピー機能	157
▪ Natural への修正の適用	157
▪ CMS 環境下のバッチモードの Natural	157
▪ TCP/IP 通信の使用	158
▪ Rexx からの Natural サブプログラムの呼び出し	158

このドキュメントでは、Natural を VM/CMS 環境で実行する際の特別な考慮事項について説明します。

以下のトピックについて説明します。

Natural からの CP および CMS コマンドの発行

Natural システムコマンド CMS を使用して CP および CMS コマンドを発行できます。例えば、CMS FLIST * DATA B や CMS CP SPOOL PRT * のように指定します。

パラメータを指定せずに CMS を入力すると、CP/CMS コマンドのメニュープロンプトが表示されます。メニューを終了するには、先頭位置にピリオド (.) を入力します。

Natural プログラム内から CP または CMS コマンドを発行するには、以下のステートメントをコーディングします。

```
CALL 'CMS' command rc
```

上記の意味は以下のとおりです。

command 英数字の変数または定数です。

rc CP または CMS コマンドからのリターンコードを受け取る変数（フォーマット/長さ I4）です。

2 番目のパラメータ (*rc*) の指定は任意です。完全なコマンド解決は、通常の CMS 対話型コマンドモードの場合とまったく同様に提供されます。

CMS プログラムスタックの読み込み

CMS プログラムスタックから Natural 変数に 1 行読み込むには、以下のようにコーディングします。

```
CALL 'CMSREAD' line
```

line は英数字の変数です。

プログラムスタックから読み込まれた行は、変数の長さに合うように切り捨てられるか空白で埋められます。

プログラムスタックが空の場合は、CMSREAD は文字列 *EOD* を返します。

ハードコピー機能

Natural CMS インターフェイスのハードコピー機能を有効にするには、以下のパラメータを指定します。

```
HCAM=CMS
```

これは、NATPARM ASSEMBLE で指定するか、Natural の起動時にダイナミックに指定します。

Natural 端末コマンド %H は、出力を仮想プリンタに送信します。%HL を指定すると、NATURAL LISTING A と呼ばれるファイルが作成されます。

Natural への修正の適用

Software AG では、Natural インストールテープの出荷後に見つかった問題を解決するために、修正を Zap 形式で提供しています。

- Zap を適用する前に、必ずファイルのバックアップコピーを作成してください。
- これらの Zap を Natural テキストファイルに適用する際は、NATZAP 機能を使用してください。
- Zap を適用した後は、Natural モジュールと DCSS を再構築する必要があります。そのためには、それぞれ NATBLDM コマンドと NATBLDS コマンドを使用します。

NATZAP の詳細については、Natural インストール ID で「HELP NATZAP」と入力してください。

CMS 環境下のバッチモードの Natural

「[バッチモードの Natural \(すべての環境\)](#)」および「[CMS 環境下のバッチモードの Natural](#)」を参照してください。

TCP/IP 通信の使用

Natural ステートメント `REQUEST DOCUMENT` を使用して、`http` サーバーに接続し、HTML または XML ファイルを取得します。ファイル `TCPIP DATA` には、TCP/IP クライアントプログラムのコンフィグレーション情報が含まれています。このファイルは TCP/IP クライアントミニディスクに常駐し、通常は以下のコマンドでアクセスします。

```
VMLINK TCPIP
```

`REQUEST DOCUMENT` を使用する場合は、Natural を呼び出す Rexx プログラムに以下の行を含めます。

```
EXEC VMLINK TCPIP
```

Rexx からの Natural サブプログラムの呼び出し

Rexx プログラムでは、`CALLNAT` 機能を使用して Natural サブプログラムを実行し、パラメータを渡すことができます。Rexx プログラムは、Natural がアクティブなときに実行する必要があります。例えば、以下のように Natural プログラムで `CALL` ステートメントを使用します。

```
CALL 'CMS' 'EXEC MYREXX'
```

`MYREXX` では、`CALLNAT` 機能を使用して Natural サブプログラムを実行できます。

```
result = callnat('MYNAT',parm1,parm2)
```

`MYNAT` は、Rexx プログラムを呼び出した Natural プログラムと同じ Natural ライブラリ内にある必要があります。`MYNAT` が正常に実行されると、Rexx 変数 `result` には、呼び出されたサブプログラムの名前が 8 文字になるまで空白で埋められて含まれます（この例では、`result='MYNAT '`）。エラーが発生した場合は、`result` には、接頭辞「`NAT`」の付いた Natural エラー番号（`NAT0082` など）が含まれます。

Rexx と Natural の間でデータを交換するには、以下の指定の後で、ステートメント READ WORK FILE と WRITE WORK FILE を使用します。

```
DEFINE WORK FILE n 'STEM rexxstem.'
```

ワークファイルは、パラメータマクロ NETWORK、または AM=CMS を指定したプロファイルパラメータ WORK を使用して宣言する必要があります。「[出力ファイルおよびワークファイルのサポート](#)」も参照してください。

26 出力ファイルおよびワークファイルのサポート

- 出力ファイルおよびワークファイルの定義 162
- アクセスメソッド STD 162
- アクセスメソッド CMS 162

このドキュメントでは、VM/CMS 対応 Natural で出力ファイルおよびワークファイルを使用する際の特別な考慮事項について説明します。

以下のトピック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定義

出力ファイルとワークファイルは、マクロ NTPRINT および NTWORK を使用して Natural パラメータモジュールで定義します。対応するダイナミックパラメータは PRINT と WORK です。

以下で、サブパラメータ AM (アクセスメソッド) および DEST (出力先) について説明します。これらは両方とも NTPRINT と NTWORK で使用できます。

Natural/CMS では、出力ファイルおよびワークファイルの両方に2つのアクセスメソッド「STD」と「CMS」を使用できます。

アクセスメソッド STD

(AM=STD)

このアクセスメソッドでは、z/OS QSAM アクセスメソッドの CMS シミュレーションを使用します。AM=STD は、テープやスプール (RDR、PRT、PUN) ファイルの読み取り/書き込みを行う場合、または z/OS フォーマットのディスクからワークファイルを読み取る場に指定します。

FILEDEF コマンドは、対応する出力ファイルまたはワークファイルを開く前に発行する必要があります。FILEDEF コマンドで使用する DD 名は、サブパラメータ DEST で指定した名前と同じである必要があります。

アクセスメソッド CMS

(AM=CMS)

このアクセスメソッドでは、標準 CMS ファイルシステムを使用して、アクセスしたミニディスクおよび SFS ディレクトリの CMS ファイルの読み取り/書き込みを行います。

結果の CMS ファイルの名前は以下のとおりです。

- 出力ファイルの場合は CMPRT nn
- ワークファイルの場合は CMWKF nn

nn はファイルの番号です。

これらのファイルタイプは、サブパラメータ DEST で指定した名前と同じです。ファイルモードは常に "A1" です。

AM=CMS の場合の特殊な出力先名

DEST=FD	出力先 FD により、CMS ファイルを Natural の出力またはワークファイルに柔軟に割り当てることができます。Natural では、出力先 FD で出力またはワークファイルを開いたときに、それぞれ DD 名 CMPRT nn または CMWKF nn の FILEDEF を検索します (nn は出力またはワークファイル番号)。次に、FILEDEF コマンドで指定されている CMS ファイル ID を使用します。
DEST=LISTING	この DEST 設定は出力ファイルにのみ適用されます。 この出力先を指定すると、出力ファイルは使用可能なフリースペースが最も多い CMS ディスクに書き込まれます。CMS ファイル ID は CMPRT nn LISTING m で、 m はフリースペースが最も多いミニディスクのファイルモードです。プリンタが閉じている場合は、出力ファイルは仮想プリンタに出力されてから削除されます。
DEST=UEX $xxxxx$	この DEST 設定は出力ファイルにのみ適用されます。 UEX で始まる出力先を指定すると、出力ファイルは LISTING が指定された場合と同様に処理されます。また、プリンタが閉じている場合は、この名前の CMS コマンドが Natural によって発行されます。CMS コマンド (Rexx プロシージャなど) は、出力ファイルの CMS ファイル ID をパラメータで受け取ります。

例：

例 1：

以下のような FILEDEF と NATPARM 設定が有効であるとします。

```
FILEDEF CMWKF05 CLEAR
FILEDEF CMPRT01 DISK MY REPORT D
FILEDEF CMPRT04 DISK MY REPORT A
```

```
NETWORK (1),AM=CMS,DEST=FRED
NETWORK (5),AM=CMS,DEST=FD
NETWORK (6),AM=CMS,DEST=PAUL
NTPRINT (1,4),AM=CMS,DEST=FD
NTPRINT (2),AM=CMS,DEST=LISTING
NTPRINT (5),AM=CMS,DEST=PAUL
```

出力ファイルおよびワークファイルのサポート

この場合は、以下の CMS ファイルが生成されます。

```
CMWKF01 FRED A1  
FILE CMWKF05 A1  
CMWKF06 PAUL A1
```

```
MY REPORT D1  
MY REPORT A1  
CMWKF05 PAUL A1
```

一時ファイル CMPRT02 LISTING *m* は出力されてから削除されます (*m* はフリースペースが最も多いミニディスクのファイルモード)。

例 2 :

```
NTPRINT (1),AM=CMS,DEST=UEXLOCAL
```

これにより、CMS ファイルが生成されます。

```
CMPRT01 UEXLOCAL m
```

CMS コマンド UEXLOCAL がパラメータのファイル ID で発行されます。例えば、この名前の Rexx プロシージャが存在する場合は、このプロシージャがどのプリンタのために `arg fn ft fm` を使用して呼び出されたのかを判別できます。

例 3 :

出力先は、DEFINE WORK FILE ステートメントを使用して動的に定義することもできます。また、DEFINE WORK FILE を使用して、入力元、出力先、Rexx ステムを指定することもできます。

```
DEFINE WORK FILE n 'STEM rexxstem.'
```

入力ファイルを開くと、Natural は値 *rexxstem.0* を使用して読み込むレコードの数を判別します。次に、*rexxstem.1* から *rexxstem.max* (*max* = *rexxstem.0*) までのレコードを読み取ってから、データの終わりを返します。

出力ファイルの場合は、Natural は *rexxstem.1* から *rexxstem.n* までを書き込み、ワークファイルが閉じられると *rexxstem.0* を *n* に設定します。

27 BS2000/OSD 環境

- 関連トピック 166
- BS2000/OSD 専用の他の Natural 機能 167

このドキュメントでは、Natural を BS2000/OSD オペレーティングシステム環境で実行する際の特別な考慮事項について説明します。

- **BS2000/OSD 環境での Natural 共有ニュークリアス** バッチモードや TP モニタ TIAM および UTM 環境で使用可能な共通の Natural 共有ニュークリアスの使用方法について説明します。
- **Natural ロードプールの更新** ロードプール更新プログラムの適用性と使用方法について説明します。
- **メッセージ処理の最適化** Natural で使用する画面出力最適化メソッドと、最新の端末画面内容を復元する機能について説明します。
- **サポートされる Siemens 端末タイプ** BS2000/OSD 環境の Natural でサポートされる各種の Siemens 端末タイプに関する情報を提供します。
- **9750 デバイスのファンクションキーサポート** 9750 タイプの Siemens デバイス用にサポートされている固有の Natural ファンクションキー割り当てについて説明します。
- **共通メモリプール** グローバルおよびローカルの共通メモリプールについて説明します。
- **ダイナミック再ロード可能 3GL プログラムの呼び出し** Natural アプリケーション内でダイナミック再ロード可能 3GL プログラムを呼び出すときのアドレスモードの選択ルールを定義します。
- **出力ファイル/ワークファイルサーバー NATPWSV2** BS2000/OSD での RPC バッチサーバー環境の出力ファイル/ワークファイルサーバー NATPWSV2 について説明します。
- **RPC サーバフロントエンド** BS2000/OSD で出力ファイル/ワークファイルサーバー NATPWSV2 を使用する RPC バッチサーバー環境の RPC サーバフロントエンドについて説明します。

関連トピック

以下の項目も参照してください。

- *Natural* での TP モニタの使用
- *openUTM* 環境での *Natural*
- *TIAM* 環境での *Natural*
- **BS2000/OSD 環境下のバッチモードの *Natural***

BS2000/OSD 専用の他の Natural 機能

Natural では、BS2000/OSD 専用の以下の機能を使用できます。

- **P-Key** ユーティリティ
Siemens 975X 端末でのプログラマブル P キーのロードをサポートします (UTM および TIAM 環境)。
- **スワッププールマネージャ**
Natural スワッププールの使用を制御します (UTM および CICS 環境)。

これらの機能は、Natural ユーティリティ SYSTP の一部です。

28

BS2000/OSD 環境での Natural 共有ニュークリアス

- Natural 共有ニュークリアスの使用ルール 170

このドキュメントでは、バッチモードやTPモニタTIAMおよびUTM環境で使用可能なNatural共有ニュークリアスを使用する際に適用されるルールについて説明します。

以下のトピックについて説明します。

UTMについては、Naturalの『TP モニタインターフェイス』ドキュメントの「複数のアプリケーションと1つの共通Natural」も参照してください。

Natural 共有ニュークリアスの使用ルール

BS2000/OSD 環境での Natural 共有ニュークリアスには、以下のルールが適用されます。

1. Natural 共有ニュークリアスは、バッチやTIAMおよびUTMドライバの対応するリエントラント部分なしでリンクされます。これらのモジュールは、対応するアプリケーションのフロントエンド部分にリンクされる必要があります。

例：Natural 共有ニュークリアスの名前は NATSHARE です。

```
/EXEC $TSOSLNK
MOD NATSHARE,XREF=YES,MAP=Y,XDSEC=Y, SORT=Y
LINK-SYMBOLS *KEEP
INCLUDE NATINV,libname
.
.
INCLUDE NATURAL,libname
.
.
INCLUDE NATLAST,libname
BIND
/SETSW ON=1
LIB NATURAL.USER.MOD,BOTH
PAR O=Y
ADDR *OMF
END
/SETSW OFF=1
```

2. バッチやTIAMおよびUTMアプリケーション固有のNaturalパラメータモジュールも、対応するアプリケーションのフロントエンド部分にリンクされます。また、Natural共有ニュークリアスには、共通のNaturalパラメータモジュール（例えばCSTATICエントリ用）を含めることができます。リンクされるNaturalニュークリアスに選択された名前は、Naturalのロード先であるグローバル共通メモリプールの名前と同じです。この名前は、以下のキーワードパラメータのオペランドとして使用されます。

NUCNAME	マクロ NAMBS2、NAMTIAM、および NATUTM 内
NAME	CMPSTART および ADDON (BS2STUB) 内

例：

```
NRTSTART NAMTIAM CODE=FRONT,
      NUCNAME=NATSHARE
      PARMODE=(31,ABOVE),
      .
      .
NUTFRONT NATUTM APPLNAM=NATUTM,
      .
      .
      NUCNAME=NATSHARE
      PARMODE=(31,ABOVE)
```

3. Natural 共有ニュークリアスは **CMPSTART** プログラムによって起動されます。

例：

```
/EXEC (CMPSTART,NATURAL.MOD)
NAME=NATSHARE,SIZE=2MB,POSI=ABOVE,ADDR=250,SCOP=GLOBAL
PREFIX=YES,LIBR=NATURAL.USER.MOD
```

4. Natural 共有ニュークリアスへのリンクは、**BS2STUB** マクロの生成により、バッチやTIAMまたはUTMアプリケーションで作成されます。「**CMPSTART プログラム**」を参照してください。

例：

```
NRTSTUB BS2STUB PARMOD=31,PROGMOD=ANY
      ADDON NAME=NATSHARE,
      STAT=GLOBAL
NUTSTUB BS2STUB PARMOD=31,PROGMOD=ANY
      ADDON NAME=NATSHARE,
      STAT=GLOBAL
```

5. アプリケーションのフロントエンド部分には、対応するドライバのリエントラント部分が含まれている必要があります (NAMBS2 CODE=RENT、NAMTIAM CODE=RENT、または NURENT)。

例：

NAMTIAM のフロントエンド部分：

```
/EXEC $TSOSLNK                                /* Front part of NAMTIAM
PROG NATURAL,LOADPT=X'1000000',XREF=YES
TRAITS RMODE=ANY,AMODE=31
INCLUDE NRTSTART,libname                       /* Front part of NAMTIAM
INCLUDE NRTRENT,libname                       /* Reentrant part of NAMTIAM
INCLUDE NRTSTUB,libname                       /* BS2STUB
INCLUDE NRTPARM,libname                       /* Natural Parameter Module
```

UTM フロントエンド部分：

```
/EXEC $TSOSLNK                                /* UTM Front-end part
PROG NUTvrs,FILENAM=NATUTM,LOADPT=X'1000000',XREF=YES
TRAITS RMODE=ANY,AMODE=31
INCLUDE KDCNUT,libname                        /* UTM KDCROOT
INCLUDE NUTSTART,libname                     /* NATUTM
INCLUDE NUTRENT,libname                      /* NURENT
INCLUDE NUTSTUB,libname                      /* BS2STUB
INCLUDE NUTPARM,libname                      /* Natural Parameter Module
INCLUDE SWPPARM,libname                      /* Swap Pool Parameter Module
```

上記の例で、*libname*はライブラリの名前、*vrs*は製品のバージョン、リリース、およびシステムメンテナンスレベルを表します。

29 Natural ロードプールの更新

▪ 前提条件／制限	174
▪ 手順	174
▪ PREFRESH プログラムのキーワードパラメータ	175

このドキュメントでは、Naturalロードプールの更新に適用される前提条件、制限、および手順について説明し、PREFRESHプログラムで提供されるキーワードパラメータのリストを示します。

以下のトピックについて説明します。

前提条件／制限

- Natural ロードプールは、CMPSTART プログラムを使用してキーワードパラメータ ACCS=WRITE で起動されている必要があります。
- バッチアプリケーションでも使用される Natural ロードプールは、Natural バッチアプリケーションの稼働中は更新してはなりません。更新が許可されるのは、TIAM および UTM アプリケーションの場合のみです。
- 新しい Natural ニュークリアスは、グローバル共通メモリプールにのみロードできます。

手順

- 新しい Natural ニュークリアスを共通メモリプールにロードする場合は、リンクされる（リエントラント）ニュークリアスの名前と既存の名前が同じである必要があります。Natural ニュークリアスの名前は、グローバル共通メモリプールの名前と同じです。

例：

既存の Natural ニュークリアスは、CMPSTART プログラムを使用して以下のパラメータで起動されました。

```
/EXEC (CMPSTART,NATURAL.MOD)
NAME=NATSHARE,POSI=ABOVE,ADDR=250,PFIX=YES,SIZE=2MB,ALNK=NO
ACCS=WRITE,LIBR=NATURAL.USER.MOD.A
```

- 新しくリンクされる Natural ニュークリアスは、NATURAL.USER.MOD.B ライブラリからグローバル共通メモリプールにロードされます。これは PREFRESH プログラムによって実行されます。

例：

```
/.PREFRESH LOGON
/OPTION DUMP=YES
/SYSFILE SYSOUT=LST.PREFRESH.NATSHARE
/SYSFILE SYSDTA=(SYSCMD)
/EXEC (PREFRESH,NATURAL.MOD)
NAME=NATSHARE,LIBR=NATURAL.USER.MOD.B
/LOGOFF N
```

または

```
/load (prefresh,natural.mod) <enter>
% BLS0517 MODULE 'PREFRESH' LOADED
/r <enter>
*name=natshare,libr=natural.user.mod.b <enter>
* <enter>
REFR050: LOAD POOL NATSHARE IS SUCCESSFULLY REFRESHED
/
```

- 新しいNaturalニュークリアスが正常にロードされたことは、メッセージによって確認されます。

```
REFR050: LOAD POOL name IS SUCCESSFULLY REFRESHED
```

PREFRESH プログラムのキーワードパラメータ

PREFRESH プログラムには、以下のキーワードパラメータがあります。

NAME | **LIBR** | **LOAD** | **ALNK** | **TIM1** | **TIM2**

PREFRESH プログラムの構文は以下のとおりです（使用可能なデフォルト値がある場合はそれを示しています）。

```
REFRESH NAME=name,LIBR=library,LOAD=BIND,ALNK=NO,TIM1=10,TIM2=20
```

NAME - 共通メモリプールおよびモジュールの名前

このパラメータは、モジュールの名前と共通メモリプールの名前を指定します。名前の指定は必須です。デフォルト値はありません。

NAME=xxxxxxxx	xxxxxxxx は有効なモジュールおよび共通メモリプールの名前です。 既存のモジュール/共通メモリプール名と同じ名前である必要があります。 最大文字数は 8 です。
---------------	---

LIBR - ロードライブラリ

このパラメータは、定義済みモジュールのロード元を指定します。名前の指定は必須です。デフォルト値はありません。

LIBR=library	library はロードライブラリの名前です。
--------------	-------------------------

LOAD - モジュールのロードメソッド

このパラメータは、モジュールを共通メモリプールにロードする際に使用するマクロを指定します。

LOAD=ASHARE	ASHARE マクロが使用されます。
LOAD=BIND	デフォルトで BIND マクロが使用されます。

 **重要:** LOAD=ASHARE を定義する場合は、共通メモリロードプールの起動 (CMPSTART プログラム使用) に LOAD=ASHARE も定義する必要があります。

ALNK - AUTOLNK 機能のアクティブ化

このパラメータは、ダイナミックバインダローダー (DBL) の AUTOLNK 機能をアクティブにするかどうかを指定します。

ALNK=YES	AUTOLNK 機能はアクティブになります。
ALNK=NO	デフォルトで AUTOLNK 機能は非アクティブになります。

TIM1 - ロードプール更新開始までの待機時間 (秒数)

このパラメータは、新しいNaturalニュークリアスがロードされるまで待機する時間を秒数で指定します。これにより、ニュークリアスで現在アクティブなNaturalセッションと同期します。

TIM1=xx	xx は 1~99 の範囲で指定する必要があります。
TIM1=10	デフォルト値は 10 秒です。

TIM2 - 新しい Natural ニュークリアスロード後の待機時間 (秒数)

このパラメータは、新しいNaturalニュークリアスのロードが完了した後、対応するアプリケーションのシリアライゼーションIDが有効になるまで待機する時間を秒数で指定します。これにより、新しくロードしたニュークリアスのすべてのアドレス定数の相対化と同期します。

TIM2=xx	xx は 1~99 の範囲で指定する必要があります。
TIM1=20	デフォルト値は 20 秒です。

30 メッセージ処理の最適化

- 画面出力処理 180
- 画面内容の復元 180

以下のトピックについて説明します。

画面出力処理

Natural は、さまざまなメッセージ最適化機能を備えています。Natural では、出力画面を送信する前に、画面のどの部分に変更されたかを判別して、実際に変更されたデータのみを送信します。

これは、2つの連続する端末出力の間で、端末内容の一部または全体が以下の方法で変更された場合に考慮されます。

- CLEAR キーの使用
- システムレベルでのダイアログステップの介入 (K2 割り込みや類似の割り込み)

これは、特に、外部 CALL インターフェイスによって Natural から呼び出されたサブプログラムがダイアログ出力を生成する場合に当てはまります。

画面内容の復元

上記の場合は、以下のいずれかの方法を使用して、最新の端末画面内容を復元できます。

- 端末コマンド %R の発行
- ステートメント SET CONTROL 'R' の使用

31 Natural でサポートされる Siemens 端末タイプ

■ 9748 タイプ	182
■ 975n シリーズ	182
■ 9763M タイプ	183

このドキュメントでは、Natural での Siemens 端末タイプのサポート方法について説明します。

以下のタイプがサポートされています。

- 974 *n*
- 975 *n*
- 976 *n*
- テレックスデバイス

9748 タイプ

現在、さまざまな 9748 タイプのデータステーションがあります。デバイスの古さによっては、9750、9755、または 9756 モードの方がより強力にサポートできます。このタイプの古いシリーズのデバイスは、1行当たりのフィールド数に制限があるため、9750として定義する必要があります。

ネットワークでは各種の端末タイプが PDN ですべて 9750 として定義されていることがよくありますが、端末タイプも Natural セッション中に端末コマンド %T= で変更して現在使用中のデバイスタイプとの整合性を維持できます。

975n シリーズ

975*n* シリーズの各種のデバイスはそれぞれ大きく異なります（1行当たりのフィールド区切り文字の最大数、保護された空行の明るさ、フィールドプロパティに対する表示特性の標準調整など）。

これらのデバイスをサポートする 4つの端末ドライバルーチンが用意されています。これにより、9755 または 9756 タイプのモノクロデバイスに対して、さまざまな表示特性に関する最適なサポートが可能になります。異なるデバイスを PDN で 975*n* として生成できます。

デバイスの中には、オペレーティングシステムの照会で認識できないものもあります（SVC 70）。このため、Natural では、生成時にこれらの "論理端末タイプ" を各種の物理端末タイプと関連付けられるようにしています。

- TIAM 環境では、これはパラメータ T975X によって実行されます。
- UTM 環境では、KDCDEF アプリケーション生成用の PTERM ステートメントのパラメータ TERMN が、この目的のために使用されます。

9763M タイプ

9763M タイプの端末（モノクロ）は 9756 タイプの端末として扱われます。

32 9750 デバイスのファンクションキーサポート

- キー割り当て 186
- キー割り当てのモード 186

以下のトピックについて説明します。

キー割り当て

Naturalでは、ファンクションキーはデータを特定のコマンド／実行情報とともにプログラムに転送する機能を持ちます。

現在の Siemens 端末デバイスタイプでサポートされているのは F1～F5 キーのみなので、この目的にはプログラマブルPキー（P1～P20）が使用されます。つまり、これらのキーにファンクションキー値 PF1～PF20（3270 用語）が割り当てられます。

押されたキーの ID は、Natural がロードしたキー割り当てとキー送信コード F5 を組み合わせて作成されます。これにより、DUE1 を使用して送信された類似のデータタイプの区別が可能になります。Natural では、F5 を使用して、ファンクションキー解決を認識し、P キー値をコードとして解釈します。他のインスタンスでは、データは実行中のプログラムに転送されます。

キーのロードは、端末コマンドを使用して、または実行中のプログラムから SET CONTROL ステートメントを使用して制御します。

キー割り当てのモード

キー割り当てのモードには、以下の3つのタイプがあります。

KN	端末タイプ 974 <i>n</i> 、9750～9755 では、リテラル %K1～%K20 がキーに割り当てられます（端末コマンド %KN またはステートメント SET CONTROL 'KN'）。 端末タイプ 9756、9758、976 <i>n</i> では、キー送信コード F1～F20 がキー P1～P20 にロードされます。
KO	リテラル 01～20 とキー送信コード F5 がキーに割り当てられます（端末コマンド %KO またはステートメント SET CONTROL 'KO'）。
KS	リテラル A～T とキー送信コード F5 がキーに割り当てられます（端末コマンド %KS またはステートメント SET CONTROL 'KS'）。

KS モードでは、各出力メッセージの最後の2つの端末位置にダミーフィールドが生成されます。このフィールドは、キー値を受信して転送するために使用されます。データ転送の前に、キーに割り当てられた移動機能を使用してカーソルがこのフィールドに移動されます。

各端末コマンドの末尾に N を指定する（%KNN、%KON、または %KSN）と、対応するファンクションキーモードがアクティブになるのみで、値は P キーにロードされません。

すべてのモードで、カーソル位置に依存するキー処理は、現在の割り当てによって結果が異なります。例えば、ヘルプキーは、フィールド割り当てに基づいて、特定のフィールドのグローバ

ルまたはローカルいずれかのヘルプ処理を呼び出します。このような機能は、PF21～PF23の解釈済みキー（F1～F3）を使用して制御する必要があります。

33 共通メモリプール

- グローバル共通メモリプール 190
- ローカル共通メモリプール 194

このドキュメントでは、BS2000/OSD 環境の Natural でグローバル共通メモリプールを起動および停止するプログラム、および BS2000/OSD 環境の Natural でローカルまたはグローバル共通メモリプールを定義するマクロについて説明します。

以下のトピックについて説明します。

グローバル共通メモリプール

BS2000/OSD 環境の Natural でグローバル共通メモリプールを起動および停止するために、以下のプログラムが用意されています。

■ CMPSTART

■ CMPEND

 **注意:** 以下の説明で、*vrs* は製品のバージョン、リリース、およびシステムメンテナンスレベルを表します。

CMPSTART プログラム

CMPSTART プログラムの機能は以下のとおりです。

- グローバル共通メモリプールを独自の開始タスクで起動します。
- 定義済みのモジュールをグローバル共通メモリプールにロードします。
- グローバル共通メモリプールを初期化します。

キーワードパラメータ `TXTSIZE` および `BPLIST` (下記参照) は、Natural グローバルバッファプールを起動するときおよび CMPSTART プログラムにのみ有効です。

キーワードパラメータ `JV` および `JVSUFx` (下記参照) は、グローバル共通メモリプールを起動するときの CMPSTART プログラムにのみ有効です。

その他のすべてのキーワードパラメータは、BS2STUB モジュールの生成に使用される `ADDON` マクロのキーワードパラメータと同じです。

以下のキーワードパラメータを使用できます。

`TXTSIZE` | `BPLIST` | `JV` | `JVSUFx`

TXTSIZE - バッファプールのテキストレコードのサイズ

このキーワードパラメータは、Natural バッファプールのテキストレコードのサイズを KB 単位で定義します。

TXTSIZE=xx	xx に指定できる値は 1、2、4、8、12、16 です。
TXTSIZE=4	デフォルトで Natural バッファプールのテキストレコードのサイズは 4 KB になります。

BPLIST - グローバルバッファプールのプリロードリスト

このキーワードパラメータは、Natural グローバルバッファプールのプリロードリストの名前を定義します。プリロードリストの定義済み Natural プログラムは、最初のユーザーのログオン時に Natural グローバルバッファプールにロードされます。

BPLIST=name	Natural プロファイルパラメータ BPLIST を参照してください。
-------------	---------------------------------------

JV - ジョブ変数の作成

このキーワードパラメータは、ジョブ変数を作成するかどうかを定義します。このジョブ変数により、共通メモリプールのステータスをジョブ制御言語で制御できるようになります。

0	共通メモリプールの用意ができていません（作成モード）。
1	共通メモリプールの用意ができています（正常に有効化され初期化されました）。

ジョブ変数名は 2 つの部分から成ります。

- 1 つ目の部分は、共通メモリプールの名前です（キーワードパラメータ NAME のオペランド）。
- 2 つ目の部分は、キーワードパラメータ JVSUFIX のオペランドです（下記参照）。

ジョブ変数ナビゲーションのロジック：

CMPSTART プログラムが起動すると、ジョブ変数が使用可能かどうかを確認されます。使用可能な場合は、ジョブ変数の値が"0"に設定されます。使用できない場合は、ジョブ変数はカタログ化され、値が"0"に設定されます。共通メモリプールが正常に有効化され初期化されると、ジョブ変数の値が"1"に設定されます。グローバル共通メモリプールが終了すると、ジョブ変数は削除されます。

JV=YES	ジョブ変数が作成されます。
JV=NO	デフォルトでジョブ変数は使用されません。

JVSUFIX - ジョブ変数名の接尾辞

このキーワードパラメータは、ジョブ変数名の 2 番目の部分を定義します。

JVSUFIX=xxxxxxxx	ジョブ変数名の 2 番目の部分を最大 8 文字で指定します。
JVSUFIX=.SAG.JV	これはデフォルト値です。

例：

```
NAME=EDTvrsgA,TYPE=EDT,JV=YES,JVSUFIX=.SAG##JV . . . . .
```

ジョブ変数名は EDTvrsgA.SAG##JV です。

オペレータコマンド

以下のオペレータコマンドは、グローバル共通メモリプールを終了します。

```
/INTRtsn,STOP
```

または

```
/INTRtsn,END
```

以下のオペレータコマンドは、グローバル共通メモリプールの名前、場所、アドレス、サイズ、およびアクティブ時間をコンソールに表示します。

```
/INTRtsn,DPRM
```

以下のオペレータコマンドは、グローバル共通メモリプールの開始タスクをダンプ付きで終了します。

```
/INTRtsn,DUMP
```

例：

■ グローバルロードプール（共有ニュークリアス）を起動するには

```
/.NATSHRE LOGON
/OPTION DUMP=YES
/SYSFILE SYSDTA=(SYSCMD)
/SYSFILE SYSOUT=LST.NATSHARE
/EXEC (CMPSTART,NATvrs.MOD)
NAME=NATSHARE,SIZE=2MB,POSI=ABOVE,ADDR=250,SCOP=GLOBAL
PFI=YES,ALNK=NO,LIBR=NATvrs.USER.MOD
/SYSFILE SYSDTA=(PRIMARY)
/LOGOFF
/* NATSHARE IS THE NAME OF THE LINKED NATURAL REENTRANT MODULE. IT IS ALSO THE
/* NAME OF THE COMMON MEMORY POOL. THE ADDRESS OF THE GLOBAL NATURAL LOAD POOL
/* MUST BE DEFINED. THE ADDRESS MUST BE FIXED (PFI=YES).
```

■ **Natural** グローバルバッファプールを起動するには

```
/.BPvrsGA LOGON
/OPTION DUMP=YES
/SYSFILE SYSDTA=(SYSCMD)
/SYSFILE SYSOUT=LST.BPvrsGA
/EXEC (CMPSTART,NATvrs.MOD)
NAME=BPvrsGA,TYPE=NAT,POSI=ABOVE,SIZE=2048KB,SCOP=GLOBAL
/SYSFILE SYSDTA=(PRIMARY)
/LOGOFF
/* FOR A NATURAL BUFFER POOL, THE OPERAND OF PARAMETER "TYPE" MUST BE DEFINED
/* AS 'NAT'.
```

■ **ESA** データスペースを備えた **Natural** グローバルバッファプールを起動するには

```
/.BPvrsA LOGON
/OPTION DUMP=YES
/SYSFILE SYSOUT=LST.BPvrsGA
/SYSFILE SYSDTA=(SYSCMD)
/EXEC (CMPSTART,NATvrs.BS2.MOD)
NAME=BPvrsGA,TYPE=NAT,POSI=ABOVE,SIZE=10MB,ADDR=260,DESA=YES
DATA=32MB
/SYSFILE SYSDATA=(PRIMARY)
/LOGOFF N
```

CMPEND プログラム

CMPEND プログラムは、すべてのグローバル共通メモリプールの開始タスクを終了します。CMPEND で入力するのは、グローバル共通メモリプールの名前です。

例：

```
/SYSFILE SYSDTA=(SYSCMD)
/EXEC (CMPEND,NATvrs.MOD)
NATSHARE,BPvrsGA
/* THE DELIMITER FOR THE DEFINED NAMES IS ' ' OR ','.
```

ローカル共通メモリプール

このセクションでは、BS2000/OSD 環境の Natural でローカル（またはグローバル）共通メモリプールを定義できるマクロについて説明します。

- BS2STUB マクロ
- ADDON マクロ
- ADDEND マクロ
- BS2STUB マクロのアセンブルの例

BS2STUB マクロ

BS2STUB マクロの機能は以下のとおりです。

- ローカル共通メモリプールを起動します。
- 定義済みのグローバル共通メモリプールに接続します。
- 定義済みのモジュールをローカル共通メモリプールにロードします。
- 呼び出された 3GL プログラムをダイナミックにロードします。

BS2STUB マクロには以下のパラメータがあります。

```
name BS2STUB PARMOD=nn,PROGMOD=xxx
```

name - CSECT の名前

<i>name</i>	CSECT の名前を指定します。最初の 3 文字に NAT を使用することはできません。
<i>name</i> BS2STUB	これはデフォルトの名前です。

PARMOD - 24/31 ビットアドレッシングモード

このパラメータは、24 または 31 ビットアドレッシングモードを使用するかどうかを指定します。

PARMOD= <i>nn</i>	<i>nn</i> に指定できる値は 24 または 31 (ビット) です。
PARMOD=31	デフォルトでアドレッシングモードは 31 ビットに設定されます。

PROGMOD - 16 MB 境界より上または下にロード

このパラメータは、ダイナミックロードされるプログラムを 16 MB 境界より上または下のどちらにロードするかを指定します。

PROGMOD=ANY	ANY は、モジュールが 16 MB 境界より上または下にロードされることを示します。これはデフォルト設定です。
PROGMOD=24	24 は、モジュールが 16 MB 境界より下にロードされることを示します。

ADDON マクロ

ADDON マクロは、BS2STUB プログラムの ADDON テーブルで共通メモリプールを定義します。このマクロには、以下のキーワードパラメータがあります。これは、CMPSTART プログラムにも適用されます。

ACCS | ADDR | ALNK | DATA | DESA | LIBR | LOAD | NAME | PFIK | POSI | SCOP | SIZE | STAT | TYPE | WAIT

ACCS - 共通メモリプールへのアクセス

このパラメータは、共通メモリプールへのアクセス方法を指定します。

ACCS=READ	読み取り専用アクセスであることを示します（書き込み禁止）。 ACCS=READ を設定するには、ユーザー ID がユーザーカタログの BS2000/OSD CSTMP マクロに対して認証されている必要があります（JOIN コマンドで C-M=YES を使用）。
ACCS=WRITE	デフォルトで共通メモリプールは書き込み可能になります。

ADDR - 共通メモリプールアドレスのサイズ

このパラメータは、共通メモリプールの定義済みアドレスのメガバイト数を指定します。サイズの指定は必須です。デフォルト値はありません。

ADDR= <i>number</i>	<i>number</i> は 0 以上である必要があります。
---------------------	---------------------------------

ALNK - AUTOLNK 機能のアクティブ化

このパラメータは、ダイナミックバイндаローダー（DBL）の AUTOLNK 機能をアクティブにするかどうかを指定します。

ALNK=NO	AUTOLNK 機能は非アクティブになります。
ALNK=YES	デフォルトで AUTOLNK 機能はアクティブになります。

DATA - データスペースエリアのサイズ

このパラメータは、DESA パラメータと組み合わせて指定でき、起動するバッファプールまたはスワッププールのデータスペースエリアのサイズを定義します。以下の設定が可能です。

DATA= <i>nnn</i> MB	データスペースエリアのサイズを MB 単位で指定します。
DATA= <i>nnn</i> KB	データスペースエリアのサイズを KB 単位で指定します。

ADDON マクロでの DATA パラメータの使用

- ローカルバッファプールを起動するには、DESA=YES を指定し、DATA パラメータを使用してデータスペースエリアのサイズを MB または KB 単位で指定します。デフォルト値がないため、サイズの指定は必須です。
- グローバルバッファプールまたはグローバルスワッププールを接続するには、DESA=YES を指定し、DATA パラメータは CMPSTART プログラムに指定済みなので省略します。

CMPSTART プログラムでの DATA パラメータの使用

グローバルバッファプールを起動するには、DESA=YES を指定し、DATA パラメータを使用してデータスペースエリアのサイズをMBまたはKB単位で指定します。デフォルト値がないため、サイズの指定は必須です。

DESA - ESA データスペースエリア

このパラメータは、NaturalバッファプールまたはNaturalスワッププールにESAデータスペースエリアを作成するかどうかを決めるために指定する必要があります。

DESA=YES	ESA データスペースエリアが作成されます。
DESA=NO	デフォルトで ESA データスペースエリアは作成されません。

- ESA データスペースは、TYPE=NAT または TYPE=SWP のバッファプールにのみサポートされません。
- パラメータ DESA=YES が関連するのは、ESA データスペースのあるグローバル共通メモリプール（独自の開始タスクを持つ CMPSTART）または ESA データスペースのあるローカル共通メモリプール（BS2STUB/ADDON）を作成する場合のみです。
- 既存のグローバル共通メモリプールに接続する場合（BS2STUB/ADDON）は、パラメータ DESA は無意味です。

 **注意:** ESA データスペースは、独自の開始タスクを持つ1つのグローバル共通メモリプールにのみ作成してください。ESA データスペースを作成したタスクが正常終了または異常終了すると、ESA データスペースは使用できなくなります。

LIBR - ロードライブラリ

このパラメータは、定義済みモジュールのロード元を指定します。デフォルト値はありません。パラメータ LIBR のオペランドを指定しない場合は、共通メモリプールのみが有効になります（ENAMP+REQMP）。

LIBR= <i>library</i>	<i>library</i> はロードライブラリの名前です。
LIBR=BLSLIB	リンク名が BLSLIB および BLSLIB01~BLSLIB99 のライブラリが使用されます。
LIBR=CLASS-4	モジュールはサブシステムとしてクラス 4 メモリにロードされます。

共通メモリプール

LOAD - モジュールを共通メモリプールにロードするメソッド

このパラメータは、モジュールを共通メモリプールにロードする際に使用するマクロを指定します。

LOAD=ASHARE	ASHARE マクロが使用されます。 ASHARE を定義している場合は、パラメータ PFIX のオペランドを YES にする必要があります。
LOAD=BIND	デフォルトで BIND マクロが使用されます。

NAME - 共通メモリプール／モジュールの名前

このパラメータは、モジュールの名前と共通メモリプールの名前のいずれかまたは両方を指定します。名前の指定は必須です。デフォルト値はありません。

NAME=name	name は、共通メモリプールまたはモジュールの有効な名前です。
-----------	----------------------------------

名前の最大文字数は以下のとおりです。

8 文字	モジュール名（共通メモリプールの名前）。Natural バッファプール。
16 文字	他のすべての共通メモリプール。

PFIX - 固定アドレス

このパラメータは、共通メモリプールのアドレスを固定するかどうかを指定します。

PFIX=YES	共通メモリプールのアドレスを固定する必要があります。
PFIX=NO	デフォルトで共通メモリプールのアドレスを固定する必要がありません。

グローバル Natural ロードプールの場合は、このパラメータを YES に設定する必要があります。

POSI - 16 MB 境界からの相対位置

このパラメータは、共通メモリプールの位置を指定します。位置は 16 MB 境界より上または下を指定できます。

POSI=ABOVE	共通メモリプールは 16 MB 境界より上に配置されます。
POSI=BELOW	デフォルトで共通メモリプールは 16 MB 境界より下に配置されます。

SCOP - 共通メモリプールの範囲

このパラメータは、共通メモリプールの範囲を指定します。

SCOP=LOCAL SCOP=GROUP SCOP=GLOBAL	共通メモリプールの範囲については、BS2000/OSD ドキュメントの ENAMP マクロの説明を参照してください。
SCOP=GLOBAL	これはデフォルト設定です。

SIZE - 共通メモリプールのサイズ

このパラメータは、共通メモリプールのサイズを MB または KB 単位で指定します。

SIZE=nKB SIZE=nMB	共通メモリプールのサイズを n KB または n MB に指定します。
SIZE=1MB	デフォルトで共通メモリプールのサイズは 1 MB になります。

STAT - 共通メモリプールのステータス

このパラメータは、共通メモリプールのステータスを指定します。

STAT=GLOBAL	共通メモリプールのステータスは GLOBAL です (CMPSTART によって起動)。
STAT=LOCAL	共通メモリプールのステータスは LOCAL です (BS2STUB によって起動)。 デフォルトで共通メモリプールのステータスは LOCAL になります。



注意: CMPSTART プログラムが実行されている場合は、STAT パラメータは無視されます。

TYPE - 共通メモリプールのタイプ

このパラメータは、共通メモリプールのタイプを指定します。タイプの指定は必須です。デフォルト値はありません。

TYPE=COM	Natural DCOM プール
TYPE=EDT	エディタバッファプール
TYPE=MON	Natural Monitor プール (SYSMON)
TYPE=NAT	Natural バッファプール
TYPE=SRT	ソートバッファプール
TYPE=SWP	Natural スワッププール
TYPE=USR	ユーザーバッファプール

WAIT - アプリケーション起動時の共通メモリプールの有効化または待機

このパラメータは、共通メモリプールを、アプリケーションの起動時にすぐに有効にするか、Natural から要求があるまで待機してから有効にするかを指定します。

WAIT=YES	共通メモリプールは Natural からの要求を待機し、要求があった時点で有効になります。
WAIT=NO	デフォルトで共通メモリプールはアプリケーションの起動と同時に有効になります。



注意: CMPSTART プログラムが実行されている場合は、WAIT パラメータは無視されます。

ADDEND マクロ

ADDEND マクロは、ADDON マクロの定義の終了を定義します。ADDEND にはパラメータはありません。

BS2STUB マクロのアセンブルの例

```
BS2STUBA BS2STUB PARMOD=31,PROGMOD=24          31-BIT ADDRESSING MODE,  
*                                                LOAD 3GL PROGRAMS BELOW  
* +-----+  
* I Define the Natural global load pool with Name NATSHARE  
* +-----+  
*         ADDON NAME=NATSHARE,STAT=GLOBAL  
* +-----+  
* I Define the Natural local swap pool  
* +-----+  
*         ADDON NAME=SWAPvrsLA,TYPE=SWP,SIZE=16MB,STAT=LOCAL,POSI=ABOVE  
* +-----+  
* I Connecting a Natural global buffer pool with ESA data space  
* +-----+
```

```
ADDON NAME=BPvrsGA,TYPE=NAT,STAT=GLOBAL
* +-----+
* | Creating/Connecting a Natural local buffer pool with ESA data space
* +-----+
      ADDON NAME=BPvrsLA,TYPE=NAT,POSI=ABOVE,SIZE=10MB,
          STAT=LOCAL,SCOP=LOCAL,DESA=YES,DATA=32MB
      ADDEND
      END
```


34 Naturalアプリケーションでのダイナミック再ロード可能 3GL プログラムの呼び出し

▪ ストレージ割り当てルール	204
▪ スレッド作成ルール	204
▪ アドレスモードの依存関係	204

Natural アプリケーションでのダイナミック再ロード可能 3GL プログラムの呼び出し

このドキュメントでは、Natural アプリケーション内でダイナミック再ロード可能 3GL プログラムを呼び出すときに適用されるアドレスモードの選択ルールについて説明します。

以下のトピックについて説明します。

ストレージ割り当てルール

ダイナミック再ロード可能 3GL プログラムが 16 MB 境界より上または下のどちらにロードされるかは、BS2STUB マクロのキーワードパラメータ PROGMOD に基づきます。

パラメータ	説明
PROGMOD=ANY	プログラムは 16 MB 境界より上または下にロードされます。 どちらにロードされるかは、アプリケーションのアドレスモードと、ロードされる 3GL プログラム内の AMODE または RMODE ステートメントの存在によって決まります。
PROGMOD=24	3GL プログラムは常に 16 MB 境界より下にロードされます。

スレッド作成ルール

Natural ユーザースレッドが 16 MB 境界より上または下のどちらに作成されるかは、NATUTM マクロのキーワードパラメータ NUAADDR と、NAMTIAM および NAMBS2 マクロの REQMLC によって決まります。

アドレスモードの依存関係

以下に、ダイナミック再ロード可能 3GL プログラムを呼び出すための生成コンフィグレーションで使用されるアドレスモードの概要を示します。

1. **ILCS** または **CRTE** インターフェイスを使用した **3GL** プログラム呼び出し
2. **ILCS** または **CRTE** インターフェイスを使用しない **3GL** プログラム呼び出し
3. **PEND PR** を使用した **Natural/UTM** ドライバからの **31** ビット非対応 **UTM** 部分プログラム呼び出し

1. ILCS または CRTE インターフェイスを使用した 3GL プログラム呼び出し

'ILCS=YES' または 'ILCS=CRTE'

事例	アプリケーション生成に使用	3GL プログラム呼び出しに使用
1	PARMOD=31 または PARMOD=(31, ABOVE)	AMODE=31
2	PARMOD=24	AMODE=24

2. ILCS または CRTE インターフェイスを使用しない 3GL プログラム呼び出し

'ILCS=NO'

事例	アプリケーション生成に使用	3GL プログラム呼び出しに使用
1	PARMOD=31 Natural ユーザースレッドは 16 MB 境界より上に配置され、3GL プログラムは 16 MB 境界より上または下にロードされます。	AMODE=31
2	PARMOD=31 Natural ユーザースレッドと 3GL プログラムは 16 MB 境界より下にロードされます。	AMODE=24
3	PARMOD=(31, ABOVE) Natural ユーザースレッドは 16 MB 境界より上に配置され、3GL プログラムは 16 MB 境界より上または下にロードされます。	AMODE=31
4	PARMOD=(31, ABOVE) Natural ユーザースレッドは 16 MB 境界より下に配置され、3GL プログラムは 16 MB 境界より下にロードされます。	AMODE=24
1.	モジュール BS2GLUE が、ロードされる 3GL プログラムと同じライブラリにある必要があります。または、モジュール BS2GLUE のロードモジュールライブラリが、STARTJOB で BLSLIB として定義されている必要があります。	
2.	このようなコンフィグレーションが Natural/UTM アプリケーションで存在する場合は、キーワード パラメータ KB を KB=NO として定義する必要があります。	
5	Natural/UTM アプリケーションが PARMOD=31 を使用して作成されました。 Natural ユーザースレッドは 16 MB 境界より上または下に配置され、NURENT マクロのキーワードパラメータ CALLM31 は CALLM31=YES として定義されています。	AMODE=31

Natural アプリケーションでのダイナミック再ロード可能 3GL プログラムの呼び出し

3. PEND PR を使用した Natural/UTM ドライバからの 31 ビット非対応 UTM 部分プログラム呼び出し

アプリケーションは PARMOD=31 を使用して生成され、NATUM マクロのキーワードパラメータ SWAMODE は SWAMODE=YES として定義されています。

Natural は、UTM KDCS インターフェイスの各呼び出しの前に元の 24 ビットアドレスモードに切り替わり、制御が UTM ドライバに返されると、31 ビットアドレスモードに切り替わります。

35 出力ファイル／ワークファイルサーバー NATPWSV2

■ セットアップ	208
■ オペレーション	209

このドキュメントでは、BS2000/OSDでのRPCバッチサーバー環境の出力ファイル／ワークファイルサーバー NATPWSV2について説明します。このサーバーは、**RPCサーバーフロントエンド**によって起動されます。

以下のトピックについて説明します。

『メッセージおよびコード』ドキュメントの「出力ファイル／ワークファイルサーバー NATPWSV2のエラーメッセージ」も参照してください。

セットアップ

出力ファイル／ワークファイルサーバー NATPWSV2 は、前方イベントメソッドを使用して RPC バッチサーバー NATFSTB2 と通信します。

出力ファイル／ワークファイルサーバーをセットアップするには、次の手順に従います。

- モジュール NATFSTB2 を **Natural** ニュークリアスにリンクします。NATWKFB2 プログラムが NATFSTB2 モジュールに置き換えられます。
- 新しいプールタイプ PWK（出力ファイル／ワークファイル制御プール）の共通メモリプールのために、モジュール NATPWSV2 を ADDON パラメータ定義とともに BS2STUB プログラムでリンクする必要があります。この共通メモリプールは、定義済みの固定アドレスを使用してセットアップする必要があります。以下に例を示します。

```
PWKSTUB  BS2STUB  PARMOD=31,PROGMOD=ANY,UNRES=*DBLOPT
          ADDON  NAME=PWK#POOL,          NAME OF CONTROL POOL          -
                TYPE=PWK,                TYPE OF CONTROL POOL          -
                SIZE=1MB,                 POOL SIZE IN MB               -
                STAT=LOCAL,               POOL STATUS IS LOCAL          -
                SCOP=GLOBAL,              SCOPE IS GLOBAL                -
                POSI=ABOVE,                POOL POSITION IS ABOVE         -
                ADDR=19,                   ADDRESS IS X'1300000'         -
                PFIX=YES,                  POOL ADDRESS IS FIXED         -
                ACCS=WRITE                  NO POOL PROTECTION            -
          ADDEND
```

同じ ADDON パラメータ定義（ADDR= を除く）が、RPC バッチサーバーのフロントエンド部分にリンクされる BS2STUB プログラムに含まれている必要があります。

出力ファイル／ワークファイルサーバーのリンクの例：

```
/EXEC $TSOSLNK  
PROG PWKSRV, FILENAM=E.NATPWSV2, LOADPT=X'1000000', XREF=YES  
TRAITS RMODE=ANY, AMODE=31  
INCLUDE NATPWSV2, NATURAL.NATvrs.MOD  
INCLUDE PWKSTUB, USERLIB  
BIND
```

上記の意味は以下のとおりです。

vrs Natural for Mainframes の現在のバージョンです。

USERLIB ユーザー固有のライブラリです。

オペレーション

出力ファイル／ワークファイルサーバーと RPC バッチサーバー間のデータ交換は、出力ファイル／ワークファイル制御プール (TYPE=PWK) で行われます。

出力ファイル／ワークファイルサーバーの起動

出力ファイル／ワークファイルサーバーは、RPC バッチサーバーの前に起動しておく必要があります。

RPC バッチサーバーでは、初期化された出力ファイル／ワークファイル制御プールが存在していると想定します。この初期化は、出力ファイル／ワークファイルサーバーの起動時に行われます。

Natural RPC バッチサーバーと出力ファイル／ワークファイルサーバーは相互に 1 対 1 でのみ通信します (TSN1 ⇔ TSN2)。

使用されるすべての出力ファイル／ワークファイル (リンク名 P01～P32 および W01～W32) は、出力ファイル／ワークファイルサーバーのジョブ制御に FILE コマンドを使用して定義されている必要があります。

開始ジョブの例：

```
/.PWKSRV LOGON
/ER LST.PWKSERVER.
/STEP
/OPTION DUMP=YES
/FILE WORK.W01,LINK=W01
/FILE WORK.W02,LINK=W02
.
.
/FILE PRINT.P01,LINK=P01
/FILE PRINT.P02,LINK=P02
.
.
/EXEC (NATPWSV2,NATURAL.NATvrs.MOD)
/LOGOFF N
```

出力ファイル／ワークファイルサーバーの終了

出力ファイル／ワークファイルサーバーは、CMPEND プログラムを使用して P1 イベントという方法で終了できます。出力ファイル／ワークファイルサーバーを終了するイベント名は、出力ファイル／ワークファイル制御プールの名前です。

出力ファイル／ワークファイルサーバーの終了手順の例：

```
/BEGIN-PROCEDURE LOGGING COMMANDS
/ASSIGN-SYSDTA TO=*SYSCMD
/SET-JOB-STEP
/START-PROGRAM FROM-FILE=*MODULE(LIBRARY=NATURAL.NATvrs.MOD,-
/      ELEMENT=CMPEM)
PWK#POOL      <== name of the print file/work file control pool
/SET-JOB-STEP
/ASSIGN-SYSDTA TO=*PRIMARY
/EXIT-PROCEDURE
```

すべてのエラーメッセージ（出力ファイル／ワークファイルサーバーの異常終了）は、SYSLST99 のファイル LST.PWKSERVER.tsnn に書き込まれます。

36 RPC サーバーフロントエンド

- セットアップ 212

このドキュメントでは、BS2000/OSD で出力ファイル/ワークファイルサーバ [NATPWSV2](#) を使用するRPCバッチサーバ環境にRPCサーバフロントエンドをセットアップする方法について説明します。

以下のトピックについて説明します。

セットアップ

NaturalRPCバッチサーバの生成では、Naturalバッチドライバ (NAMBS2 マクロ) のフロントエンド部分を新しいキーワードパラメータ `SERVER=YES` でアセンブルする必要があります。

例：

```
SERVFRNT NAMBS2 CODE=FRONT, -
                APPLNAM=NATSERV, -
                NUCNAME=RPCSERV, -
                DYNPAR=SYSDTA, For server parameters -
                SERVER=YES, Generate RPC server -
                ROLLTSZ=384, Roll thread size in KB -
                . All other parameter definitions as for
                . the generation of the front-end part
                . of the Natural batch driver
END
```

NaturalRPCバッチサーバのリエントラント部分の生成では、Naturalバッチドライバの生成と同じキーワードパラメータ定義を使用できます。

BS2STUB モジュール (RPCバッチサーバのフロントエンド部分) の生成では、必要な共通メモリプールを定義する必要があります。

出力ファイル/ワークファイルサーバ [NATPWSV2](#) を使用する場合は、出力ファイル/ワークファイル制御プールを定義し、Naturalのリエントラント部分のNATWKFB2モジュールをNATFSTB2モジュールに置き換えます。

```
RPCSTUB2 BS2STUB PARMOD031,PROGMOD=ANY,UNRES=*DBLOPT
        ADDON NAME=RPCSERV, Name of reentrant part (load pool)
-
        .
        .
        .
        ADDON definition for Natural Buffer Pool
        .
        ADDON definition for Natural Editor Pool
        .
        ADDON definition for Natural Swap Pool
        .
```

ADDON	NAME = PWK#POOL,	Name of print file/work file control pool
-	TYPE=PWK,	Pool type
-	SIZE=1MB,	Pool size in MB
-	STAT=LOCAL,	Pool status is local
-	SCOP=GLOBAL,	Scope is global
-	POSI=ABOVE,	Pool position is above
-	PFIX=YES,	Pool address is fixed
-	ACCS=WRITE	No pool protection
ADDEND		

Natural RPC バッチサーバでは、異なるクライアントコンテキストをユーザースレッドに保存します。これらのユーザースレッドは、スワッププールまたは Natural ロールファイルで管理されます。このため、Natural ロールファイルと Natural スワッププールは必須です。

出力ファイルとワークファイルの処理では、新しいタイプ「PWK」を使用して出力ファイル／ワークファイルサーバを生成する必要があります（「[出力ファイル／ワークファイルサーバ NATPWSV2](#)」を参照）。

RPC バッチサーバと出力ファイル／ワークファイルサーバの間のデータ交換は、新しいタイプ「PWK」を使用して共通メモリプール（出力ファイル／ワークファイル制御プール）で行われます。

RPC バッチサーバ（NATFSTB2 モジュール）と出力ファイル／ワークファイルサーバ（NATPWSV2 モジュール）の間の通信は、P1 前方イベントという方法で行われます。出力ファイル／ワークファイルサーバを使用する場合は、リエントラント部分のリンクジョブで NATWKFB2 モジュールを NATFSTB2 に置き換える必要があります。

Natural RPC バッチサーバのフロントエンド部分のリンク例：

```
/EXEC $TSOSLNK
PROG SERVER, FILENAM=BATCH.SERVER, LOADPT=X'1000000', XREF=YES
TRAITS RMODE=ANY, AMODE=31
INCLUDE NATSFED2, NATvrs.MOD           Must be bound as first module
INCLUDE RPCSFE, NATvrs.MOD             RPC front-end stub
INCLUDE SERVFRNT, USERLIB              Front-end part of Natural batch driver
INCLUDE SERVRENT, USERLIB              Reentrant part of Natural batch driver
INCLUDE RPCSTUB2, USERLIB              BS2STUB (see above)
INCLUDE SWPRMSRV, USERLIB              Swap pool parameter module
INCLUDE NATPRMSV, USERLIB              Natural parameter module
INCLUDE ADAUSER, ADAvrs.MOD
INCLUDE SSFB2C, ADAvrs.MOD
BIND
```

上記の意味は以下のとおりです。

vrs Natural for Mainframes または Adabas for Mainframes の現在のバージョンです。
USERLIB ユーザー固有のライブラリです。

スワッププールパラメータモジュールの生成方法については、「[Natural スワッププールの定義](#)」の「[マクロ NTSWPRM のキーワードパラメータ](#)」を参照してください。

Natural RPC バッチサーバのリエントラント部分のリンク例：

```
/EXEC $TSOSLNK
MOD RPCSERV,XREF=Y,MAP=Y,XDSEC=Y, SORT=YES
TRAITS RMODE=ANY,AMODE=ANY
LINK-SYMBOLS *NOESD
INCLUDE NATINV,NATvrs.MOD      Must be bound as first module
INCLUDE NATURAL,NATvrs.MOD    Natural nucleus
COMMENT NATWKFB2,NATvrs.MOD   Is replaced by
INCLUDE NATFSTB2,NATvrs.MOD   print file/work file server stub
.           All other
.           moduls
INCLUDE NATLAST,,NATvrs.MOD
BIND
```

上記の例で、*vrs* は Natural for Mainframes の現在のバージョンです。

Natural バッチサーバのパラメータの例：

```
AUTO=ON,
STACK=(LOGON DFSERVER),
RPC=(
SERVER=ON,
SRVNODE='10.20.91.202:3860:TCP',SRVNAME=DFSRV1,
RPCSIZE=128,MAXBUFF=30,
TRACE=2
),
RCA=BROKER,RCALIAS=(BROKER,BKIMBTIA),
MADIO=0,MAXCL=0,MT=0,MENU=OFF,
PRINT=((10),AM=STD),WORK=((1-10),AM=STD)
```

Natural RPC サーバークライアントのパラメータの例：

```
STACK=(LOGON DFCLIENT),
RPC=(
DFS=(DFSRV1,BKR043,,,NOSERVDIR),
RPCSIZE=128,MAXBUFF=30,
AUTORPC=ON,TRYALT=OFF
),
RCA=BROKER,RCALIAS=(BROKER,BKIMBTIA),
MADIO=0,MAXCL=0,MT=0,ETID=' '
```

NaturalRPC バッチサーバーには、サーバーメッセージの出力用に P10 という名前のファイルが必要です。出力ファイル/ワークファイルサーバーを使用する場合は、NaturalRPC バッチサーバーのジョブ制御で定義されていない限り、このファイルが出力ファイル/ワークファイルサーバーのジョブ制御で FILE 命令を使用して定義されている必要があります。

開始ジョブの例：

```
/.SERVER LOGON
/SYSFILE SYSOUT=SERVER.OUT
/SYSFILE SYSLST=SERVER.LIST
/FILE NATvrs.EDIT.WORKFILE, LINK=CMEDIT
/FILE NATvrs.SERVER.ROLLFILE, LINK=PAMNAT, SHARUPD=YES
/FILE SERVER.MSG, LINK=P10           Is required for the server messages
/FILE ADAvrs.MOD, LINK=DDLIB
/FILE ADAPARM, LINK=DDLNKPAR
/FILE EXXvrs.LIB, LINK=BLSLIB01      Broker
/FILE EXXvrs.LIB, LINK=ETBLIB       Load library
/SYSFILE SYSDTA=SERVERPARMS
/EXEC BATCH.SERVER
/LOGOFF N
```

上記の例で、*vrs* は、Natural for Mainframes、Adabas for Mainframes、または EntireX Communicator の現在のバージョンです。

EntireX Broker の生成および起動方法については、EntireX Communicator のドキュメントを参照してください。

37 バッチモードでの Natural

このドキュメントでは、Natural をバッチモードで実行する場合の考慮事項について説明します。

- **z/OS 環境下のバッチモードの Natural** z/OS オペレーティングシステムで Natural をバッチモードで実行する際の特別な考慮事項について説明します。
- **z/VSE 環境下のバッチモードの Natural** z/VSE オペレーティングシステムで Natural をバッチモードで実行する際の特別な考慮事項について説明します。
- **CMS 環境下のバッチモードの Natural** CMS で Natural をバッチモードで実行する際の特別な考慮事項について説明します。
- **BS2000/OSD 環境下のバッチモードの Natural** BS2000/OSD オペレーティングシステムで Natural をバッチモードで実行する際の特別な考慮事項について説明します。
- **バッチモードの Natural (すべての環境)** Adabas データセット、ソートデータセット、バッチ環境のサブタスクセッションサポートなど、Natural をバッチモードで実行する際の全般的な考慮事項について説明します。

Natural をバッチモードで使用する場合に適用される Natural プロファイルパラメータの概要については、「[プロファイルパラメータの概要](#)」の「[バッチモード](#)」も参照してください。

38 z/OS 環境下のバッチモードの Natural

- Natural z/OS バッチモードインターフェイスに関する全般的な情報 220
- Natural z/OS 生成パラメータ 220
- z/OS バッチモードの Natural で使用されるデータセット 223

このドキュメントでは、OS オペレーティングシステムで Natural をバッチモードで実行する際の特別な考慮事項について説明します。

以下のトピックについて説明します。

バッチモードの Natural 全般に適用される考慮事項については、以下のセクションも参照してください。

- [Adabas データセット](#)
- [ソートデータセット](#)
- [バッチモード環境のサブタスクセッションサポート](#)

Natural z/OS バッチモードインターフェイスに関する全般的な情報

Natural z/OS バッチモードインターフェイス NATOS 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複数のサービスルーチンで構成されます。

NATOS はソースモジュールとして提供され、必要に応じてカスタマイズできます。「[z/OS 環境での Natural のインストール](#)」も参照してください。NATOS は、アセンブルして Natural ニュークリアスにリンクすることも、単独で実行して共有ニュークリアスに接続することもできます。

NATOS 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つのバッチリージョン内で複数の Natural セッションを同時に開始できます。「[バッチモード環境のサブタスクセッションサポート](#)」を参照してください。

Natural z/OS 生成パラメータ

NTOS マクロには、z/OS 対応 Natural のバッチモード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、[ABEXIT](#)、[LBPNAME](#)、[LE370](#)、[SUBPOOL](#)、[TI0BSZ1](#)、[TI0BSZ2](#)、および [USERID](#) です。

ABEXIT - アベンド処理

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。

ABEXIT=ESTAE	Natural はすべてのアベンドをインターセプトし、該当するエラーメッセージを表示します。これはデフォルト値です。
ABEXIT=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。
ABEXIT=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、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。

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。共有バッファプール環境の名前を定義し、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。

LBPNAME= <i>name</i>	<i>name</i> の長さは 1~8 文字です。
LBPNAME=	デフォルト値は空白で、この場合はローカルバッファプールは共有されません。

例えば、z/OS 環境でバッチモードで、または TSO 環境で、複数の Natural セッションを同時に実行すると、Natural RPC サーバーの実行時に、各セッションで各ローカルバッファプールにストレージが割り当てられます。Natural z/OS バッチモードサーバーを除き、ローカルバッファプールはデフォルトごとに共有されるわけではありません。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は、それらをセッションごとに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。

LE370=YES	IBM の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。
LE370=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。 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、終了されることを意味します。
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をサポートします。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 による LE エラーハンドラの設定は行われません。これは、LE サブプログラムの実行中に未処理のエラーが発生すると、LE エンクレーブが終了し、Natural セッションが失われることを意味します。

以下に示すように、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* の実行 - その他のトピック」の「[LE サブプログラム](#)」を参照してください。

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。

SUBPOOL=nnn	nnn に指定できる値は "0" ~ "127" です。
SUBPOOL=0	デフォルト値は "0" です。

TI0BSZ1 - バッチ処理のプライマリ I/O バッファのサイズ

このパラメータは、バッチ処理のプライマリ 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。サーバー処理の場合は、パラメータ TI0BSZ2 が代わりに使用されます。

TI0BSZ1=nnnn	nnnn は、プライマリ 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。
--------------	--

『TP モニタインターフェイス』ドキュメントの「*Natural* TSO インターフェイスの生成パラメータ」で、生成パラメータ TI0BSZ に関する記載も参照してください。

TIOBSZ2 - サーバー処理のプライマリ I/O バッファのサイズ

このパラメータは、サーバー処理のプライマリ 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。バッチ処理の場合は、パラメータ TIOBSZ1 が代わりに使用されます。

TIOBSZ2=nnnn	nnnn は、プライマリ 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 65520 です。
--------------	---

『TP モニタインターフェイス』ドキュメントの「Natural TSO インターフェイスの生成パラメータ」で、生成パラメータ TIOBSZ に関する記載も参照してください。

USERID - システム変数 *INIT-USER の内容

このパラメータは、システム変数 *INIT-USER の内容を指定します。

USERID=YES	変数には、セキュリティアクセスコントロールブロック (ACEE) からのユーザー ID (RACF や ACF2 などのセキュリティパッケージが含まれている場合)、またはジョブカードからのユーザー ID が設定されます。
USERID=NO	ユーザー ID はジョブ名です。これはデフォルト値です。

*INIT-USER の内容は、セッションの初期化時にユーザー ID 出口 NATUEX1 で変更できます。詳細については、「Natural の構成」の「Natural ユーザー出口」の「[NATUEX1 - 認証制御のユーザー出口](#)」を参照してください。

z/OS バッチモードの Natural で使用されるデータセット

Natural z/OS バッチモードセッションで特定の機能を使用するには、以下のデータセットが必要です。

データセット	説明
CMEDIT	Software AG Editor ワークファイル
CMHCOPY	ハードコピー出力
CMOBJIN	Natural INPUT ステートメントの入力
CMPLOG	ダイナミックプロファイルパラメータレポート出力
CMPRINT	プライマリレポート出力
CMPRMIN	ダイナミックプロファイルパラメータ入力
CMPRTnn	追加レポート 01~31
CMSYNIN	プライマリコマンド入力

データセット	説明
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

これらのデータセットについては、以下で説明します。

シーケンシャルデータ出力セットの場合は、デフォルトの DCB RECFM/LRECL 情報は以下のとおりです。

RECFM=FBA および 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 で定義されていない場合は、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 のインストール方法については、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 - ハードコピーのレポート出力（オプション）

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、以下のいずれかの方法で変更できます。

- 出力ファイル 0 のプロファイルパラメータ PRINT のサブパラメータ DEST
- プロファイルパラメータ HCDEST (PRINT=((0),DEST=...) と等価)
- セッション時にシステム変数 *HARDCOPY を設定
- セッション時に端末コマンド %H を使用

出力ファイル 0 のプロファイルパラメータ PRINT のサブパラメータを使用して、ハードコピーデータセットのデフォルト値を変更できます。デフォルトのデータセット名 CMHCOPY は暗黙的にハードコピー出力データセットに対する CLOSE=FIN を示します。つまり、出力のためにハードコピー出力データセットを開いた後に名前を変更しても適用されません。開いたときに別の名前が定義されていた場合は、出力ファイル 0 のプロファイルパラメータ PRINT のサブパラメータ CLOSE に従ってハードコピーデータセットは閉じられます。

セッション中に、ハードコピーデータセットを解放してダイナミックアロケーションによって再割り当てすることができます（開く前、または閉じた後）。この場合は、アプリケーションプログラミングインターフェイス USR2021N を使用します。「SYSEXT-Natural アプリケーションプログラミングインターフェイス」を参照してください。

CMOBJIN - Natural INPUT ステートメントの入力

このデータセットを使用すると、プライマリ入力データセット CMSYNIN からではなく、Natural INPUT ステートメントによってデータを読み取ることができます。

CMOBJIN の使用はプロファイルパラメータ OBJIN によって制御されます。Natural の入力レコードデータの長さは、プロファイルパラメータ SL によって決まります。サポートされる最大レコード長 (LRECL) は 255 です。レコードフォーマット (RECFM) は固定でも可変でも可能です。

CMPLUG - 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 CMPLUG 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 CMPLUG が使用できない場合は、評価されたダイナミックプロファイルパラメータは CMPRINT に書き込まれます。

CMPRINT - プライマリレポート出力

CMPRINT は、Natural プログラム内の DISPLAY、PRINT、および WRITE ステートメントの結果をレポートするプライマリ出力に使用されます。

CMPRINT は、JCL で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。

```
//CMPRINT DD SYSOUT=*
```

この処理は、最初のレコードの書き込み時に行われます。

CMPRMIN - ダイナミックパラメータデータセット

CMPRMIN をダイナミックパラメータデータセットとして使用すると、EXEC ステートメントのジョブ制御キーワード PARM の文字列の長さ制限を超えることができます。

このファイルが使用可能な場合は、セッションの初期化時に読み取られ、ダイナミックプロファイルパラメータが取得されます。

CMPRMIN からのすべての入力レコードが、1つのパラメータ文字列に連結されます。各 CMPRMIN レコードの最初の 72 文字目までが有効です。各レコードの末尾の空白は切り捨てられます。空白以外の最後の文字がコンマの場合は、末尾の空白はすべて切り捨てられます。それ以外の場合は、1つの空白がデリミタ文字として残されます。コンマは挿入されません。

PARM ジョブ制御キーワードを使用して、ダイナミックパラメータを追加できます。追加したダイナミックパラメータは、CMPRMIN の入力から作成されたパラメータ文字列の末尾に連結されます。これらを使用して、CMPRMIN からのパラメータを上書きできます。

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 ファイル出力ステートメントで使用できます。DCB 情報 (RECFM、LRECL、BLKSIZE など) が使用できない場合は、PRINT 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 NTPRINT マクロによってデフォルトが定義されます。出力ファイルの名前はサブパラメータ DEST で上書きできます。

CMSYNIN - プライマリコマンド入力

このデータセットは、コマンド入力および Natural INPUT ステートメントの要求データの読み取りに使用されます。データの読み取りはプロファイルパラメータ OBJIN で制御されます (**CMOBJIN** も参照)。

Natural の入力レコードデータの長さは、プロファイルパラメータ SL によって決まります。サポートされる最大レコード長 (LRECL) は 255 です。レコードフォーマット (RECFM) は固定でも可変でも可能です。

CMTRACE - Natural トレースのレポート出力 (オプション)

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されます。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、以下のような外部モジュールをロードするためのデフォルトのロードライブラリ名です。

- 共有ニュークリアス (プロファイルパラメータ NUCNAME)
- 別の 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)
- セッションの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すべての外部サブプログラム

ロードライブラリ名はプロファイルパラメータ LIBNAM で変更できます。指定するロードライブラリ名は、DD ステートメントで JCL に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 WORK *nn* や WRITE WORK *nn* 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) が使用できない場合は、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 NETWORK マクロによってデフォルトが定義されます。

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。

39

z/VSE 環境下のバッチモードの Natural

- NATVSE - Natural z/VSE バッチモードインターフェイス 230
- NTVSE マクロ - z/VSE 環境下の Natural の生成パラメータ 230
- z/VSE バッチモードセッションで使用される Natural データセット 236
- z/VSE ライブラリメンバに対する NATVSE 出力およびワークファイルサポート 242
- NATVSE ダイナミックワークファイルアロケーション (DYNALLOC) のサポート 243
- z/VSE 環境下の Natural のデバッグ機能 247
- NATVSE アテンションの中断 251

このドキュメントでは、VSE オペレーティングシステムで Natural をバッチモードで実行する際の特別な考慮事項について説明します。

以下のトピックについて説明します。

バッチモードの Natural 全般に適用される考慮事項については、以下のセクションも参照してください。

- [Adabas データセット](#)
- [ソートデータセット](#)
- [バッチモード環境のサブタスクセッションサポート](#)

NATVSE - Natural z/VSE バッチモードインターフェイス

Natural z/VSE バッチモードインターフェイス NATVSE は、z/VSE 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複数のサービスルーチンで構成されます。

NATVSE はソースモジュールとして提供され、必要に応じてカスタマイズできます。「[z/VSE 環境での Natural のインストール](#)」も参照してください。ここで、生成パラメータを設定できます。アセンブルして Natural ニュークリアスにリンクすることも、単独で実行して共有ニュークリアスに接続することもできます。

NATVSE は 16 MB 境界より下で実行する必要があります。1つのバッチリージョン内で複数のセッションを同時に開始できます。「[バッチモード環境のサブタスクセッションサポート](#)」を参照してください。

NTVSE マクロ - z/VSE 環境下の Natural の生成パラメータ

NTVSE マクロには、NATVSE 内部デフォルトを変更する複数の生成パラメータ（NATVSE コピーブック内で設定）があります。

これらのパラメータは次のとおりです。

NAME | BUFSIZE | CANCEL | DSECTS | FILEID | FILMNGR | FILSCAN | FLUSH | IDUMP | LE370 | LIBRID
| MAXABND | RCSIZE | RJEUSER | SEGMENT | THDSIZE | USERID | WAITIME

NAME - 再配置可能モジュールの名前

NAME は、特定のアセンブリによって作成される再配置可能モジュールの名前を指定します。設定可能値：

NAME=xxxxxx	xxxxxx は、作成される再配置可能モジュールの名前です。 最大長は 8 文字です。
NAME=NATVSE	これはデフォルト値です。

BUFSIZE - Natural I/O バッファのサイズ

BUFSIZE は、すべての入出力操作に使用される Natural I/O バッファのサイズを指定します。

BUFSIZE=nnnn	nnnn は、Natural I/O バッファのサイズ (KB 単位) を指定します。 最小値は 8 (KB) です。
BUFSIZE=8192	これはデフォルト値です。

CANCEL - セッション終了

CANCEL は、セッション終了時の Natural z/VSE インターフェイスの続行方法を指定します。設定可能値：

CANCEL=YES	Natural が正常に終了したか、セッションが Natural TERMINATE ステートメントで終了された場合を除き、ジョブは CANCEL マクロまたは JDUMP マクロによってキャンセルされます。 これはデフォルト値です。
CANCEL=NO	NATVSE は、常に、リターンコードを設定して RETURN または EOJ マクロで終わります (効果はセッションアベンドフラグ UPSI XXXXX1XX を設定した場合と同じ)。



注意: 互換性保持の目的で、CANCEL=YES がデフォルトです。ただし、CANCEL=NO を指定することをお勧めします。特に、VSE 条件ジョブ制御機能を利用する場合にお勧めします。

DSECTS - オペレーティングシステム DSECTS のリスト

DSECTS は、オペレーティングシステム DSECTS をリストするかどうかを指定します。設定可能値：

DSECTS=YES	オペレーティングシステム DSECTS がリストされます。
DSECTS=NO	オペレーティングシステム DSECTS はリストされません。 これはデフォルト値です。

FILEID - ラベル情報の確認

FILEID は、DLBL または TLBL ファイル ID の先頭と照合する最大 8 文字の文字列を指定します。一致した場合は、このラベル情報は無視されます。設定可能値：

FILEID=xxxxxxx	xxxxxxx は任意の文字列で、特殊文字を含む場合は一重引用符で囲む必要があります。
FILEID='IGNORE'	これはデフォルト値です。

これは、**CMWKFnn*** や **CMPRTnn*** の DLBL または TLBL ステートメントが（パーティションの）標準ラベルに指定されているが使用できない場合に特に便利です。

例えば、// DLBL CMPRTO1, '...' ステートメントが指定されている場合、WRITE(1) 出力をプリンタ SPOOL に転送できません。その場合は、JCS ステートメント // DLBL CMPRTO1, 'IGNORE' と、関連する **SYSnnn** の適切なプリンタ割り当てを使用します。

FILMNGR - Natural の出力またはワークファイルの管理

FILMNGR は、Natural の出力またはワークファイルの管理方法を指定します。設定可能値：

FILMNGR=YES	出力またはワークファイルのラベル情報があり、ラベルなしのワークファイルに LABEL=OFF/ON が指定されている場合に、このファイルが使用可能であることを Natural に示します。これは、Natural の出力およびワークファイルがファイル管理システムで管理される場合に特に関連します。 これはデフォルト値です。
FILMNGR=NO	Natural の出力またはワークファイルの論理ユニット番号が、適切なデバイスタイプに割り当てられている必要があります。

FILSCAN - 出力またはワークファイルのスキャン

FILSCAN は、出力またはワークファイルのスキャンするかどうかを指定します。設定可能値：

FILSCAN=YES	Natural z/VSE インターフェイスは、Natural セッションパラメータによって特定のファイルアクセスメソッドが定義されていない、すべての Natural 出力およびワークファイルの z/VSE ラベルエリアをスキャンします。これにより、オーバーヘッドが発生することがあります。 これはデフォルト値です。
FILSCAN=NO	「使用可能」にするためには、すべての Natural 出力およびワークファイルへのアクセスをセッションパラメータで明示的に指定する必要があります。これにより、すべてのファイルアクセス操作が、定義したファイルに集中します。

FLUSH - EOF までのフラッシュカード入力ファイル

FLUSH は、CMSYNIN/CMOBJIN カード入力ファイルによる、セッション終了時の Natural z/VSE インターフェイスの続行方法を指定します。設定可能値：

FLUSH=YES	セッション終了時に、Natural z/VSE インターフェイスは、Natural によって EOF が検出されない限り、SYSIN/SYSRDR/SYSIPT カード入力ファイルを EOF まで読み取ります。そのため、バッチモードの Natural セッションを STACK データで完全に駆動する場合は、空の CMSYNIN/CMOBJIN ファイルに JCL で余分の「/*」を入れる必要があります。 これはデフォルトです。
FLUSH=NO	バッチモードの Natural セッションを STACK データで完全に駆動する場合に、余分の SYSIN/SYSRDR/SYSIPT カード入力ファイル（空ファイル）は必要ありません。SYSIN/SYSRDR/SYSIPT カード入力ファイルはそのまま残るので、CMSYNIN/CMOBJIN が完全に取得されなかった場合は、INVALID STATEMENT 演算子によってメッセージが表示されるか、INVALID STATEMENT によってジョブがキャンセルされます。

IDUMP - ダンプ作成モード

IDUMP は、Natural z/VSE インターフェイスが生成するダンプの種類を指定します。

設定可能値：

IDUMP=YES	Natural z/VSE インターフェイスは、IDUMP マクロを使用してダンプを作成します。 これはデフォルト値です。
IDUMP=NO	Natural z/VSE インターフェイスは、SDUMP マクロを使用してダンプを作成します。

LE370 - IBM 言語環境の使用

LE370 は、Natural を IBM 言語環境で実行するかどうかを指定します。設定可能値：

LE370=YES	IBM 言語環境のランタイム環境は、Natural セッションの初期化時に初期化されます。IBM 言語環境のサブルーチンプログラム（ダイナミックまたはスタティック）が Natural によって呼び出される場合は、「YES」を設定する必要があります。
LE370=NO	IBM 言語環境のランタイム環境は、Natural セッションの初期化時に初期化されません。 これはデフォルト値です。

LIBRID - DLBL ファイル ID 情報の確認

LIBRID は、DLBL ファイル ID の先頭と照合する最大 8 文字の文字列を指定します。一致した場合は、ファイル ID の残りの部分で、z/VSE ライブラリまたはライブラリチェーンのライブラリメンバを指定する情報がスキャンされます。設定可能値：

LIBRID=xxxxxxx	xxxxxxx は 8 文字の任意の文字列で、特殊文字を含む場合は一重引用符で囲む必要があります。
LIBRID='LIBR: '	これはデフォルト値です。

MAXABND - アベンドの最大数

MAXABND は、NATVSE が許容する（つまり、NATVSE がアベンドをインターセプトして Natural アベンドハンドラを呼び出す）アベンドの最大数を指定します。この数を超えると、NATVSE は回復不可能なアベンド状況またはアベンドループと見なして Natural セッションを異常終了します。設定可能値：

MAXABND=nnnn	nnnn は、アベンドの最大数です。
MAXABND=16	これはデフォルト値です。

RCSIZE - サーバー環境用のデフォルトのロールキャッシュサイズ

RCSIZE は、環境初期化要求でロールキャッシュサイズが渡されない場合の、サーバー環境用のデフォルトのロールキャッシュサイズを指定します。設定可能値：

RCSIZE=nnnnnnnnnn	nnnnnnnnnn は、デフォルトのロールキャッシュサイズ (KB 単位) です。
RCSIZE=0	これはデフォルト値です。

RJEUSER - XPCC マクロ要求によるサブミットのユーザー ID

RJEUSER は、XPCC マクロ要求を介してサブミットにユーザー ID を設定するかどうかを指定します。設定可能値：

RJEUSER=YES または RJEUSER=(YES,VSE)	RJEUSER=YES がデフォルト値です。必須のサブミットユーザー ID として、システム変数 *INIT-USER が使用されます。
RJEUSER=(YES,NAT)	必須のサブミットユーザー ID として、システム変数 *USER が使用されます。
RJEUSER=NO	ユーザー ID R000 が使用されます。

SEGMENT - 出力スプールファイルを閉じるときの動作

SEGMENT は、出力 SPOOL ファイル (印刷またはパンチ) を CLOSE するときの Natural z/VSE インターフェイスの動作を指定します。設定可能値：

SEGMENT=YES	ファイルは、そのファイルに対して CLOSE=FIN が有効でない限り、POWER セグメントクローズによって閉じられます。
SEGMENT=NO	SPOOL ファイルは、POWER セグメントクローズなしで閉じられます。 これはデフォルト値です。

THDSIZE - Natural サーバー環境用のデフォルトのスレッドサイズ

THDSIZE は、環境初期化要求でスレッドサイズが渡されない場合の、Natural サーバー環境用のデフォルトのスレッドサイズを指定します。設定可能値：

THDSIZE=nnnnnnnnnn	nnnnnnnnnn は、デフォルトのスレッドサイズ (KB 単位) です。
THDSIZE=0	これはデフォルト値です。

USERID - システム変数 *INIT-USER の内容

このパラメータは、システム変数 *INIT-USER の内容を指定します。設定可能値：

USERID=YES	z/VSE ユーザー ID が JCL で指定されている (// ID USER=xxx) 場合は、このユーザー ID が使用されます。z/VSE ユーザー ID が JCL で指定されておらず、POWER from-user が JECL で指定されている (* \$\$ JOB FROM=xxx) 場合は、このユーザー ID が使用されます。それ以外の場合は、VSE ジョブ名が Natural ユーザー ID として使用されます。
USERID=NO	Natural ユーザー ID として VSE ジョブ名が使用されます。 これはデフォルト値です。

WAITIME - セッションロールアウトのタイムリミット

WAITIME は、タイムリミットをミリ秒単位で指定します。これは、Natural サーバー環境での CMROLL 呼び出しに適用されます。CMROLL 呼び出しで経過した時間が WAITIME の時間以上になると、セッションはロールアウトされ、セッションのスレッドが解放され、セッションは待機状態になります。設定可能値：

WAITIME=nnnnn	nnnnn はミリ秒単位のタイムリミットです。
WAITIME=1000	これはデフォルト値です。

z/VSE バッチモードセッションで使用される Natural データセット

Natural z/VSE バッチモードセッションで特定の機能を使用するには、以下のデータセットが必要です。

データセット	説明
CMEDIT	Software AG Editor ワークファイル
CMHCOPY	ハードコピー出力
CMOBJIN	Natural INPUT ステートメントの入力
CMPLOG	ダイナミックプロファイルパラメータレポート出力
CMPRINT	プライマリレポート出力
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
CMSYNIN	プライマリコマンド入力
CMTRACE	外部トレース出力
CMWKFnn	ワークファイル 01~32

これらのデータセットについては、以下で説明します。

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 で定義されていない場合は、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 のインストール方法については、「*Software AG Editor のインストール*」を参照してください。

CMHCOPY - ハードコピーのレポート出力（オプション）

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、以下のいずれかの方法で変更できます。

- 出力ファイル 0 のプロファイルパラメータ PRINT のサブパラメータ DEST
- プロファイルパラメータ HCDEST (PRINT=((0),DEST=...) と等価)
- セッション時にシステム変数 *HARDCOPY を設定
- セッション時に端末コマンド %H を使用

出力ファイル 0 のプロファイルパラメータ PRINT のサブパラメータを使用して、ハードコピーデータセットのデフォルト値を変更できます。デフォルトのデータセット名 CMHCOPY は暗黙的にハードコピー出力データセットに対する CLOSE=FIN を示します。つまり、出力のためにハードコピー出力データセットを開いた後に名前を変更しても適用されません。開いたときに別の名前が定義されていた場合は、出力ファイル 0 のプロファイルパラメータ PRINT のサブパラメータ CLOSE に従ってハードコピーデータセットは閉じられます。

デフォルトでは、CMHCOPY ファイルは SYSLST に割り当てられ、DTFPR マクロによって処理されます。

ファイル名 CMHCOPY の適切なラベル情報が提供されている場合は、z/VSE マクロ DTFSD または DTFMT を以下の条件で使用して、出力をディスクまたはテープに送信することもできます。

```
RECFORM=UNDEF, BLKSIZE=133
```

z/VSE ライブラリに送信する場合は、レコードフォーマットは固定、レコード長は 80、デフォルトのメンバタイプは PRINT です。

CMOBJIN - Natural INPUT ステートメントの入力

CMOBJIN は、Natural INPUT ステートメントで読み取られるデータに使用されます。このタイプのデータは、関連するソースプログラムや関連する RUN または EXEC コマンドの直後の CMSYNIN 入力ストリームに配置することもできます。

プロファイルパラメータ OBJIN を N に設定すると、Natural は CMSYNIN から入力を読み取ります。OBJIN を Y に設定すると、Natural は CMOBJIN から入力を読み取ります。OBJIN を R に設定すると、Natural は CMOBJIN のラベル情報の有無に基づいて、特定のセッション用に選択されたオプションを判別します。

デフォルトでは、CMOBJIN 入力ファイルは SYSIPT に割り当てられます。これは、プロファイルパラメータ READER を使用して SYSRDR に割り当てることができます。

物理/論理 (POWER) リーダーファイルの代わりに、シーケンシャルディスクまたはラベル付きテープを使用することもできます。その場合は、ファイル名 CMOBJIN の適切なラベル情報を提供する必要があります。

サポートされているファイルフォーマットは以下のとおりです。

```
DTFSD/DTFMT: RECFORM=FIXUNB, RECSIZE=81
DTFSD/DTFMT: RECFORM=FIXUNB/FIXBLK, RECSIZE=80
LIBR:         RECFORM=FIX, RECSIZE=80 , default member type CARD
```

適切なラベル情報を提供する必要があります。割り当てる場合は、ファイル名 CMSYNIN または CMOBJIN を使用する必要があります。

CMPLOG - ダイナミックパラメータのレポート出力 (オプション)

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 が使用できない場合は、評価されたダイナミックプロファイルパラメータは CMPRINT に書き込まれます。

デフォルトでは、CMPLOG ファイルは SYSLST に割り当てられ、DTFPR マクロによって処理されません。

ファイル名 CMPLOG の適切なラベル情報が提供されている場合は、z/VSE マクロ DTFSD または DTFMT を以下の条件で使用して、出力をディスクまたはテープに送信することもできます。

```
RECFORM=UNDEF, BLKSIZE=133
```

z/VSE ライブラリに送信する場合は、レコードフォーマットは固定、レコード長は 80、デフォルトのメンバタイプは PRINT です。

CMPRINT - プライマリレポート出力

CMPRINT は、Natural プログラム内の DISPLAY、PRINT、および WRITE ステートメントの結果をレポートするプライマリ出力に使用されます。

デフォルトでは、CMPRINT ファイルは SYSLST に割り当てられ、DTFPR マクロによって処理されます。

ファイル名 CMPRINT の適切なラベル情報が提供されている場合は、z/VSE マクロ DTFSD または DTFMT を以下の条件で使用して、出力をディスクまたはテープに送信することもできます。

```
RECFORM=UNDEF, BLKSIZE=133
```

z/VSE ライブラリに送信する場合は、レコードフォーマットは固定、レコード長は 80、デフォルトのファイルタイプは PRINT です。

CMPRMIN - ダイナミックパラメータデータセット

CMPRMIN をダイナミックパラメータデータセットとして使用すると、EXEC ステートメントのジョブ制御キーワード PARM の文字列の長さ制限を超えることができます。

このファイルが使用可能な場合は、セッションの初期化時に読み取られ、ダイナミックプロファイルパラメータが取得されます。

CMPRMIN からのすべての入力レコードが、1つのパラメータ文字列に連結されます。各 CMPRMIN レコードの最初の 72 文字目までが有効です。各レコードの末尾の空白は切り捨てられます。空白以外の最後の文字がコンマの場合は、末尾の空白はすべて切り捨てられます。それ以外の場合は、1つの空白がデリミタ文字として残されます。コンマは挿入されません。

PARM ジョブ制御キーワードを使用して、ダイナミックパラメータを追加できます。PARM キーワードにダイナミックパラメータ文字列が含まれている場合は、これらのプロファイルパラメータは CMPRMIN の入力から作成されたパラメータ文字列の末尾に連結されます。これらを使用して、CMPRMIN からのパラメータを上書きできます。PARM キーワードが SYSRDR または SYSIPT として指定されている場合は、Natural は追加のプロファイルパラメータをそれぞれ SYSRDR または SYSIPT から CMPRMIN データセットの論理拡張として取得します。つまり、同じルールが適用されます。

CMPRMIN は、シーケンシャルディスクまたはラベル付きテープデータセットです。サポートされているファイルフォーマットは以下のとおりです。

```
DTFSD/DTFMT: RECFORM=FIXUNB,RECSIZE=81
DTFSD/DTFMT: RECFORM=FIXUNB/FIXBLK,RECSIZE=80
LIBR:         RECFORM=FIX,RECSIZE=80 , default member type CARD
```

CMPRTnn - 追加レポート

CMPRT nn は、セッション中にコンパイルまたは実行された Natural プログラムによって参照される追加の各レポートに使用されます。" nn " は、DISPLAY、PRINT、または WRITE ステートメントで使用されているレポート番号に対応する 01~31 の 2 桁の 10 進数である必要があります。

CMPRT nn の代わりに、プロファイルパラメータ PRINT の DEST サブパラメータを適切な値に設定することで、別の名前を使用できます。以下に例を示します。

```
PRINT=((nn),...,DEST=PRNTFIL)
```

ファイル名 CMPRT nn とともにラベル情報を提供すると、出力をディスクまたはテープに書き込むことができます。Natural は、この出力ファイルを非ブロック形式の固定長ワークファイルと同様に処理します。ディスクまたはテープに「出力」する場合は、ワークファイルの場合と同じロジックが適用されます ([下記参照](#))。

z/VSE ライブラリメンバにマップする場合は、これらのファイルのレコードフォーマットは固定、レコード長は 80、デフォルトのファイルタイプは PRINT です。

CMSYNIN - プライマリ入力

CMSYNIN は、Natural コマンドと Natural ソースプログラム（およびオプションで Natural プログラムの実行中に INPUT ステートメントによって読み取られるデータ）を含むプライマリ入力ファイルに使用されます。

デフォルトでは、CMSYNIN 入力ファイルは SYSRDR に割り当てられます。これは、プロファイルパラメータ READER を使用して SYSIPT に割り当てることができます。

物理/論理 (POWER) リーダーファイルの代わりに、シーケンシャルディスクまたはラベル付きテープを使用することもできます。その場合は、ファイル名 CMSYNIN の適切なラベル情報を提供する必要があります。

サポートされているファイルフォーマットは以下のとおりです。

```
DTFSD/DTFMT: RECFORM=FIXUNB,RECSIZE=81
DTFSD/DTFMT: RECFORM=FIXUNB/FIXBLK,RECSIZE=80
LIBR:         RECFORM=FIX,RECSIZE=80 , default member type CARD
```

CMTRACE - Natural トレースのレポート出力 (オプション)

プロファイルパラメータ ETRACE が "ON" に設定されているか、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。

デフォルトでは、CMTRACE ファイルは SYSLST に割り当てられ、DTFPR マクロによって処理されます。

ファイル名 CMTRACE の適切なラベル情報が提供されている場合は、z/VSE マクロ DTFSD または DTFMT を以下の条件で使用して、出力をディスクまたはテープに送信することもできます。

```
RECFORM=UNDEF, BLKSIZE=133
```

z/VSE ライブラリに送信する場合は、レコードフォーマットは固定、レコード長は 80、デフォルトのメンバタイプは PRINT です。

CMWKFnn - ワークファイル 01~32

CMWKF nn は、セッション中にコンパイルまたは実行された Natural プログラムによって参照される各 Natural ワークファイルに使用されます。 nn は、READ WORK FILE または WRITE WORK FILE ステートメントで使用されているレポート番号に対応する 01~32 の 2桁の 10進数である必要があります。

CMWKF nn の代わりに、プロファイルパラメータ WORK の DEST サブパラメータを適切な値に設定することで、別の名前を使用できます。

Natural z/VSE 生成パラメータ FILMNGR=YES が指定され、ワークファイルのラベル情報が存在する場合、またはラベルなしのワークファイルのプロファイルパラメータ WORK の LABEL サブパラメータに OFF または NOTM が指定されている場合は、Natural はファイルが使用可能であると認識します。それ以外の場合は、Natural ワークファイルの論理ユニット番号が、適切なデバイスタイプに割り当てられている必要があります。

z/VSE ライブラリメンバにマップする場合は、これらのファイルのレコードフォーマットは固定、レコード長は 80、デフォルトのメンバタイプは WORK です。

Natural のプリンタまたはワークファイルが IGN に割り当てられている場合は、これらのファイルに対するすべての I/O 要求はダミーとして処理され、Natural エラーは生成されません。ただ

し、割り当てがない場合、またはプリンタ/ワークファイルがUAに割り当てられている場合は、このファイルを使おうとすると必ずエラーになります。

z/VSE ライブラリメンバに対する NATVSE 出力およびワークファイルサポート

NATVSE は、すべての Natural データセットへの入出力のために z/VSE ライブラリメンバへのアクセスをサポートします。z/VSE ライブラリメンバにアクセスする場合は、「カードイメージフォーマット」（80 バイトのレコード長）のみがサポートされます。

z/VSE ライブラリメンバへのアクセスは、関連付けられた DLBL ステートメントのファイル ID によってトリガされます。DLBL ステートメントのファイル ID フィールドの初めにある特定の文字列（NATVSE 生成パラメータの「LIBRID」を参照）は、実際に Natural データセットがファイル ID フィールドの残りで指定される z/VSE ライブラリメンバであることを示します。

以下の指定が可能です。

C= <i>chain</i>	JCL で定義されたライブラリ連結チェーンを指定します。
S= <i>library.sublib</i>	特定のライブラリ内の特定のサブライブラリを指定します。
M= <i>mbrname.mbrtype</i>	ライブラリメンバ名とそのタイプを指定します。

以下のルールが適用されます。

- 上記の指定はすべて省略可能です。
- 各パラメータは 1 回だけ指定できます。
- 複数のパラメータはコンマまたは空白で区切ります。
- チェーン (C=) およびサブライブラリ (S=) の指定は省略可能ですが、指定する場合はどちらか一方しか指定できません。
- チェーン (C=) もサブライブラリ (S=) も指定しない場合は、デフォルトの C=SOURCE が使用されます。
- ライブラリメンバ (M=) を指定しない場合は、デフォルトの M=*filename.type* が使用されます。

filename は、DLBL ステートメントのファイル名です。

type は、Natural ファイルクラス、つまり Natural ワークファイルの WORK、Natural 出力ファイルの PRINT、Natural 入力ファイル CMPRMIN、CMSYNIN、および CMOBJIN の CARD を示します（各 Natural データセットのデフォルトのメンバタイプについては、以下を参照）。

- ライブラリメンバ指定のサブパラメータに指定されたアスタリスクは、デフォルトが使用されることを示します。したがって、M=*. * という指定は、このパラメータを省略することと同じです。

- メンバタイプのサブパラメータを省略した場合も、デフォルトが使用されます。

例：

```
// LIBDEF PROC,SEARCH=(...)
// LIBDEF SOURCE,SEARCH=(...)
// DLBL CMWKF01,'LIBR:M=FILE1.TEST S=SAGLIB.USRLIB'
// DLBL CMWKF02,'LIBR:          S=SAGLIB.USRLIB'           ->  M=CMWKF02.WORK
// DLBL CMWKF03,'LIBR: M=TEST   C=PROC'                   ->  M=TEST.WORK
// DLBL CMPRT04,'LIBR:M=*.LISTING,S=SAGLIB.USRLIB'        ->  M=CMPRT04.LISTING
// DLBL CMPRT05,'LIBR:'                                     ->
M=CMPRT05.PRINT,C=SOURCE
// DLBL CMPRT06,'LIBR:M=WORK'                               ->
M=WORK.PRINT,C=SOURCE
// DLBL CMWKF07,'LIBR:          M=*.DATA'                 ->
M=CMWKF07.DATA,C=SOURCE
// DLBL CMPRMIN,'LIBR:M=*. *'                              ->
M=CMPRMIN.CARD,C=SOURCE
```



注意:

1. 出力ファイルにチェーンが指定されているかデフォルトが使用される場合は、出力はチェーン内に指定されている最初のサブライブラリに書き込まれます。
2. Natural 出力ファイルのターゲットサブライブラリ内に同名で同タイプのメンバがすでに存在する場合は、無条件で既存のメンバが置き換えられます。
3. DLBL ステートメントのファイル ID フィールドは長さが 44 文字分しかなく、すべての（サブ）パラメータを完全に示すには十分ではありません。このため、デフォルトを活用することをお勧めします。メンバ名については、Natural プロファイルパラメータ PRINT または WORK の DEST サブパラメータを使用してファイル名を指定するオプションもあります。

NATVSE ダイナミックワークファイルアロケーション (DYNALLOC) のサポート

z/VSE 環境の Natural には、ワークファイルをダイナミックに定義する機能があります。つまり、これらのファイルを JCL であらかじめ定義しておく必要がありません。z/VSE 環境の Natural では、DEFINE WORK FILE ステートメントを使用して定義したワークファイルのパーティションの一時ラベルエリアに、ラベルを追加します。

このため、z/VSE 環境の Natural では、既存のラベル情報は変更されません。Natural セッションによってダイナミックに追加されたすべてのファイルラベルは、セッション終了時に削除されます。

以下では次のトピックについて説明します。

- 前提条件
- DEFINE WORK FILE キーワードパラメータ
- DEFINE WORK FILE キーワードパラメータの使用ルール
- 例

前提条件

Natural アプリケーションプログラム内でファイルエクステンション情報を修正することはできないため、特にこれらのプログラムが同じパーティションまたは複数のパーティションで同時に実行される場合は、ディスクファイルマネージャが必要です。z/VSE 環境で少なくとも VSAM/SAM は使用可能なので、これは制限ではありません。

z/VSE 環境での Natural によるダイナミックアロケーションサポートのために、Natural プロファイルパラメータ WORK の以下のキーワードサブパラメータが使用できるようになりました。

- BLOCKS (ストレージブロック数)
- DISP (ファイルオープンモード)

DEFINE WORK FILE キーワードパラメータ

以下のキーワードパラメータは、VSE 環境でのダイナミックワークファイルに使用できます。

キーワードパラメータ	目的
CAT= <i>catalog</i>	ダイナミックワークファイルに対する VSAM/SAM の使用をトリガします。カタログは、1~7 文字の VSAM カタログファイル名です。VSAM/SAM を使用する場合は、このパラメータは必須で、VSAM ジョブカタログを明示的に指定 (CAT=IJSYSUC) する必要があります。
VOL= <i>volser</i>	これを指定した場合は、// EXTENT 情報が 1~5 文字のボリュームシリアル番号で生成されます。情報の内容は、先行する DLBL 情報に基づきます (下記参照)。
DSN= <i>fileid</i>	ファイル ID を設定する DSN (オプション) です。

DEFINE WORK FILE キーワードパラメータの使用ルール

パラメータ値の最後がわかりやすいように、CAT または VOL パラメータは DEFINE WORK FILE 文字列の最初に指定し、DSN= パラメータは最後のキーワードとして指定する必要があります。

つまり、CAT または VOL パラメータが指定され、DSN キーワードパラメータが指定されていない場合は、最後のキーワードの後にくる DEFINE WORK FILE パラメータ文字列のすべてのデータは設定されるファイル ID と見なされます。

VSAM/SAM の場合は、NATVSE は以下のラベル情報をダイナミックに追加します。

```
// DLBL xxyyyyz,'file-id',0,VSAM,CAT=catalog, +
  RECORDS=n1,RECSIZE=n2,DISP=(dsp1,dsp2)
// EXTENT ,volser
  optional
```

上記の意味は以下のとおりです。

xx パーティションの SYSLOG ID。BG や F4 など。

yyyy 編集された z/VSE の 2 バイトの 16 進タスク番号 (同じパーティションで複数の Natural サブタスクを可能にするため)。

z Natural ワークファイル番号。ファイル 1~9 には "1"~"9"、ファイル 10~32 には "A"~"W"。

n1 プロファイルパラメータ WORK のキーワードサブパラメータ BLOCKS によって指定される値。

n2 プロファイルパラメータ WORK のキーワードサブパラメータ BLKSIZE によって指定される値。

dsp1, dsp2 プロファイルパラメータ WORK のキーワードサブパラメータ DISP によって指定される値。

catalog DEFINE WORK FILE ステートメントを使用して設定する必要がある VSAM カタログ。

volser ファイルの割り当て先のボリュームシリアル番号。

EXTENT カードが生成されるのは、*volser* が Natural アプリケーションの DEFINE WORK FILE ステートメントに設定されている場合のみです。

CA-DYNAM/D などの他のディスクファイル管理システムでは、NATVSE は以下のラベル情報をダイナミックに追加します。

```
// DLBL xxyyyyz,'file-id',0
// EXTENTsysnnn,volser,,,1,n1 optional
```

上記の意味は以下のとおりです。

xx パーティションの SYSLOG ID。BG や F4 など。

yyyy 編集された z/VSE の 2 バイトの 16 進タスク番号 (同じパーティションで複数の Natural サブタスクを可能にするため)。

z Natural ワークファイル番号。ファイル 1~9 には 1~9、ファイル 10~32 には A~W。

n1 プロファイルパラメータ WORK のキーワードサブパラメータ BLOCKS によって指定される値。

sysnnn プロファイルパラメータ WORK のキーワードサブパラメータ SYSNR によって指定される値。

volser ファイルの割り当て先のボリュームシリアル番号。

EXTENT カードが生成されるのは、*volser* が Natural アプリケーションの DEFINE WORK FILE ステートメントに設定されている場合のみです。*n1*が0の場合は、エクステント情報（開始トラック/ブロック、およびトラック/ブロックの数）は省略されます。

サーバーとしての Natural の場合は、ファイル名は以下のフォーマットで設定します。

```
XYYYYYZ
```

上記の意味は以下のとおりです。

xyyyyyy 16 進形式のサーバーセッション番号（編集済み）。最初の文字は 0~F を A~P に変換することで強制的にアルファベットにします。

z z/VSE バッチ処理環境の通常の Natural の Natural ワークファイル番号。

z/VSE 環境の Natural で上記のような新しいファイル名を作成できるようにするには、ダイナミックに割り当てられるファイルが、プロファイルパラメータ WORK のキーワードサブパラメータ DEST='*' またはステートメント定義の DEFINE WORK FILE '*' で設定される '*' という名前を持つ必要があります。

使用するファイル ID も、DEFINE WORK FILE ステートメントを使用して設定する必要があります。

ファイル ID に対する通常の z/VSE 制限が適用されます。

Natural マルチタスクまたはサーバー環境では、「ファイル条件が同じ」にならないように、一意の情報をファイル ID に指定することをお勧めします。

例

Natural パラメータ：

```
WORK=((1-6),AM=1,DEST='*',BLOCKS=100),WORK=((2),  
DISP=(OLD,DELETE)),WORK=((6),BLOCKS=0)
```

アプリケーション内での **Natural** ワークファイル定義：

```
DEFINE WORK 1 'CAT=IJSYSUC,VSAM.SAM.FILE'
DEFINE WORK 2 'CAT=IJSYSCT,DSN=ANOTHER FILE'
DEFINE WORK 3 'CAT=IJSYSUC,VOL=DOSRES,ONE MORE FILE'
DEFINE WORK 4 '==.CATALOGED.FILE'
DEFINE WORK 5 'VOL=POOL01,DSN=FILE WITH EXTENT INFO'
DEFINE WORK 6 'VOL=DOSRES,ANY FILE'
```

生成された z/VSE ラベル：

```
// DLBL xxyyyy1,'VSAM.SAM.FILE',0,VSAM,CAT=IJSYSUC, +
RECORDS=100,RECSIZE=4628,DISP=(NEW,KEEP)
// DLBL xxyyyy2,'ANOTHER FILE',0,VSAM,CAT=IJSYSCT, +
RECORDS=100,RECSIZE=4628,DISP=(OLD,DELETE)
// DLBL xxyyyy3,'ONE NORE FILE',0,VSAM,CAT=IJSYSUC, +
RECORDS=100,RECSIZE=4628,DISP=(NEW,KEEP)
// EXTENT ,DOSRES
// DLBL xxyyyy4,'==.CATALOGED.FILE',0
// DLBL xxyyyy5,'FILE WITH EXTENT INFO',0
// EXTENT SYS005,POOL01,,1,100
// DLBL xxyyyy6,'ANY FILE',0
// EXTENT SYS006,DOSRES
```

z/VSE 環境下の Natural のデバッグ機能

Natural z/VSE バッチモードインターフェイスには、問題を突き止めるのに役立ついくつかのデバッグ機能が含まれています。

これらの機能は、JCL の UPSI 設定で制御されます。

UPSI 設定は、Natural セッションパラメータ (UPSI=1XXXXXX など) として指定することもできます。これは、JCL の UPSI 設定が、フロントエンド Natural や Natural によって呼び出されるプログラムなど別のプログラムには異なる意味を持つという点で、副次的な効果を生成する場合に特に役立ちます。

以下の UPSI 設定が可能です。

UPSI 設定	意味
UPSI 1XXXXXXXX	ダンプフラグ
UPSI X1XXXXXXXX	トレースフラグ
UPSI XXX1XXXX	ストレージフリーズフラグ
UPSI XXXXX1XX	セッションアベンドフラグ
UPSI XXXXXX1X	アベンド終了フラグ
UPSI XXXXXXXX1	フォーマットされたダンプのみフラグ

これらの設定については、以下で説明します。また、[デバッグ用ドキュメント](#)の取得方法を説明するジョブ例も示します。

UPSI 1XXXXXXXX - ダンプフラグ

通常、Naturalで問題が発生した場合は、実際にアベンドが発生しない限り、対応するジョブはダンプなしでキャンセルされます。この UPSI フラグを設定すると、エラーが発生した場合は、つまり Natural セッションの終了メッセージが NAT9995 以外の場合は、ジョブの終了時に必ずダンプが作成されます。

UPSI X1XXXXXXXX - トレースフラグ

このフラグを設定すると、Naturalの特定の時点でレジスタセーブエリアのスナップショットが取得されます。

 **注意:** 出力例によっては、このフラグを設定することで大量の出力が発生する場合があります。

すべての NATVSE サービスルーチンの入口で、このルーチンの名前と汎用レジスタ 0~15 (GRG) の名前が表示されます。

 **注意:** レジスタ 14 から呼び出し元を特定することができます。

すべての NATVSE サービスルーチンの出口で、このルーチンの名前、現在の汎用レジスタ (GRG)、および現在割り当てられているセーブエリア (CSA) のレジスタ 0~15 が表示されます。

 **注意:**

1. CSAの内容は、汎用レジスタから取得されるレジスタ15のリターンコードを除いて、サービスルーチンの呼び出し元に返されます。
2. HSAの内容は呼び出し元に返されます。これは、このセーブエリアにレジスタ15のリターンコードが含まれている（リターンコードが設定されている場合）ことを意味します。

GRG レジスタが設定されている場合は必ず、デバッグトレースプログラムは呼び出し元ルーチンの名前とルーチンの先頭からの呼び出しのオフセットを特定しようとします。

デバッグトレース出力の `SYSnnn` 番号は、この `SYSnnn` 番号がプリンタデバイスに割り当てられていれば、`SYS040` です。それ以外の場合は、`SYSLST` が使用されます。これは、デバッグトレースの出力とその他の Natural 出力を分ける場合に特に役立ちます。分ける場合は、`SYS040` を適切に割り当て、この論理出力ユニットに `POWER * $$$ LST` ステートメントを指定します。

UPSI XXX1XXXX - ストレージフリーズフラグ

Natural のデフォルトでは、正常終了でも異常終了でも、セッション終了時にはストレージを含むすべてのリソースが解放されます。UPSI 1 を設定しても、Natural の終了時には関連するストレージがすべて解放されているため、ダンプが役に立たないことがあります。このフラグを設定すると、以前に取得された GETVIS ストレージがこのジョブ内では解放されなくなります。これは、Natural ニュークリアス (NATVSE にリンクされていない場合) および `RCA=ON` のサブ製品を含む Natural によって呼び出されるすべての外部サブルーチンプログラムに適用されます。

 **注意:** このフラグは慎重に取り扱ってください。より多くのパーティションの GETVIS ストレージが使用されますが、オペレーティングシステムのストレージ要求を満たせない場合は、GETVIS 要求が失敗し、ジョブがキャンセルされることがあるからです。

UPSI XXXXX1XX - セッションアベンドフラグ

デフォルトでは、重大なエラーが発生した場合 (`NAT9995` および `NAT9987` を除く `NAT9nnn` 終了メッセージ)、Natural セッションはキャンセルされます。このフラグを設定すると、Natural はキャンセルせず、ジョブ制御に Natural リターンコードを渡して「正常に」終了します。

UPSI XXXXXX1X - アベンド終了フラグ

このフラグは、アベンドが繰り返し発生する場合に役立ちます。

通常、バッチモードでは、プログラムチェック (`NAT095n` エラーメッセージ) から回復するために、Natural はアクティブなプログラム (`STXIT PC`) に対するチェックアベンド出口を持ちます。DU=ON が指定されている場合は、この出口によってスナップダンプが作成され、クリーンセッション終了のために制御が Natural に渡されます。

このフラグを設定すると、Natural セッションはアクティブなプログラムに対するアベンド出口なしに実行されます。これは、すべてのプログラムチェックがオペレーティングシステムによって直接処理されることを意味します。

このフラグが設定されていると、ダンプフラグ、ストレージフリーズフラグ、セッションアベンドフラグ、およびフォーマットされたダンプのみフラグは無視されます。

UPSI XXXXXX1 - フォーマットされたダンプのみフラグ

DU=ON の場合は、NATVSE アベンド出口ルーチンは、プログラムチェックアベンドの発生時（および UPSI XXXXXX1X フラグが設定されていない場合）に、Natural セッションのスナップダンプを作成します。

- 失敗した命令、プログラムチェックコード (SOcn)、汎用レジスタ、現在アクティブなルーチン、このルーチン内での失敗した命令のオフセット、および絶対 (PSW) アドレスが、現在割り当てられているセーブエリア (CSA) のレジスタ 0~15 とともに表示されます。
- さらに、非リエントラントの Natural z/VSE ドライバ、Natural によって GETMAIN されたすべてのエリア、およびバッファプール内のすべての Natural プログラムがダンプされます。
- その後、クリーンセッション終了のために制御が Natural に渡されます。
- 最後に、z/VSE JDUMP マクロによってジョブが終了され、ダンプにはすべてのパーティションが含まれます。

多くの場合は、デバッグに関連するのはダイナミック Natural セッションエリアのみであるため、UPSI フラグの設定によってスタティックセッションエリアのダンプは抑制されます。

デバッグ用ドキュメントの取得

問題を分析する必要がある場合は、関連する可能性がある情報、特に、実行された JCS および対応するコンソールログが重要となります。

以下のジョブ例では、包括的なドキュメントの取得方法を示します。

```
// JOB sampljob
// OPTION LOG,PARTDUMP to see JCL on printer
/* Library Definitions: labels and LIBDEFS
...
/* ADARUN Parameter Input Definition
// ASSGN SYS000,SYSRDR
/* Natural Work File Definitions
// DLBL CMWKFnn,'...',... disk work file
// EXTENT SYSnnn,volser,,nn,mm
// ASSGN SYSnnn,DISK,VOL=volser,SHR
// TLBL CMWKFnn,'...',... labelled tape work file
// ASSGN SYSnnn,uuu assignment to tape unit
/* Natural Print File Definitions
// ASSGN SYSnnn,uuu assignment to print UR unit
// DLBL CMPRTnn,'...',... print file on disk
// EXTENT SYSnnn,volser,,nn,mm
// ASSGN SYSnnn,DISK,VOL=volser,SHR
// TLBL CMPRTnn,'...',... print file on labelled tape
// ASSGN SYSnnn,uuu assignment to tape unit
/* Debugging Options
// ASSGN SYS040,SYSLST debugging trace unit
// UPSI lxxx00xx flags as discussed above
```

```
// EXEC Natural,SIZE=...  
... dynamic parameters  
/* end of dynamic parameters  
... ADARUN parameters  
/* end of ADARUN parameters  
... Natural input  
/* end of Natural input  
// EXEC LISTLOG print console messages  
/& end of job
```

NATVSE アテンションの中断

Natural z/VSE バッチモードインターフェイス (NATVSE) は、コンソールコマンド MSG *xx* によってアテンションの中断をサポートします。*xx* は、コンソールのオペレータが NAT1016 アテンション中断イベントを強制できる z/VSE パーティション ID です。

この特殊な機能は、Natural プロファイルパラメータ ATTN で制御されます。

40 CMS 環境下のバッチモードの Natural

- CMS 環境下のバッチモードによる Natural の実行 254

このドキュメントでは、CMS オペレーティングシステムで Natural をバッチモードで実行する際の特別な考慮事項について説明します。

バッチモードの Natural 全般に適用される考慮事項については、以下のセクションを参照してください。

- *Adabas* データセット
- ソートデータセット

CMS 環境下のバッチモードによる Natural の実行

Natural をダイナミックパラメータ BATCH を使用して起動すると、バッチモードになります。

Natural は端末とは通信しませんが、DD 名が CMSYNIN のデータセットから入力を取得し、DD 名が CMPRINT のデータセットに出力を送信します。これらのデータセットについては、以下で説明します。

Natural をバッチモードで起動する前に、これらの DD 名に対して FILEDEF コマンドを実行しておく必要があります。

BATCH を他のダイナミックパラメータと組み合わせて指定する場合は、以下の例に示すように、BATCH を最初のパラメータにする必要があります。

```
FILEDEF CMPRINT PRINTER
FILEDEF CMSYNIN DISK BATCH INPUT A
NATvr BATCH, FNAT=(10,13), FUSER=(132,12)
```

この例では、*vr* は現在のバージョンとリリース番号です。

詳細な例については、NATBATCH EXEC と NATINPL EXEC を参照してください。

CMPRINT - プライマリレポート出力

CMPRINT は、Natural プログラム内の DISPLAY、PRINT、および WRITE ステートメントの結果をレポートするプライマリ出力に使用されます。

CMSYNIN - プライマリ入力

CMSYNIN は、Natural コマンドと Natural ソースプログラム（およびオプションで Natural プログラムの実行中に INPUT ステートメントによって読み取られるデータ）を含むプライマリ入力ファイルに使用されます。

実際に処理される行当たりの文字数は、プロファイルパラメータ SL の現在の設定によって決まります。この設定は、ソースステートメントと実行時入力データの両方に適用されます。これにより、必要に応じて、ID またはシーケンス番号を各レコードの右端の列に配置することができます。

41 BS2000/OSD 環境下のバッチモードの Natural

■ BS2000/OSD バッチモードセッションで使用される Natural データセット	258
■ キーワードパラメータ	262
■ BS2000/OSD ジョブ変数	271

このドキュメントでは、BS2000/OSD オペレーティングシステムで Natural をバッチモードで実行する際の特別な考慮事項について説明します。

以下のトピックについて説明します。

「BS2000/OSD 環境下の Natural のバッチエラーメッセージ」も参照してください。

バッチモードの Natural 全般に適用される考慮事項については、以下のセクションを参照してください。

- **Adabas** データセット
- ソートデータセット
- バッチモード環境のサブタスクセッションサポート

BS2000/OSD バッチモードセッションで使用される Natural データセット

Natural BS2000/OSD バッチモードセッション時には、以下のシーケンシャルデータセットがオプションで使用されます。

データセット	説明
CMPRMIN	ダイナミックパラメータデータセット
CMSYNIN	プライマリ入力
CMOBJIN	Natural INPUT ステートメントの入力
CMPRINT	プライマリレポート出力
CMPLOG	ダイナミックパラメータのレポート出力 (オプション)
CMTRACE	Natural トレースのレポート出力 (オプション)
CMHCOPY	ハードコピーのレポート出力 (オプション)
CMPRT	追加レポート
CMWKF	Natural ワークファイル

CMPRMIN - ダイナミックパラメータデータセット

CMPRMIN をダイナミックパラメータデータセットとして使用すると、EXEC ステートメントのジョブ制御キーワード PARM の文字列の長さ制限を超えることができます。

CMPRMIN からのすべての入力レコードが、1つのパラメータ文字列に連結されます。各 CMPRMIN レコードの最初の 72 文字目までが有効です。各レコードの末尾の空白は切り捨てられます。コンマは挿入されません。

PARM ジョブ制御キーワードを使用して、ダイナミックパラメータを追加することができます。追加したダイナミックパラメータは、CMPRMIN からのパラメータの後に連結されます。つまり、PARM 文字列を使用して、CMPRMIN データセットで指定されたダイナミックパラメータを上書きできます。

ダイナミックパラメータの読み取りの詳細については、マクロ NAMBS2 のキーワードパラメータ DYNPAR=FILE を参照してください（「[DYNPAR](#)」を参照）。

CMSYNIN - プライマリ入力

CMSYNIN は、Natural コマンドと Natural ソースプログラム（およびオプションで Natural プログラムの実行中に INPUT ステートメントによって読み取られるデータ）を含むプライマリ入力ファイルに使用されます。

実際に処理される行当たりの文字数は、プロファイル/セッションパラメータ SL の現在の設定によって決まります。この設定は、ソースステートメントと実行時入力データの両方に適用されます。これにより、必要に応じて、ID またはシーケンス番号を各レコードの右端の列に配置することができます。

```
/SYSFILE SYSDTA=file.
```

CMOBJIN - Natural INPUT ステートメントの入力

CMOBJIN は、Natural INPUT ステートメントで読み取られるデータに使用されます。このタイプのデータは、関連するソースプログラムや関連する RUN または EXEC コマンドの直後の CMSYNIN データセットに配置することもできます。

プロファイルパラメータ OBJIN を "N" に設定すると、Natural は CMSYNIN から入力を読み取ります。OBJIN を "Y" に設定すると、Natural は CMOBJIN から入力を読み取ります。

OBJIN を "R" に設定すると、Natural は CMOBJIN の DD ステートメントの有無に基づいて、特定のセッション用に選択されたオプションを判別します。

CMOBJIN に関する DCB の考慮事項は **CMSYNIN** と同じです。レコードの読み取りでは、プロファイル/セッションパラメータ SL で指定された最大文字数が解釈されます。エラーが発生した場合は、プロファイル/セッションパラメータ CC の設定に従って対処されます。

```
/SYSFILE SYSDTA=file
```

`/SYSFILE SYSIPT=file` は、ダイナミックセッションパラメータを読み取れるように設定する必要があります。ファイル当たり最大 3 行がダイナミックセッションパラメータとして読み取られます。

CMPRINT - プライマリレポート出力

CMPRINT は、Natural プログラム内の DISPLAY、PRINT、および WRITE ステートメントの結果をレポートするプライマリ出力に使用されます。

```
/SYSFILE SYSOUT=file
```

または

```
/SYSFILE SYSLST=file
```

GLOBAL SET パラメータ &WRITE の値に基づきます。

CMPLLOG - ダイナミックパラメータのレポート出力 (オプション)

プロファイルパラメータ PLOG が "ON" に設定されている場合は、すべての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。CMPLLOG が定義されていない場合は、代わりに CMPRINT が使用されます。

```
/SYSFILE SYSOUT=file
```

または

```
/SYSFILE SYSLST=file
```

GLOBAL SET パラメータ &WRITE の値に基づきます。

CMTRACE - Natural トレースのレポート出力 (オプション)

プロファイルパラメータ ETRACE が ON に設定されている場合は、すべてのトレース出力はセッション中にこのデータセットに書き込まれます。

```
/SYSFILE SYSOUT=file
```

または

```
/SYSFILE SYSLST=file
```

GLOBAL SET パラメータ &WRITE の値に基づきます。

CMHCOPY - ハードコピーのレポート出力 (オプション)

ハードコピーの出力先。このデータセットの名前は、プロファイルパラメータ HCDEST または端末コマンド %H を使用して変更できます。

```
/SYSFILE SYSOUT=file
```

または

```
/SYSFILE SYSLST=file
```

GLOBAL SET パラメータ &WRITE の値に基づきます。

CMPRTnn - 追加レポート

CMPRT nn は、セッション中にコンパイルまたは実行された Natural プログラムによって参照される追加の各レポートに使用されます。 nn は、DISPLAY、PRINT、および WRITE ステートメントで使用されているレポート番号に対応する 01~31 の 2 桁の 10 進数である必要があります。

CMPRT nn の代わりに、プロファイルパラメータ PRINT の DEST サブパラメータを適切な値に設定することで、別の名前を使用できます。以下に例を示します。

```
PRINT=((nn),...,DEST=PRINTFIL)
```

DCB 情報は省略可能です。省略した場合は、プロファイルパラメータ PRINT、サブパラメータ RECFM、LRECL、および BLKSIZE で提供されます。RECFM を指定していない場合は、"VBA" に設定されます。DCB 情報が提供されていない場合は、以下の処理が適用されます。

```
/FILE file, LINK=Pnn.
```

CMWKFnn - Natural ワークファイル

CMWKFnn は、セッション中にコンパイルまたは実行された Natural プログラムによって参照される各 Natural ワークファイルに使用されます。nn は、READ WORK FILE または WRITE WORK FILE ステートメントで使用されているレポート番号に対応する 01~32 の 2 桁の 10 進数である必要があります。

CMWKFnn の代わりに、プロファイルパラメータ WORK のサブパラメータ DEST を適切な値に設定することで、別の名前を使用することができます。

DCB 情報は出力ファイル用で省略可能です。省略した場合は、プロファイルパラメータ WORK、サブパラメータ RECFM、LRECL、および BLKSIZE で提供されます。DCB 情報が提供されていない場合は、以下の処理が適用されます。

```
/FILE file, LINK=Wnn
```

キーワードパラメータ

Natural BS2000/OSD バッチモードドライバは、NAMBS2 マクロをアセンブルして生成されます。ドライバモジュールの条件付きアセンブルの制御には、以下のキーワードパラメータを使用できます。

パラメータは名前のアルファベット順に示しています。

- ADACOM
- ADDBUFF
- APPLNAM
- CODE
- DELETE
- DYNPAR
- ILCS
- JV
- LF
- LINK
- LINK2/LINK3/LINK4
- NUCNAME
- PARMOD

- REQMLOC
- SYSDTA
- TERM
- TRACE
- USERID
- WRITE

ADACOM

設定可能値：	ADACOM=ADAUSER、ADACOM=ADABAS、ADACOM=ADALNK
デフォルト値：	ADACOM=ADALNK

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。使用する Adabas リンクモジュールを決定します。設定可能値：

ADACOM=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)。
ADACOM=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)。
ADACOM=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。

ADDBUFF

設定可能値：	1~8
デフォルト値：	なし

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。
 端末 I/O バッファの追加のページ数 (4 KB 単位) を決定します。

APPLNAM

設定可能値：	<i>name</i>
デフォルト値：	NATBS2

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。

name は、Natural バッチアプリケーションの名前 (最大 8 文字) です。この名前は、Natural バッチタスクが初期化される際のシリアライゼーション ID の一部です。

CODE

設定可能値：	FRONT/RENT
デフォルト値：	FRONT

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。

生成する Natural BS2000/OSD インターフェイスの部分を決定します。

CODE=FRONT	フロントエンド部分を生成／アセンブルすることを示します。
CODE=RENT	リエントラント部分を生成／アセンブルすることを示します。

DELETE

設定可能値：	ON/OFF
デフォルト値：	ON

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。

DELETE=ON	Natural パラメータモジュール内のプロファイルパラメータ DELETE の設定は、ダイナミックにロードされた非 Natural プログラムをロード先の Natural プログラムの終了時にアンロードするかどうか、またコマンドモードに入ったときにアンロードするかどうかを決定します。
DELETE=OFF	いったんダイナミックにロードされた非 Natural プログラムは、Natural セッション全体を通じて保持されます。

DYNPAR

設定可能値：	SYSDTA/SYSIPT/FILE/NO
デフォルト値：	NO

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。

DYNPAR=NO	ダイナミックパラメータは読み取られません。
DYNPAR=SYSDTA	ダイナミックパラメータは SYSDTA から読み取られます。SYSDTA が SYSCMD に割り当てられている場合は、/EXEC Natural カードの後に少なくとも 1 つの /EOF カードが必要です。 例：

	<pre> /LOGON /SYSFILE SYSDTA=(SYSCMD) /EXEC NATBAT /EOF * Null dynamic parameters LOGON SYSEXTP L * * FIN /LOGOFF </pre>
DYNPAR=SYSIPT	ダイナミックパラメータは SYSIPT から読み取られます。
DYNPAR=FILE	<p>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 ファイルの入力は単一のテキスト文字列として解釈されます。つまり、個々のエントリは行末にある場合でもコンマで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。</p> <p>例：</p> <pre> /FILE NAT.PARAMS, LINK=CMPRMIN </pre>

ILCS

設定可能値：	YES/NO/CRTE
デフォルト値：	NO

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。

ILCS=CRTE	<p>3GL サブプログラムは、共通ランタイム環境規則を使用して起動されます。これを可能にするには、ILCS 初期化ルーチン ITOSL# が Natural フロントエンドにリンクされている必要があります。</p> <pre> INCLUDE ITOSL#,SYSLNK.CRTE.010 RESOLVE,SYSLNK.CRTE.010 </pre>
ILCS=YES	<p>3GL サブプログラムは、拡張 ILCS リンケージ規則を使用して起動されます。これを可能にするには、ILCS 初期化ルーチン ITOINITS が Natural フロントエンドにリンクされている必要があります。</p> <pre> INCLUDE ITOINITS,SYSLNK.ILCS RESOLVE,SYSLNK.ILCS </pre>
ILCS=NO	標準の処理が適用されます。

JV

設定可能値：	ON/OFF
デフォルト値：	OFF

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。

JV=ON	ジョブ変数がリンク名 *NATB2JV で宣言されていれば、Natural セッションの終了時に作成されるコンディションコードがそのジョブ変数に渡されます。
JV=OFF	BS2000/OSD インストールに Siemens 製品の「ジョブ変数」が含まれていない場合は、このパラメータを OFF に設定する必要があります。そうしないと、NAMBS2 コンパイルでアセンブルエラーが発生します。

LF

設定可能値：	X'zz'
デフォルト値：	X'25'

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。

このパラメータを使用して、ローカルプリンタで印刷するときに行送りに使用される制御文字を指定します。

LINK

設定可能値：	<i>name</i> (<i>name,name,...</i>)
デフォルト値：	なし

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。

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 *name(s)* は、このパラメータで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、指定したプログラムおよびモジュールのために実行され、プログラムおよびモジュールのロードアドレスをダイナミックローダーのリンクテーブルに入力します。したがって、Natural プログラムから呼び出されたときに、これらのプログラムをダイナミックにロードする必要がありません。ダイナミックにロードされるプログラムについては、Natural パラメータモジュール内でロードライブラリを定義する必要があるだけです。

例：

```
LINK=PROG1
LINK=(PROG1,PROG2,MODUL111)
```

LINK2/LINK3/LINK4

設定可能値：	<i>name</i> (<i>name,name,...</i>)
デフォルト値：	なし

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 の構文と似ていません。

例：

```
NAMBS2 LINK=(PROG1,PROG2,...),
LINK2=(PROG54,...)
NAMBS2 LINK=(PROG1,PROG2,PROG3,PROG4)
```

NUCNAME

設定可能値：	<i>name</i>
デフォルト値：	NB2RENT

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。ADDON マクロの Natural プールおよびロード情報には、この名前を使用する必要があります（ADDON マクロは BS2STUB をアセンブルします）。

PARMOD

設定可能値：	(<i>nn,loc</i>) <i>nn</i> : 24/31 <i>loc</i> : BELOW/ABOVE
デフォルト値：	(31, ABOVE)

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 BS2000/OSD アプリケーションの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。

Natural バッファプール、Natural BS2000/OSD アプリケーションのリエントラント部分、Adabas または Adabas Fast Path プールが 16 MB より上にある場合は、31 ビットモードが必要です。

このパラメータの2番目の部分 (*loc*) は、Natural BS2000/OSD アプリケーションのフロントエンド部分の位置の定義に使用されます。16 MB より上のアプリケーションのフロントエンド部分をロードする場合は、このパラメータを以下のようにフロントエンド部分のリンクプロシージャ内に定義する必要があります。

```
LOADPT=*XS
```

または

```
LOADPT=X'address'
```

例：

```
/EXEC TSOLINK
PROG NAT230, FILENAM=NAT230, LOADPT=*XS, ...
TRAITS RMODE=ANY, AMODE=31
INCLUDE ...
/* PARMOD=(nn,loc) MUST BE IDENTICAL IN THE FRONT-END AND REENTRANT PARTS
```

REQMLOC

設定可能値：	RES/BELW/ABOVE
デフォルト値：	RES

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、要求された Natural ワークエリアを、要求メモリを使用してシステムで割り当てる位置を決定します。

REQMLOC=BELOW	すべてのエリアは 16 MB より下で要求されます。
REQMLOC=ABOVE	すべてのエリアは 16 MB より上で要求されます。
REQMLOC=RES	すべてのエリアは、リエントラント部分の位置に基づいて要求されます。

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

設定可能値：	PRIMARY/SYSCMD
デフォルト値：	PRIMARY

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。

SYSDTA=PRIMARY	SYSDTAからのダイナミックパラメータの読み取りが終了した後、SYSDTAはSYSFILE SYSDTA=(PRIMARY)に設定されます。
SYSDTA=SYSCMD	SYSDTAからのダイナミックパラメータの読み取りが終了した後、SYSDTAはSYSFILE SYSDTA=(SYSCMD)に設定されます。

TERM

設定可能値：	PRGR/STEP
デフォルト値：	PRGR

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。

TERM=PRGR	Natural バッチアプリケーションは終了します。
TERM=STEP	システムはさらに次の SET-JOB-STEP コマンドを実行します。

TRACE

設定可能値：	<i>nn, ll</i> <i>nn</i> : 01~99 <i>ll</i> : 71~132
デフォルト値：	99,71

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。

このパラメータを使用して、トレースファイルの数とトレース出力レコードの最大長を指定します。 *nn* は SYSLST*nn* トレースファイルの数、 *ll* はトレース出力レコードの最大長（文字数）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。この場合は、Natural バッチモードドライバにより、以下のようなトレースファイルが作成されます。

例：

```
NATURAL.TRACE.BTCH.7777,SPACE=(30,3)
SYSFILE SYSLSTnn=Natural.TRACE.BTCH.7777
/* 7777 is the task sequence number
```

Natural バッチモードセッションが終了する前に、トレースファイルは以下のようにして閉じられます。

```
SYSFILE SYSLSTnn=(PRIMARY)
```

USERID

設定可能値：	YES/SYSTEM/NO/USER
デフォルト値：	USER

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。

USERID=SYSTEM または USERID=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。
USERID=USER または USERID=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。BS2000/OSD ジョブ名が LOGON コマンドで指定されていない場合は、Natural ユーザー ID は USERID=SYSTEM または YES によって作成されます。

WRITE

設定可能値：	SYSOUT/SYSLST
デフォルト値：	SYSLST

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、Natural によって生成された出力を SYSOUT または SYSLST に書き込むかどうかを制御します。

BS2000/OSD ジョブ変数

Natural バッチモードドライバは、BS2000/OSD 機能の「ジョブ変数」を使用してリターンコードをユーザーまたは後続のジョブ（ステップ）に渡します。リターンコードは、Natural 自体（1～31の範囲）によって、または TERMINATE ステートメントをコンディションコードオプションとともに使用している場合は Natural アプリケーション（32～256の範囲）によって作成されます。

リターンコードを含めるジョブ変数は、リンク名 *NATB2JV を使用して宣言する必要があります。ジョブ変数のサポートは、Natural BS2000/OSD バッチモードドライバ NATBS2 内の SET パラメータ &JV の設定に依存します。

例：

```
/LOGON  
/DCLJV NATBJV, LINK=*NATB2JV  
/EXEC NATnnnB  
*TERMCC  
/LOGOFF
```

リターンコード 36 を NATBJV に割り当てるには、Natural プログラム TERMCC を以下のように指定します。

```
ASSIGN CC(N8) = 36  
TERMINATE CC  
END
```


42 バッチモードの Natural (すべての環境)

■ Adabas データセット	274
■ ソートデータセット	274
■ バッチモード環境のサブタスクセッションサポート	274

このドキュメントでは、Natural をバッチモードで実行する場合の全般的な考慮事項について説明します。

以下のトピックについて説明します。

Adabas データセット

Adabas データセットは、シングルユーザーモードでのみ指定する必要があります。これらのデータセットは、Adabas を使用する通常のアプリケーションプログラムの実行に必要なものと同じです。Adabas データセットの詳細については、関連する Adabas ドキュメントを参照してください。

ソートデータセット

SORT ステートメントを含む Natural プログラムが Natural セッション中に実行される場合は、ソートデータセットを指定する必要があります。

要件は、オペレーティングシステムのソートプログラムを呼び出す通常の COBOL または PL/1 アプリケーションプログラムの実行要件と同じで、使用中のソートプログラムによって異なります。

Natural は、中間データセットの SORTIN および SORTOUT を必要としませんが、ユーザー出口ルーチンインターフェイスの E15 および E35 を経由してソートプログラムと通信します。

バッチモード環境のサブタスクセッションサポート

- [目的](#)
- [前提条件](#)
- [機能](#)
- [Natural セッションの開始](#)
- [サブタスクの開始](#)
- [ユーザーパラメータエリアへのアクセス](#)



注意: CMS 対応 Natural では、サブタスクはサポートされていません。

目的

サブタスクサポートにより、1つのアドレススペース内で複数の Natural バッチモードセッションを実行できます。後続のジョブステップを実行するのではなく、1つのアドレススペース内で並行処理できるので、スループットが飛躍的に向上します。

通常、クライアント/サーバーアプリケーションおよび製品でこの機能 (Natural リモートプロシージャコールなど) を活用します。複数のサーバーサブタスクを開始して、リモートクライアントと通信することができます。

前提条件

Natural ニュークリアスを再起動する場合は、Natural ニュークリアスがリエントラントモジュールとしてリンクされている (リンケージエディタオプション RENT) 必要があります。

Adabas リンクルーチン (ADALNK) がリエントランシーサポートによって生成される必要があります。

機能

サブタスクを開始するには、Natural プログラムから CALL ステートメントを発行します。拡張フロントエンドパラメータリストによって、新しい Natural セッション (「サブタスク」) が開始されます。このリストには、以下に示す最大3つのパラメータセットが含まれます。

- ダイナミック Natural プロファイルパラメータ
- 起動パラメータ
- ユーザーパラメータ

標準 I/O データセットの変数名 (CMPRINT など) とバッチモードインターフェイス起動用の他のパラメータを、起動パラメータエリアの起動プログラムから渡すことができます。標準 I/O データセットは、未定義またはダミーのデータセットにすることができます。また、1つのセッションで所有することも、複数のセッションで共有することもできます。

さらに、Natural プログラムでユーザーパラメータエリアを読み取るために、CALL インターフェイスが用意されています。

Natural セッションの開始

- 拡張パラメータリスト
- 起動パラメータエリア
- ユーザーパラメータエリア

拡張パラメータリスト

拡張パラメータリストがない Natural バッチモードインターフェイスでは、標準のリンケージコールを使用して、最初の制御をオペレーティングシステムから取得します。レジスタ 1 は、最後のアドレスインジケータとして、高位ビットでアドレスをポイントします。このアドレスは、以下のパラメータエリアの長さを含むハーフワードフィールドをポイントします。

拡張パラメータリストには、最大 3 つのパラメータアドレスが含まれます。これは、最後のアドレス（最初、2 番目、または 3 番目のアドレス）の高位ビットで示されます。すべてのパラメータアドレスは、以下のパラメータエリアのパラメータ長を含むハーフワードフィールドをポイントします。長さ 0 は、パラメータエリアが存在しないことを示します。

- 最初のパラメータエリアは、Natural セッションのダイナミックプロファイルパラメータを含みます。
- 2 番目のパラメータエリアは、バッチモードインターフェイスを初期化するための特殊な起動パラメータを含みます。
- 3 番目のパラメータエリアは、Natural セッション中にアクセスできるユーザーパラメータエリアを含みます。

起動パラメータエリア

同じリージョン内で複数のバッチモード Natural (サブ) タスクが実行されている場合は、デフォルトではこれらのセッションはまったく同じ Natural 標準 I/O データセット (CMPRINT や CMSYNIN など) にアクセスします。これは、これらのファイル名の設定に使用できる Natural プロファイルパラメータがないためです。また、デフォルトでは Natural システム変数の *INIT-ID と *INIT-USER は、バッチモード用の定義なので同じです。

Natural サブタスクセッション用に一意の標準 I/O データセット名と一意の ID を指定するために、拡張パラメータリスト内の起動パラメータを使用して Natural システムのデフォルトを上書きできます。起動パラメータエリアは、ペアの 8 文字フィールドから成るテーブルです。

- 最初のエント리는、置換対象の 8 バイトのキーワードを含みます。
- 2 番目のエント리는、8 バイトの置換値を含みます。

キーワードと置換値は、必要に応じて末尾を空白で埋める必要があります。

有効なキーワードは以下のとおりです。

CMHCOPY	恒久的なハードコピーの出力先
CMSYNIN	コマンド入力データセット名
CMOBJIN	オブジェクト入力データセット名
CMPRINT	標準出力データセット名
CMPRMIN	ダイナミックパラメータ入力データセット名
CMPLOG	ダイナミックパラメータ出力データセット名
CMTRACE	トレース出力データセット名
INITID	ジョブステップ名 (システム変数 *INIT-ID)
MSGCLASS	CMPRINT および CMTRACE のダイナミックアロケーションのプールクラス (z/OS のみ)
NATRJE	ジョブサブミットデータセット名 (z/OS のみ)
STEPLIB	プログラムロードライブラリ名 (プロファイルパラメータ LIBNAM (ロードライブラリの名前) も参照。z/OS のみ)
SUBPOOL	z/OS ストレージサブプール (0~127、右詰め)
USERID	最初のユーザー ID (システム変数 *INIT-USER)

これらのエントリの使用は任意で、特定の指定順序はありません。データセットの空白値は、このデータセットが使用できないか空であることを表します。

z/VSE に関する注意事項：

デフォルトでは、すべての出力 (CMPRINT、CMHCOPY、CMTRACE、および CMPLOG の結果) は SYSLSLST に送信されます。SYS で始まるこれらのファイルの上書き指定は、z/VSE システム番号の上書きと見なされます。可能なフォーマットは SYSnnn です。nnn は 000~099 の範囲の 3桁の番号で、無効な番号 nnn を指定すると無視されます。

ユーザーパラメータエリア

ユーザーパラメータエリアのフォーマットはフリーです。ユーザーパラメータエリアは、特殊な CALL インターフェイスによって任意の Natural プログラムからアクセスできます (「[ユーザーパラメータエリアへのアクセス](#)」を参照)。

サブタスクの開始

Natural プログラムで以下のコールインターフェイスを使用して、同じアドレススペースでサブタスクを開始できます。

PGMNAME	制御を取得する Natural ニュークリアス名 (必須)。同じニュークリアスで再起動するには、先頭の文字にアスタリスクを指定します。実際のニュークリアス名がこのフィールドに返されます。
NATPARML	Natural ダイナミックパラメータエリア
STRPARML	起動パラメータエリア
USRPARML	ユーザーパラメータエリア

すべてのパラメータエリアは、以下のパラメータの長さで開始する必要があります。以下に、CMTASK の使用例を示します。

例：

```
DEFINE DATA LOCAL
01 PGMNAME (A8) INIT <'*'>
01 PARM1
02 NATPARML (I2) INIT <30>
02 NATPARMS (A30) INIT <'INTENS=1,IM=D,STACK=MYPROG'>
01 PARM2
02 STRPARML (I2) INIT <32>
02 STRPARAM1 (A16) INIT <'CMPRINT SYSPRINT'>
02 STRPARAM2 (A16) INIT <'CMPRMIN MYPARMS'>
01 PARM3
02 USRPARML (I2) INIT <80>
02 USRPARMS (A80) INIT <'special user parameters'>
END-DEFINE
CALL 'CMTASK' PGMNAME NATPARML STRPARML USRPARML
END
```

サンプルプログラム ASYNBAT は SYSEXTP ライブラリ内にあります。

ユーザーパラメータエリアへのアクセス

起動時に渡されたユーザーパラメータエリアは、以下の CALL ステートメントを使用して、任意の Natural プログラムから読み取ることができます。

```
CALL 'CMUPARM' USRPARML USRPARMS
```

USRPARML は、USRPARMS エリアの長さ (I2) (呼び出し前) と、返されたデータの長さ (呼び出し後) です。USRPARMS はパラメータデータエリアです。

返されるデータの長さがエリアの長さを超える場合は、データはエリアの長さに合わせて切り捨てられます。以下のリターンコードが返されます。

0	データは正常に移動されました。
4	データは移動されましたが切り捨てられました。
8	データを使用できません。
12	長さの値が正の値ではありません。
16	パラメータ数が不足しています。

サンプルプログラム GETUPARM は SYSEXTP ライブラリ内にあります。

43 Natural バッファプール

このドキュメントでは、Natural 管理者が z/OS、z/VSE、および BS2000/OSD オペレーティングシステム環境で利用できるさまざまなストレージ管理機能について説明します。

- **Natural** バッファプール - 全般
- z/OS 環境下の **Natural** グローバルバッファプール
- z/VSE 環境下の **Natural** グローバルバッファプール
- z/OS および z/VSE 環境下の共通 **Natural GBP** オペレーティング機能
- BS2000/OSD 環境下の **Natural** グローバルバッファプール

Natural バッファプールの機能の概要については、『*Natural* システムアーキテクチャ』ドキュメントの「*Natural* バッファプール」を参照してください。

Natural バッファプールに影響する Natural プロファイルパラメータの概要については、「*プロファイルパラメータの概要*」の「*バッファプール*」を参照してください。

44 Natural バッファプール - 全般

- Natural バッファプールの基本動作 284
- バッファプールのモニタとメンテナンス 290
- Natural グローバルバッファプール 293

バッファプールとは、Natural プログラムを実行に備えて配置するストレージエリアのことです。Natural ユーザーからの Natural オブジェクト要求に応じて、プログラムはバッファプールにまたはバッファプールから移動されます。概念上、プログラムをリエントラントエリアにまたはリエントラントエリアからロードする点で、ロードバッファプールはオペレーティングシステムと類似の機能を提供します。Natural バッファプールは、サポートされているすべての環境で Natural に必要不可欠です。

Natural バッファプールの基本動作

Natural は、リエントラントな Natural オブジェクトコードを生成します。コンパイルされたプログラムはバッファプールにロードされ、バッファプールから実行されます。このため、Natural プログラムの 1 つのコピーを複数のユーザーが同時に実行できます。

このセクションでは、以下のトピックについて説明します。

- バッファプール内のオブジェクト
- ディレクトリエントリ
- テキストプール
- バッファプールのハッシュテーブル
- バッファプールの初期化
- バッファプールの検索メソッド
- ローカルバッファプールとグローバルバッファプール
- バッファプールキャッシュ

バッファプール内のオブジェクト

バッファプール内のオブジェクトは、プログラム、サブプログラム、マップ、およびグローバルデータエリアです。グローバルデータエリアは、コンパイル目的でのみバッファプールに配置されます。この場合は、同じ名前の 2 つのオブジェクト、つまりグローバルデータエリア自体および対応するシンボルテーブルがバッファプールにロードされます。

ディレクトリエントリ

Natural オブジェクトがバッファプールにロードされると、ディレクトリエントリと呼ばれるコントロールブロックがこのオブジェクトに割り当てられます。

ディレクトリエントリには、オブジェクトの名前、オブジェクトが属するライブラリ、オブジェクトの取得元のデータベース ID および Natural システムファイル番号などの情報と、いくつかの統計情報（特定の時間にプログラムを同時に実行しているユーザーの数など）が含まれます。

ユーザーがプログラムを実行する場合は、Natural はディレクトリエントリを調べて、プログラムがすでにバッファプールにロードされているかどうかを確認します。まだロードされていない場合は、プログラムのコピーが適切な Natural システムファイルから取得され、バッファプールにロードされます。

オブジェクトをバッファプールにロードするときに、新しくロードするオブジェクトを収容するために、現在実行中ではない1つまたは複数の他のNaturalオブジェクトがバッファプールから削除されることがあります。新しいオブジェクトがロードされると、このオブジェクトを識別するために新しいディレクトリエントリが作成されます。

オブジェクトがシステムファイルから削除されると、そのオブジェクトは使用されなくなるとすぐにバッファプールからも削除されます。オブジェクトが新しくカタログ化または格納されると、そのオブジェクトの古いバージョンは使用されなくなるとすぐにバッファプールからも削除されます。再度実行を要求された場合は、新しいバージョンがシステムファイルからバッファプールにロードされます。

テキストプール

バッファプールにロードされるプログラムの実際のオブジェクトコードは、テキストプールと呼ばれるエリアに配置され、このテキストプール内の連続したメモリ領域が割り当てられます。このテキストプールは、いくつかの4KBのバッファに分割されます。このサイズは任意で、Natural管理者が自由に変更できます。オブジェクトがロードされると、オブジェクトのオブジェクトコードを保存するために、相互に連続する1つ以上のテキストバッファが割り当てられます。

バッファプールのハッシュテーブル

このセクションは、`TYPE=NAT` のバッファプールにのみ適用されます。

ハッシュテーブルは、バッファプールディレクトリ内のオブジェクトを見つける検索時間を短縮するために使用されます。ハッシュテーブル内のエントリ数は、ディレクトリエントリ数を2倍して次の素数に切り上げた数になります。これにより、任意の時点でテーブルの半分のみが入力され、競合の確率がほとんどなくなります。その結果、ハッシュテーブル内の既存のオブジェクトを見つけるための平均テスト回数が理論的に2未満になります。

ハッシュの条件は8文字長のプログラム名です。複数のプログラム名がハッシュテーブル内の同じ場所にハッシュされる場合は、オーバーフロー技術によって競合が解決されます。

ハッシュテーブルに必要なストレージは、テキストブロック当たり約16バイトです。このため、テキストプール内の使用可能ストレージが1.6% (1KBテキストブロック) から0.1% (16KBテキストブロック) まで削減されます。

バッファプールの初期化

グローバルバッファプールの場合は、グローバルバッファプールの起動時に初期化が行われます。

ローカルバッファプールの場合は、初期化のタイミングは環境によって異なります。

- バッチモードの場合、TSO、TIAM、およびVM/CMS環境では、Naturalセッションを開始したときに初期化が行われます。
- TP モニタ環境では、通常、最初のユーザーがこの TP モニタ環境で Natural を起動したときに初期化が行われます。Com-plete および CICS 環境では、TP モニタの起動時にローカルバッファプールを初期化することも可能です。

バッファプールの検索メソッド

前述および後述のとおり、バッファプール内でスペースを割り当てるためのさまざまなメソッドがあります。

▶ **手順 44.1. 検索メソッドを選択するには、以下のパラメータを使用します。**

- Natural プロファイルパラメータ BPMETH および BPI
または、Natural パラメータモジュール内の NTBPI マクロ
または、グローバルバッファプールの機能パラメータ METHOD

これらのパラメータと NTBPI マクロの詳細については、Natural の『パラメータリファレンス』ドキュメントを参照してください。

以下に、検索メソッドに関する情報を示します。

- **METHOD=S**
- **METHOD=N**
- **検索メソッドの選択**

METHOD=S

METHOD=S は、新しいロードの実行に必要なスペースを取得するために、バッファプール内のストレージを割り当てるための検索アルゴリズムとして選択プロセスが使用されることを示します。

使用される選択プロセスは、検索アルゴリズム 1 とアルゴリズム 2 の組み合わせです。

■ アルゴリズム 1

検索アルゴリズム1では、バッファプール内で、フリースペースであるストレージ、または未使用オブジェクト（アクティブであるが使用されていないオブジェクト）によって占有されているスペースであるストレージを見つけようとしています。

必要なオブジェクトサイズとまったく同じサイズのフリースペースが見つかった場合は、選択プロセスはすぐに終了します。それ以外の場合は、検索を続行し、最適なサイズのエントリを見つけるためにバッファプールの先頭から最後まで参照してバッファプール内のエントリを比較します。また、未使用のオブジェクトの場合は、検索の際に、オブジェクトが最後にアタッチされた時間、つまりオブジェクトがロードまたは検索時に最後に参照された時間も考慮されます。

選択プロセスの終了時には、要求されたサイズ以上のフリースペースまたは未使用オブジェクトのスペースが選択されています。検索に適用される優先規則では、最初にフリースペースが選択され、次に未使用オブジェクトのスペースが選択されます。未使用オブジェクトの場合は、最も古いオブジェクトが最初に削除されます。

アルゴリズム1の選択プロセスが成功しなかった場合は、アルゴリズム2が開始されます。

■ アルゴリズム 2

選択アルゴリズム2は、アルゴリズム1が失敗した場合に開始されます。アルゴリズム2では、アルゴリズム1から渡されたバッファプール内の位置から検索を開始し、新しいロードに必要なストレージを取得するために、複数のエンティティ（フリーストレージまたは未使用のオブジェクトが占有しているスペース）を組み合わせようとしています。ただし、オブジェクトの古さは考慮されません。

アルゴリズム2ではテキストレコードセクションの最後まで処理を続行し、必要に応じて、テキストレコードセクションの先頭までラップアラウンドして先頭から最後まで最終パスを作成します。それでもスペースが使用できない場合は、アルゴリズム2は失敗し、オブジェクトはロードされず、Naturalは該当するエラーメッセージを表示します。

METHOD=N

METHOD=Nは、新しいロードの実行に必要なスペースを取得するために、次に使用可能なフリーまたは未使用スペースが使用されることを示します。未使用スペースとは、アクティブであるが使用されていないオブジェクトによって占有されているスペースのことです。

次に使用可能なスペースの検索は、ラップアラウンド方式でバッファプールを移動するポインタから開始されます。次に使用可能な任意のフリーまたは未使用オブジェクトを含むバッファプールエンティティが使用され、場合によっては要求されたストレージ容量を取得するために連結されます。

割り当て要求中にバッファプールの最後に達した場合は、ポインタはバッファプールの先頭にラップアラウンドされ、バッファプールの先頭から最後まで最終検索が1回実行されます。再度バッファプールの最後に達してもオブジェクトをロードできない場合は、ロードは失敗し、Naturalは該当するエラーメッセージを表示します。

METHOD=N は、特に、バッファプールの容量が大きい場合は**バッファプールキャッシュ**機能と組み合わせることを考慮してください。詳細については、以下の「検索メソッドの選択」も参照してください。

検索メソッドの選択

デフォルトではMETHOD=Sが使用されます。このメソッドの長所は、スペースを割り当てる際に入念な検索が行われることです。この検索では、オブジェクトのサイズおよび古さが考慮され、新しいロード用のスペースを確保するために、確実に最も必要のない未使用オブジェクトがバッファプールから削除されます。

METHOD=Sの短所は、バッファプールの先頭から最後まで参照する際に、選択プロセスによって大量のCPU時間が消費される可能性があることです。

METHOD=Nの長所は、選択プロセスが短く、通常、参照が少ないのでスペースの割り当てに必要なCPU時間が短くなることです。これは、バッファプールの容量が大きい場合に重要です。

METHOD=Nの短所は、バッファプールから削除するオブジェクトの選択がそれほど精密に行われないことです。削除されたオブジェクトの再ロードのためにAdabas I/Oが増大するのを防ぐために、METHOD=Nと**バッファプールキャッシュ**機能を組み合わせて使用することをお勧めします。

ローカルバッファプールとグローバルバッファプール

ローカルバッファプール

インストールテープで提供されたNaturalを使用している場合は、ローカルバッファプールを実行しています。これは、使用中の環境の同じパーティション（またはリージョンかアドレススペース）に割り当てられるバッファプールエリアです。

例えば、バッチモードの場合、TSOまたはCMS環境では、各ユーザー用にローカルバッファプールがあります。Com-plete、CICS、IMS/TMTPなどのTPモニタ環境では、TPモニタ当たり1つのバッファプールがあり、すべてのTPユーザーはここから実行します。

グローバルバッファプール

z/OS環境では、CSAまたはECSAストレージからグローバルバッファプールが割り当てられます。このような環境では、すべてのTSOユーザー、バッチユーザー、およびTPモニタユーザーは1つの共通グローバルエリアから実行できます。

z/VSE環境では、グローバルバッファプールはシステムGETVISエリア（上または下）から割り当てられます。このような環境では、すべてのバッチユーザーおよびTPモニタユーザーは1つの共通グローバルエリアから実行できます。

VM/CMS環境では、グローバルバッファプールを書き込み可能な非連続共有セグメントとしてインストールできます。このセグメントは、Natural/CMSの起動時にユーザーの仮想マシンに

ダイナミックにアタッチされます。Naturalの『インストール』ドキュメントの「VM/CMS 環境でのNaturalのインストール」および「Natural用VMシステムの準備」も参照してください。

BS2000/OSD環境では、グローバルバッファプールは共通メモリプールです。「[BS2000/OSD環境下のNaturalグローバルバッファプール](#)」を参照してください。

グローバルバッファプールの詳細については、「[Naturalグローバルバッファプール](#)」を参照してください。

バッファプールキャッシュ

このセクションは、`TYPE=NAT`のグローバルバッファプールおよび`TYPE=NAT`または`TYPE=SWAP`のローカルバッファプールに適用されます。

バッファプールキャッシュは、グローバルバッファプールおよびローカルバッファプールと組み合わせて使用することができます。z/VMでは使用できません。バッファプールキャッシュはNaturalプログラミングオブジェクト（プログラム、サブプログラム、マップなど）にのみ使用され、Natural for DL/Iで生成されたオブジェクトなどには使用されません。

バッファプールが複数ユーザーから要求されたすべてのオブジェクトを収容できるほど大きくない場合は、特別な過負荷対策方法によって、要求されたオブジェクトで既存のオブジェクトを置き換えます。過負荷状況の発生回数はバッファプールの効率に直接影響します。望ましくない過負荷を減らしてバッファプールのパフォーマンスを向上させる最適で最も効果的な方法は、単にバッファプールのサイズを増やすことです。

ただし、この方法はほとんどのお客様のサイトで使用できません。プライマリアドレススペース、z/OS (E)CSA、z/VSEシステムGETVISエリア、またはBS2000/OSD共通メモリプールに使用可能なストレージが不足しているからです。

上記の状況を改善するために、バッファプールキャッシュが使用されます。このメカニズムの主な目的は、「バッファプールストレージ不足」のためにバッファプールから削除されるすべてのオブジェクトが失われないようにすることです。つまり、オブジェクトの削除は、「バッファプールキャッシュへのスワップアウト」になります。この機能が意図する利点は、オブジェクトのロードに使用されるデータベースコールが削減され、その結果パフォーマンスが向上することです。

グローバルバッファプールの注意事項：

バッファプールキャッシュエリアは、データスペースに割り当てられます。バッファプール用にデータスペースが作成されると（z/OSまたはz/VSEで指定されるプロファイルパラメータBPCSIZE、またはBS2000/OSDで指定されるDATAパラメータ）、作成者タスクに所有権が割り当てられます。このタスクが終了すると、データスペースはシステムによって自動的に削除されます。この場合は、RESIDENT=Yが設定されているかどうかに関係なく、作成者タスクはまだアクティブです。

ローカルバッファプールの注意事項：（z/OSおよびz/VSEのみ。CompleteおよびIMS/TMは除く）

バッファプールキャッシュは、データスペースまたは "バーの上" のメモリオブジェクトつまり 64 ビットメモリ (z/OS のみ) に割り当てられます。バッファプール用にデータスペースまたはメモリオブジェクトが作成されると (プロファイルパラメータ BPCSIZE および BPC64 を参照)、作成者タスクに所有権が割り当てられます。このタスクが終了すると、データスペースまたはメモリオブジェクトはシステムによって自動的に削除されます。

バッファプールのモニタとメンテナンス

Natural ユーティリティ SYSBPM (Natural の『ユーティリティ』ドキュメントを参照) は、バッファプールの現在のステータスに関する統計情報を提供します。SYSBPM を使用すると、バッファプールを必要に応じて調整することもできます。

以下では次のトピックについて説明します。

- [プリロードリスト](#)
- [ブラックリスト](#)
- [バッファプールの変更の反映](#)
- [パフォーマンスの考慮事項](#)

プリロードリスト

プリロードリストは、バッファプールにロードされてそのまま常駐するオブジェクトのリストです。このようなオブジェクトは、ユーザーが実行するために要求したときにはすでにバッファプール内に存在するので、システムファイルからロードする必要がありません。

これにより、パフォーマンスが向上し、バッファプールのフラグメントを防ぎ、システムファイルを含むデータベースがアクティブでない場合でも重要なエラーメッセージを常に使用可能状態にしておくことができます。

Natural は、Natural セッションの開始時にプリロードリストをチェックして、プリロードリスト内のオブジェクトですでにバッファプールに存在するものを確認します。存在しないオブジェクトは、システムファイルからバッファプールにロードされます。このチェックとロードは、バッファプールの接続時、再接続時、および初期化時に SYSBPM ユーティリティを使用して実行することもできます。REFRESH コマンドを使用してグローバルバッファプールを再初期化する場合は、既存の Natural セッションはチェックされません。つまり、このバッファプールにアクセスする新しい Natural セッションが開始されない限り、プリロードリスト内のオブジェクトはロードされません。

プリロードリストのロードはシリアルライズされません。このため、複数の Natural セッションが同時に開始された場合は、各セッションはプリロードリストに指定されているすべてのオブジェクトをバッファプールにロードしようとします。これにより、同じ Natural オブジェクトの複数のエントリがバッファプールに複製されることがあります (下記のヒントも参照)。

プリロードリストは名前で識別され、その名前を起動パラメータとして指定（グローバルバッファプールの場合）するか NTBPI マクロで指定（ローカルバッファプールの場合）することにより、特定のバッファプールにアタッチされます。このため、バッファプールごとに異なるプリロードリストを定義することも、異なるバッファプールに同じプリロードリストを使用することもできます。

指定したプリロードリストが見つからない場合、またはプリロードリストに含まれているオブジェクトをロードできない場合は、セッションの初期化時に該当する警告メッセージが表示されます。いずれの場合も、後続する各セッションの初期化ごとにプリロードが繰り返されます。

プリロードリスト内のオブジェクトは、最初にロードされるので、バッファプールの先頭に配置されます。ただし、最初のプリロードですべてのオブジェクトをロードできなかった場合を除きます。この場合は、オブジェクトはバッファプール内のいずれかの位置に配置されます。

プリロードリストの管理には、SYSBPM ユーティリティを使用します。Natural の『ユーティリティ』ドキュメントの「SYSBPM - プリロードリストの管理」を参照してください。

 **ヒント:** ユーザーセッションによるプリロードリスト内のオブジェクトのロードに関連する問題を防ぐには、次の手順に従います。

- グローバルバッファプールの場合：
グローバルバッファプールの割り当てまたは更新の直後に、グローバルバッファプールにアクセスして FIN を実行するバッチ Natural セッションを開始します。
- CICS および Com-plete 環境下のローカルバッファプールの場合：
TP システムの起動時に、ローカルバッファプールにアクセスする非同期 Natural セッションを開始し、Natural スタックに FIN コマンドを挿入します。この Natural セッションが NTBPI マクロでプリロードリストの名前を参照していることを確認してください。

ブラックリスト

Natural オブジェクトが実行されないようにするには、目的のオブジェクトを「ブラックリスト」に配置します。配置したオブジェクトは、バッファプールにロードできなくなり、すでにロードされている場合は実行できなくなります。このオブジェクトの実行をユーザーが要求すると、該当するエラーメッセージが表示されます。

ブラックリストには、個々のオブジェクトだけでなく、ライブラリ全体を配置することもできます。

例：

- Natural アプリケーションをアップグレードする際に、アップグレードを完了するまでユーザーがアプリケーションを使用できないようにする場合などに、ブラックリストが役立ちます。ブラックリストを使用しないと、ユーザーが新しいモジュールを実行したときに古いモジュールが呼び出されて、Natural セッションが異常終了するおそれがあります。ブラックリストを使用すると、要求したオブジェクトが現在実行不可であることを示すメッセージが表示された後、ユーザーは Natural セッションを継続できます。

- パフォーマンスの面でも、ブラックリストを使用すると便利です。特定のリソース消費オブジェクトが特定の環境では実行されないようにすることができます。
- ブラックリストを使用して、テストプログラムが実稼働環境で実行されないようにすることができます。

ブラックリストの管理には、SYSBPM ユーティリティを使用します。Natural の『ユーティリティ』ドキュメントの「SYSBPM - ブラックリストの管理」を参照してください。

バッファプールの変更の反映

 **注意:** z/OS 環境では、バッファプールの変更の反映は常に、変更が発生した Natural サブシステム内に限られます (Natural サブシステムの詳細については、「[Natural サブシステム](#)」 (z/OS) または「[Natural サブシステム](#)」 (z/VSE) を参照)。したがって、この場合の "すべてのグローバルバッファプール" は、"同じサブシステム内のすべてのグローバルバッファプール" を意味します。

環境によっては、バッファプールと使用中の Natural アプリケーションの整合性を維持するために、1つのバッファプール (ローカルまたはグローバル) で発生した変更が他のすべてのグローバルバッファプールにも反映される、つまり、同じ変更が他のグローバルバッファプールでも自動的に行われるということが重要になります。これは、Sysplex 環境で特に重要です。

例えば、Natural プログラムが 1つの z/OS イメージで新しくカタログ化された場合、そのプログラムは、反映によって Sysplex 内の他のすべてのグローバルバッファプールから削除され、再実行されると新しいバージョンがシステムファイルからロードされます。

反映される変更は以下のとおりです。

- バッファプールからのオブジェクトの削除
- バッファプールのブラックリストの変更
- バッファプールの再初期化

変更は、現在の z/OS イメージ内または Sysplex 全体の他のすべてのグローバルバッファプール、または Sysplex 内の同じ名前他のすべてのグローバルバッファプールに反映できます。

反映は、別のグローバルバッファプールのバッファプール内常駐として定義されているオブジェクトには影響しません。

反映は、Natural プロファイルパラメータ BPPROP によってアクティブ化され範囲が制御されません。

 **注意:** 反映は非同期で行われ、オブジェクトは使用されなくなった場合のみバッファプールから削除されるので、オブジェクトの現在のバージョンがすべてのバッファプールで使用可能になるまで多少時間がかかります。

他のローカルバッファプールに反映することはできません。

パフォーマンスの考慮事項

バッファプールとバッファプールキャッシュのパフォーマンスに関する全般的なヒントについては、Natural の『ユーティリティ』ドキュメントの SYSBPM セクションで「パフォーマンスの考慮事項」を参照してください。

Natural グローバルバッファプール

Natural グローバルバッファプールは、オプションの Natural コンポーネントです。

これは、以下のオペレーティングシステムで使用できます。

- z/OS（「[z/OS 環境下の Natural グローバルバッファプール](#)」を参照）
- z/VSE（「[z/VSE 環境下の Natural グローバルバッファプール](#)」を参照）
- BS2000/OSD（「[BS2000/OSD 環境下の Natural グローバルバッファプール](#)」を参照）

使用されるプロファイルパラメータ

以下の Natural プロファイルパラメータを使用して、グローバルバッファプールを設定します。

BPNAME	グローバルバッファプールの名前を指定します（BPNAME を参照）。BPNAME=' '（空白）は、ローカルバッファプールへの接続を確立するために使用します。
SUBSID	使用する Natural サブシステムの ID を指定します（プロファイルパラメータ SUBSID を参照。z/OS および z/VSE 環境にのみ適用）。

Natural は、起動時にこれらのパラメータを使用してグローバルバッファプールを見つけようとします。

バッファプールのオープン／クローズ手順

Natural パラメータモジュールの NTBPI マクロまたは対応するプロファイルパラメータ BPI を使用して、複数のバッファプールを定義できます。

セッションの初期化時に、Natural は定義された最初のバッファプールへの接続を確立しようとします。これに失敗すると、定義された 2 番目のバッファプールへの接続を確立しようとします。これにも失敗すると、次に定義されているバッファプールへと順に試していきます。バッファプールへの接続の試みが失敗するたびに、該当するメッセージが表示されます。

バッファプールを停止する場合も同じ手順が適用されます。Natural セッションがまだアクティブなときに、現在接続しているバッファプールを閉じると、Natural は定義されている順番に従って別のバッファプールに接続し、セッションを継続しようとします。このため、Natural 管理者は、アクティブな Natural セッションをすべて終了しなくても、グローバルバッファプールを閉じることができます。

45 z/OS 環境下の Natural グローバルバッファプール

- Natural グローバルバッファプールの使用 296
- Natural グローバルバッファプールのオペレーティング 296
- NATGBPvr 実行ジョブの例 298
- ローカリゼーション 300

このドキュメントでは、z/OS オペレーティングシステム環境での Natural グローバルバッファプール（GBP）の目的と使用方法について説明します。

以下のトピックについて説明します。

Natural グローバルバッファプールの特定の部分は、z/OS および z/VSE 環境で共通しています。これらの部分については、以下のトピックを含む別のセクションで説明しています（「[z/OS および z/VSE 環境下の共通 Natural GBP オペレーティング機能](#)」を参照）。

- [グローバルバッファプールのオペレーティング機能](#)
- [グローバルバッファプールの機能パラメータ](#)
- [NATBUFFER 指定の例](#)

以下の項目も参照してください。

- Natural の『メッセージおよびコード』ドキュメントの「[Natural グローバルバッファプール マネージャのメッセージ](#)」

Natural グローバルバッファプールの使用

定義

Natural グローバルバッファプールは、16 MB より上の z/OS 拡張共通システム記憶域（ECSA）（または要求に応じて 16 MB より下の CSA ストレージ）から割り当てられるストレージのセグメントであり、Natural が Natural プログラムをロードおよび実行する際に使用されます。

利点

グローバルバッファプールを使用すると、異なる TP モニタ（CICS、TSO、IMS/TM などの複数のコピー）環境や複数のバッチセッションの複数の Natural セッションで同じエリアを共有するため、環境ごとにローカルバッファプールを必要とする場合よりも必要なストレージが少なくて済みます。

Natural グローバルバッファプールのオペレーティング

以下では次のトピックについて説明します。

- [Natural GBP オペレーティングプログラムのインストール](#)
- [Natural グローバルバッファプールの設定](#)
- [Natural GBP オペレーティングプログラムの起動](#)

■ Natural GBP オペレーティングプログラムの停止

Natural GBP オペレーティングプログラムのインストール



注意: *vrs* は、製品のバージョン、リリース、およびシステムメンテナンスレベルを表します。

グローバルバッファプールは、認可ライブラリから実行する必要がある NATGBP*vr* プログラムによって操作されます。

Natural のインストール時に、NATGBP*vr* モジュールが APF 認可ライブラリにリンクされます。

z/OS パラメータ ALLOWUSERKEYCSA(YES) が SYS1.PARMLIB(DIAGxx) で明示的に指定されている場合、Natural グローバルバッファプールはユーザーキー内で割り当てられます。したがって、グローバルバッファプールにアクセスする Natural セッションには、そのバッファプールに対する書き込み権限があります。

ALLOWUSERKEYCSA(NO) が有効な場合、Natural グローバルバッファプールはシステムキー内で割り当てられます。したがって、グローバルバッファプールにアクセスする Natural セッションには、そのバッファプールに対する書き込み権限はありません。これらの Natural セッションでは、システムキー内で書き込み権限が必要な機能を実行するために認可サービスマネージャ (ASM) が呼び出されます。そのため、[認可サービスマネージャ](#)のインストールは必須です。認可サービスマネージャの呼び出しによって、Natural のリソースの消費は増加します。

Natural グローバルバッファプールの設定

NATGBP*vr* から使用できる機能は、以下の場合にアクティブになります。

- パラメータカード (PARM=) によって提供
- ファイルから読み取り (下記参照)
- NATGBP*vr* 実行中に MODIFY オペレータコマンドによって提供

NATGBP*vr* は、EXEC ステートメントのパラメータフィールド (PARM=) に最初のコマンドがあると想定します。

以下を入力できます。

- いずれかの機能 (「[z/OS および z/VSE 環境下の共通 Natural GBP オペレーティング機能](#)」を参照)
- CF=<dd-name> を指定した入力ファイルへの参照。<dd-name> は JCL で定義されている DD 名です。サポートされているのは「カードイメージ」ファイル、つまり RECFM=F, LRECL=80 のみで、入力レコードの最初の 72 バイトのみが優先されます。入力ファイルから取得される各レコードはコマンドを表します。空白のレコードと接頭辞がアスタリスク (*) のレコードは無視されます。ファイルは、ファイルの終わり (EOF) に達するまで処理されます。例：
PARM='CF=SYSIN1'

パラメータフィールドが提供されていないか空白の場合は、コマンドはデフォルトでSYSINファイルから読み取られます。

機能の入力はコンソールで一度に1つずつまたはコマンドファイルを使用して1行に1つのみ可能で、それ以外の場合はエラーメッセージが返されます。

パラメータカード、ファイル入力、またはオペレータコンソール入力から取得された各コマンドは、オペレータコンソールに表示されます。

Natural GBP オペレーティングプログラムの起動

NATGBPvrプログラムを起動するには、開始タスクを開始するか、NATGBPvrを実行するジョブをサブミットします。

 **重要:** システム障害後にグローバルバッファプールを確実に維持するには、マシンの IPL 時にグローバルバッファプールが自動的に起動される必要があります。

Natural GBP オペレーティングプログラムの停止

すべてのコマンドが処理された後、以下の場合を除いて、NATGBPvr は終了します。

- RESIDENT=Y が指定されている
- キャッシュ付きのバッファプールが作成されている

NATGBPvr は、以下のいずれかのコンディションコードを返します。

0	すべての機能が正常に実行されました。
20	エラーが発生しました。詳細については、メッセージログを参照してください。

NATGBPvr 実行ジョブの例

以下に、グローバルバッファプールを作成および終了するバッチジョブの例を示します。

以下の例で、vrs または vr は、対応するバージョン、リリース、およびシステムメンテナンスレベル番号を表します。製品バージョンの詳細については、『用語集』の「バージョン」を参照してください。

例 1 :

```
//GBPSTART JOB
//*
//* Starts a global buffer pool with the name NATvrGBP, a size of 1 MB and
//* a text block size of 4 KB. The global buffer pool is allocated above 16 MB.
//* The subsystem used is NATv.
//* After the allocation, the job GBPSTART terminates.
//*
//STEP EXEC PGM=NATGBPvr,PARM='BPN=NATvrGBP,N=(1M)'
//SETPLIB DD DISP=SHR,DSN=USER.APF.LINKLIB
```

例 2 :

```
//GBPRES JOB
//*
//* Starts a global buffer pool with the name GBP, a default size of
//* 100 KB and a text block size of 1 KB. The global buffer pool is allocated
//* below 16 MB. The subsystem used is SAGS.
//* After the allocation, the job GBPRES will wait for further commands.
//* Further commands may be entered using the MODIFY operator command:
//* F GBPRES,command-string
//*
//STEP EXEC PGM=NATGBPvr,PARM='BPN=GBP,N=(,BL,1),S=SAGS,R=Y'
```

例 3 :

```
//GBPSTOP
//*
//* Stops the global buffer pool GPB if it contains no active objects. If it
//* does contain active objects, the operator console will prompt for a reply.
//* Depending on the reply, the shutdown will be forced (Y) or aborted (N).
//* The subsystem used is NATv.
//*
//STEP EXEC PGM=NATGBPvr,PARM='FSHUT,BPN=GPB'
```

例 4 :

```
//GBPSTRT2
//* Read commands from SYSIN1:
//*
//* Start 3 global buffer pools (subsystem ID Nvrs) with name
//*   NATGBP1 - size=1024KB and a cache with size 2048KB
//*   NATGBP2 - size=2048KB without cache
//* Display all buffer pools of subsystem ID "Nvrs".
//*
//* Note: The job does not terminate by itself, but stays resident and waits
//*       for operator commands, because it owns the data space allocated for
//*       buffer pool NATGBP1.
```

```
/*  
/* If the buffer pools should shut down, send operator command MODIFY with  
/* parameter "CF=SYSIN2" to execute the corresponding FSHUTs.  
/*  
/*STEP EXEC PGM=NATGBPvr,PARM='CF=SYSIN1'  
/*SYSIN1 DD *  
CREATE,BPN=NATGBP1,S=Nvrs,N=(1M),BPC=2M  
CREATE,BPN=NATGBP2,S=Nvrs,N=(2M)  
SHOWBP S=Nvrs  
/*SYSIN2 DD *  
FSHUT,BPN=NATGBP1,S=Nvrs  
FSHUT,BPN=NATGBP2,S=Nvrs  
SHOWBP S=Nvrs  
/*
```

ローカリゼーション

モジュール NATGBPTX はソース形式で提供されます。このモジュールには、すべてのエラーメッセージが英語（大文字と小文字が混在）で含まれています。メッセージは、必要に応じて他の言語に翻訳できます。この場合は、「新しい」 NATGBPTX ソースモジュールをアセンブルし、NATGBPvr をリンクし直す必要があります。

大文字の変数部分を含むグローバルバッファプールメッセージを発行するには、UCTRAN パラメータを YES に設定してグローバルバッファプールパラメータモジュール NATGBPRM をアセンブルし、NATGBPvr をリンクし直す必要があります。

NATGBPvr をリンクし直すには、以下の JCL を使用します。

```
/*SYSLIN DD *  
SETCODE AC(1)  
SETOPT PARM(REUS=RENT)  
INCLUDE NATLIB(NATGBPMG)  
INCLUDE SMALIB(NATGBPRM)  
INCLUDE SMALIB(NATGBPTX)  
INCLUDE NATLIB(NATBPMGR)  
NAME NATGBPvr(R)  
/*
```

46 z/VSE 環境下の Natural グローバルバッファプール

■ Natural グローバルバッファプールの使用	302
■ Natural グローバルバッファプールのオペレーティング	303
■ NATGBPvr 実行ジョブの例	304
■ ローカリゼーション	306

このドキュメントでは、z/VSEオペレーティングシステム環境でのNatural グローバルバッファプール（GBP）の目的と使用方法について説明します。

以下のトピックについて説明します。

Natural グローバルバッファプールの特定の部分は、z/VSEおよびz/OS環境で共通しています。これらの部分については、以下のトピックを含む別のセクションで説明しています（「[z/OS および z/VSE 環境下の共通 Natural GBP オペレーティング機能](#)」を参照）。

- [グローバルバッファプールのオペレーティング機能](#)
- [グローバルバッファプールの機能パラメータ](#)
- [NATBUFFER 指定の例](#)

以下の項目も参照してください。

- Natural の『メッセージおよびコード』ドキュメントの「[Natural グローバルバッファプール マネージャのメッセージ](#)」

Natural グローバルバッファプールの使用

定義

Natural グローバルバッファプールは、16 MB より上の z/VSE システム GETVIS ストレージ（または、要求に応じて 16 MB より下のストレージ）から割り当てられるストレージのセグメントであり、Natural が Natural プログラムをロードおよび実行する際に使用されます。Natural グローバルバッファプールはストレージキー9に割り当てられるので、参加するすべてのパーティションはそこに書き込みアクセスできます。

利点

グローバルバッファプールを使用すると、異なる TP モニタ（CICS や Com-plete などの複数のコピー）環境や複数のバッチセッションの複数の Natural セッションで同じエリアを共有するため、環境ごとにローカルバッファプールを必要とする場合よりも必要なストレージが少なく済みます。

前提条件

z/VSE 環境の Natural グローバルバッファプールには、ESA/390 または互換プロセッサのサブシステムストレージ保護機能が必要です。このため、このハードウェア機能をサポートするには、最小オペレーティングシステムレベルの z/VSE バージョン 2 リリース 4 も必要となります。

Natural グローバルバッファプールのオペレーティング

Natural GBP オペレーティングプログラムのインストール

インストール手順は必要ありません。グローバルバッファプールは、NATGBP_{vr} プログラムによって操作されます。このプログラムは、Natural ロードライブラリに含まれ、そこから実行されます。

Natural グローバルバッファプールの設定

NATGBP_{vr} から使用できる機能は、以下の場合にアクティブになります。

- パラメータカード (PARM=) によって提供
- ファイルから読み取り (下記参照)
- NATGBP_{vr} 実行中にオペレータ (AR コマンド MSG_{xx}。xx は z/VSE パーティション ID) によって提供

NATGBP_{vr} は、EXEC ジョブ制御文のパラメータフィールド (PARM=) に最初のコマンドがあると想定します。

以下を入力できます。

- いずれかの機能 (「z/OS および z/VSE 環境下の共通 Natural GBP オペレーティング機能」を参照)
- CF=<dlbl-name> を指定した入力ファイルへの参照。<dlbl-name> は JCL で定義されている DLBL 名または z/VSE (パーティション) 標準ラベルです。サポートされているのは「カードイメージ」ファイル、つまり RECFM=F,LRECL=80 のみで、入力レコードの最初の 72 バイトのみが優先されます。入力ファイルから取得される各レコードはコマンドを表します。空白のレコードと接頭辞がアスタリスク ("*") のレコードは無視されます。<dlbl-name> のアスタリスク (*) は、入力を SYSIPT から読み取る必要があることを NATGBP_{vr} に示します。ファイルは、ファイルの終わり (EOF) に達するまで処理されます。

例: PARM='CF=SYSIN1'

パラメータフィールドが提供されていないか空白の場合は、コマンドはデフォルトで SYSIPT から読み取られます。

機能の入力はコンソールで一度に1つずつまたはコマンドファイルを使用して1行に1つのみ可能で、それ以外の場合はエラーメッセージが返されます。

パラメータカード、ファイル入力、またはオペレータコンソール入力から取得された各コマンドは、オペレータコンソールに表示され、SYSLST に記録されます。

Natural GBP オペレーティングプログラムの起動

NATGBPvr プログラムを起動するには、NATGBPvr を実行するジョブをサブミットします。

 **重要:** システム障害後にグローバルバッファプールを確実に維持するには、マシンの IPL 時にグローバルバッファプールが自動的に起動される必要があります。

Natural GBP オペレーティングプログラムの停止

すべてのコマンドが処理された後、以下の場合を除いて、NATGBPvr は終了します。

- RESIDENT=Y が指定されている
- キャッシュ付きのバッファプールが作成されている

NATGBPvr は、以下のいずれかのコンディションコードを返します。

0	すべての機能が正常に実行されました。
20	エラーが発生しました。詳細については、メッセージログを参照してください。

NATGBPvr 実行ジョブの例

以下に、グローバルバッファプールを作成および終了するバッチジョブの例を示します。

以下の例で、vrs または vr は関連するバージョン (version)、リリース (release)、システムメンテナンスレベル番号 (system maintenance level number) を表します。製品バージョンの詳細については、『用語集』の「バージョン」を参照してください。

例 1 :

```
// JOB GBPSTART
/*
/* Starts a global buffer pool with the name NATvrGBP, a size of 1 MB and
/* a text block size of 4 KB. The global buffer pool is allocated above 16 MB.
/* The subsystem used is NATv.
/* After the allocation, the job GBPSTART terminates.
/*
// LIBDEF PHASE,SEARCH=SAGLIB.NATLIB
// EXEC NATGBPvr,SIZE=NATGBPvr,PARM='BPN=NATvrGBP,N=(1000)'
/*
// EXEC LISTLOG
/;&
```

例 2 :

```
// JOB GBPRES
/*
/* Starts a global buffer pool with the name GBP, a default size of
/* 100 KB and a text block size of 1 KB. The global buffer pool is allocated
/* below 16 MB. The subsystem used is SAGS.
/* After the allocation, the job GBPRES will wait for further commands.
/* Further commands may be entered using AR command MSG partition-id:
/* the job GBPRES will then prompt for console input.
/*
// LIBDEF PHASE,SEARCH=SAGLIB.NATLIB
// EXEC NATGBPvr,SIZE=NATGBPvr,PARM='BPN=GBP,N=(,BL,1),S=SAGS,R=Y'
/*
// EXEC LISTLOG
/;&
```

例 3 :

```
// JOB GBPSTOP
/*
/* Stops the global buffer pool GPB if it contains no active objects. If it
/* does contain active objects, the operator console will prompt for a reply.
/* Depending on the reply, the shutdown will be forced (Y) or aborted (N).
/* The subsystem used is NATv.
/*
// LIBDEF PHASE,SEARCH=SAGLIB.NATLIB
// EXEC NATGBPvr,SIZE=NATGBPvr,PARM='FSHUT,BPN=GBP'
/*
// EXEC LISTLOG
/;&
```

例 4 :

```
// JOB GBPSTRT2
/* Read commands from SYSIPT:
/*
/* Start 3 global buffer pools (subsystem id Nvrs) with name
/*   NATGBP1 - size=1024KB and a cache with size 2048KB
/*   NATGBP2 - size=2048KB without cache
/* Display all buffer pools of subsystem id "Nvrs".
/*
/* Note: The job does not terminate by itself, but stays resident and waits
/*       for operator commands, because it owns the data space allocated for
/*       buffer pool NATGBP1.
/*
/* If the buffer pools should shut down, wake up sleeping job by MSG partition-id
/* and enter parameter "CF=*" to execute the corresponding FSHUTs.
/*
// LIBDEF PHASE,SEARCH=SAGLIB.NATLIB
// EXEC NATGBPvr,SIZE=NATGBPvr
CREATE,BPN=NATGBP1,S=Nvrs,N=(1M),BPC=2M
CREATE,BPN=NATGBP2,S=Nvrs,N=(2M)
SHOWBP S=Nvrs
/*
FSHUT,BPN=NATGBP1,S=Nvrs
FSHUT,BPN=NATGBP2,S=Nvrs
SHOWBP S=Nvrs
/*
```

ローカリゼーション

モジュール NATGBPTX はソース形式で提供されます。このモジュールには、すべてのエラーメッセージが英語（大文字と小文字が混在）で含まれています。メッセージは、必要に応じて他の言語に翻訳できます。この場合は、「新しい」 NATGBPTX ソースモジュールをアセンブルし、NATGBPvr をリンクし直す必要があります。

大文字の変数部分を含むグローバルバッファプールメッセージを発行するには、UCTRAN パラメータを YES に設定してグローバルバッファプールパラメータモジュール NATGBPRM をアセンブルし、NATGBPvr をリンクし直す必要があります。

NATGBPvr をリンクし直すには、以下の JCL を使用します。

```
// OPTION CATAL,LIST
ACTION NOAUTO,SMAP
PHASE NATGBPvr,*
INCLUDE NATGBPMG
INCLUDE NATGBPRM
INCLUDE NATGBPTX
INCLUDE NATBPMGR
ENTRY CMSTART
/*
```


47 z/OS および z/VSE 環境下の共通 Natural GBP オペ

レーティング機能

- グローバルバッファプールマネージャパラメータモジュール 310
- グローバルバッファプールのオペレーティング機能 310
- グローバルバッファプールの機能パラメータ 312
- NATBUFFER 指定の例 317

このドキュメントでは、z/OS および z/VSE 環境で共通する Natural グローバルバッファプールのオペレーティング機能の概要について説明します。

以下のトピックについて説明します。

グローバルバッファプールマネージャパラメータモジュール

グローバルバッファプールパラメータモジュール NATGBPRM は、すべての機能およびバッファプールに適用されるグローバル処理オプションを設定するために使用します。グローバルバッファプールパラメータモジュールは、ソース形式およびすべてデフォルト設定のオブジェクト形式で提供されます。

以下のパラメータを使用できます。

- UCTRAN - 小文字／大小文字混在サポート

UCTRAN - 小文字／大小文字混在サポート

このパラメータは、グローバルバッファプールメッセージの小文字／大小文字混在サポートを有効または無効にします。

UCTRAN=NO	小文字／大小文字混在サポートが完全に有効になります。 これはデフォルト値です。
UCTRAN=YES	すべてのグローバルバッファプールメッセージは大文字で表示されます。

グローバルバッファプールのオペレーティング機能

次の機能を使用できます。

- ADDCACHE - 既存のグローバルバッファプールへのキャッシュ割り当て
- CREATE - グローバルバッファプールの作成
- DELCACHE - グローバルバッファプールのキャッシュの解放
- FSHUT - グローバルバッファプールのシャットダウン
- GLOBALS - グローバルパラメータ設定の表示
- LISTCACHE - ジョブが所有するすべてのグローバルバッファプールキャッシュのリスト
- NOP - オペレーションなし
- REFRESH - グローバルバッファプールの再初期化
- SHOWBP - 既存のバッファプールの表示
- TERMINATE - GBP オペレーティングプログラムの終了

■ ZAPS - GBP に適用された Zap の表示



注意: 機能を指定しないと、プロファイルパラメータ BPNAME を指定している場合は **CREATE**、それ以外の場合は **NOP** と見なされます。

ADDCACHE - 既存のグローバルバッファプールへのキャッシュ割り当て

この機能は、キャッシュストレージを既存のグローバルバッファプールに追加します。

CREATE - グローバルバッファプールの作成

この機能は、指定されたパラメータでグローバルバッファプールを作成します。

DELCACHE - グローバルバッファプールのキャッシュの解放

この機能は、バッファプール自体はシャットダウンせずに、グローバルバッファプールのキャッシュストレージを削除します。

FSHUT - グローバルバッファプールのシャットダウン

グローバルバッファプールがシャットダウンされ、ストレージエリアが解放されます。

バッファプール内にアクティブなオブジェクトがない場合は、直ちに FSHUT が実行されます。

バッファプール内にアクティブなオブジェクトが残っている場合は、オペレータに通知されます。オペレータに確認を求めるか直ちに FSHUT を実行するかは、パラメータ **CONFIRM** の設定によります。

GLOBALS - グローバルパラメータ設定の表示

この機能は、ステートメントに適用されなかったパラメータも含めて、すべてのグローバルパラメータ設定を表示します。

また、グローバルバッファプールのストレージキーも表示されます。

LISTCACHE - ジョブが所有するすべてのグローバルバッファプールキャッシュのリスト

この機能は、ジョブが現在所有しているすべてのグローバルバッファプールキャッシュをリストします。

NOP - オペレーションなし

このファンクションコードは、特にグローバルパラメータを設定する際に使用されます。

REFRESH - グローバルバッファプールの再初期化

REFRESH コマンドでは、すでにアクティブになっているバッファプールを再初期化できます。ストレージ割り当ては行われなため、バッファプールのサイズと場所（16 MB より上または下）は元のままで変更されません。ただし、テキストブロックのサイズは変更できます（[NATBUFFER](#) パラメータを参照）。

この機能は、Current Use Count（「[SYSBPM-ディレクトリ情報](#)」の「バッファプールオブジェクトのフィールド」を参照）が0の場合（下記の「警告」を参照）またはバッファプールが破損した場合にのみ使用してください。

 **注意:** バッファプール内でアクティブなセッションによって Natural オブジェクトが実行されているときに、このバッファプールを再初期化すると、アクティブなセッションで予測できない結果が発生し、Natural が異常終了することもあります。

SHOWBP - 既存のバッファプールの表示

現在存在しているすべてのバッファプールを表示します。

TERMINATE - GBP オペレーティングプログラムの終了

GBP オペレーティングプログラムを終了します。この終了は、アクティブなグローバルバッファプールには影響しません。

ZAPS - GBP に適用された Zap の表示

グローバルバッファプールオペレーティングプログラムに適用されたすべての Zap を表示します。

グローバルバッファプールの機能パラメータ

Natural GBP オペレーティングプログラムの機能は、パラメータを使用して制御できます。これらのパラメータは任意の順序で指定できます。一意であれば省略形を使用することもできます。

 **注意:** キャッシュが関連付けられている複数のグローバルバッファプールを起動する場合は、単一のジョブまたは単一の開始タスク（z/OS 環境のみ）を使用して、入力データセットに個別に [CREATE](#) コマンドを指定することをお勧めします。「z/OS 環境下の Natural

「グローバルバッファプール」の「例 4」または「z/VSE 環境下の Natural グローバルバッファプール」の「例 4」を参照してください。

以下のパラメータを使用できます。

- BPNAME - グローバルバッファプールの名前
- BPLIST - プリロードリストの名前
- BPCSIZE - バッファプールキャッシュのサイズ
- CONFIRM - FSHUT の確認
- IDLE - チェックまでの待機時間
- METHOD - バッファプール内のスペースを割り当てるための検索アルゴリズム
- NATBUFFER - バッファサイズ、モード、テキストブロックサイズ
- RESIDENT - 機能実行後の動作
- SUBSID - Natural サブシステム ID
- TYPE - バッファプールのタイプ

BPNAME - グローバルバッファプールの名前

このパラメータは必須です（`TERMINATE` 機能を除く）。作成されるグローバルバッファプールの名前を指定します。

BPNAME= <i>name</i>	<i>name</i> は、グローバルバッファプールの 8 バイトの名前です。指定した名前が 8 バイト未満の場合は、空白が追加されます。
---------------------	---

`DELCACHE` 機能と `FSHUT` 機能の場合は、値 "*" を指定すると、特定の Natural サブシステムのすべてのバッファプールを処理できます。

BPLIST - プリロードリストの名前

このパラメータは、プリロードリストの名前を指定します。

BPLIST= <i>name</i>	<i>name</i> は、プリロードリストの 8 バイトの名前です。指定した名前が 8 バイト未満の場合は、空白が追加されます。
---------------------	---

BPCSIZE - バッファプールキャッシュのサイズ

このパラメータは、バッファプールキャッシュへのデータスペースの割り当てに使用されるストレージの容量 (KB 単位) を指定します。

BPCSIZE=size	<p>size は、バッファプールキャッシュへのデータスペースの割り当てに使用されるストレージの容量 (KB 単位) です。有効な範囲は 100~2097148 です。</p> <p>キャッシュサイズは、MB または GB 単位で指定することもできます。例えば、10 MB の場合は 10M と指定します。</p> <p>BPCSIZE パラメータを省略 (または 0 に設定) すると、バッファプールにキャッシュは提供されません。</p> <p>注意: キャッシュは、TYPE=NAT のバッファプールにのみサポートされます。</p>
--------------	--

CONFIRM - FSHUT の確認

このパラメータは、バッファプール内にアクティブなオブジェクトが残っている場合の FSHUT の動作を制御します。

CONFIRM=Y	<p>FSHUT 機能の実行には、オペレータの確認が必要です。オペレータは、FSHUT 機能を中止するか強制的に実行するかを決定できます。</p> <p>これはデフォルト値です。</p>
CONFIRM=N	<p>FSHUT は、オペレータの介入なしに強制的に実行されます。</p>

このパラメータは、このパラメータが指定されている FSHUT コマンドにのみ有効です。このため、CONFIRM は FSHUT パラメータごとに指定する必要があり、後続の FSHUT コマンドには適用されません。

IDLE - チェックまでの待機時間

このパラメータは、タスクにバッファプールキャッシュがない場合は無視されます。

IDLE=nn	<p>nn には秒数を指定します。この時間が経過すると、GBP オペレーティングプログラムは各バッファプールキャッシュをチェックして、関連付けられたバッファプールがまだアクティブかどうかを調べます。アクティブでない場合は、そのバッファプールキャッシュが解放されます。タスクが所有していた最後のバッファプールキャッシュが解放されると、RESIDENT=Y が指定されていない限り、タスクは終了します。</p> <p>デフォルト設定は 60 秒です。</p>
---------	---

IDLE は「グローバル」パラメータなので、一度指定すると、再度指定しなくても、後続のコマンドに IDLE が適用されます。



注意: z/OS 環境では、GBP オペレーティングプログラムは指定された IDLE 時間値とジョブのタイムアウト値の比較も行います。場合によっては、タイムアウトアベンド (S322) を防ぐために IDLE 時間値が内部的に減少されます。

METHOD - バッファプール内のスペースを割り当てるための検索アルゴリズム

このパラメータは、Natural バッファプールのストレージを割り当てるためにどのアルゴリズムを使用するかを制御します。

METHOD=S	ストレージを割り当てる際に選択プロセスを使用することを示します。選択プロセスでは、バッファプールディレクトリ全体を参照し、最適なエントリを見つけるために各エントリを比較します。このメソッドは以前はアルゴリズム 1+2 として知られていました。 これはデフォルト値です。
METHOD=N	次に使用可能な未使用スペースまたはフリースペースが使用されることを示します。次に使用可能なスペースの検索がラップアラウンド方式でポインタからディレクトリエントリに実行されます。このメソッドはバッファプールキャッシュと組み合わせて使用できます。

このパラメータは、CREATE 機能にのみ有効です。割り当てメソッドを変更するには、バッファプールを再起動します。

NATBUFFER - バッファサイズ、モード、テキストブロックサイズ

このパラメータは、バッファプールのサイズとモード、およびテキストブロックサイズを指定します。

NATBUFFER=(<i>size,mode,tsize</i>)	<i>size</i>	割り当てるストレージの容量 (KB 単位) です。 Natural バッファプール (TYPE=NAT) では、デフォルトおよび最小可能サイズは 256 KB です。 その他のバッファプールでは、デフォルトおよび最小可能サイズは 100 KB です。 指定したストレージ容量は、常に 4KB の倍数に切り上げられます。 プールサイズは、MB または GB 単位で指定することもできます。例えば、10 MB の場合は 10M と指定します。 <i>size</i> で指定したストレージの次に、1 ページ (4 KB) の書き込み保護されたストレージが管理目的で割り当てられます。
	<i>mode</i>	グローバルバッファプールを 16 MB より上または下のどちらに割り当てるかを指定します。 可能な値: XA (上、デフォルト)、BL (下)。

	<i>tsize</i>	テキストブロックサイズ (KB 単位) を指定します。 可能な値 : 1、2、4、8、12、および 16。デフォルトは 4 です。
<i>size</i> 、 <i>mode</i> 、および <i>tsize</i> は、上記の順序で指定する必要があります。		

NATBUFFER を指定しない場合は、デフォルト値が使用されます。後述の「[NATBUFFER 指定の例](#)」も参照してください。

RESIDENT - 機能実行後の動作

このパラメータは、特定の機能が実行された後の GBP オペレーティングプログラムの動作を指定します。可能な値は次のとおりです。

RESIDENT=Y	GBP オペレーティングプログラムは、特定の機能の実行後もアクティブのまま次のコマンドを待機します。RESIDENT=Y は一度指定すると後続のコマンドにも適用されるため、再度指定する必要はありません。GBP オペレーティングプログラムを停止するには、 TERMINATE 機能を使用します。
RESIDENT=N	GBP オペレーティングプログラムは、特定の機能の実行後、それ以上使用可能なコマンドがなければ終了します。タスクがバッファプールキャッシュを所有している場合は、RESIDENT=N は無視され、タスクは終了しません。
RESIDENT=A	GBP オペレーティングプログラムは、すべてのコマンドを処理した後の動作を自動的に決定します。以下の場合には終了します。 <ul style="list-style-type: none"> ■ それ以上使用可能なコマンドがない。 ■ このタスクによって作成されたキャッシュが関連付けられたバッファプールがない。 つまり、このタスクによって所有されているバッファプールキャッシュがない場合は、RESIDENT=A は RESIDENT=N と同様に動作します。タスクがバッファプールキャッシュを所有している場合は、RESIDENT=A は RESIDENT=Y と同様に動作しますが、このタスクが所有するバッファプールキャッシュが関連付けられた最後のバッファプールが終了すると、自動的に RESIDENT=N に切り替わります。 これはデフォルト設定です。

RESIDENT は「グローバル」パラメータなので、一度指定すると、明示的に指定または上書きするまで、後続のコマンドにも RESIDENT が適用されます。

SUBSID - Natural サブシステム ID

このパラメータは、Natural サブシステムの ID を指定します。

SUBSID= <i>id</i>	<i>id</i> は、Natural サブシステムの 4 バイトの ID です。 SUBSID は一度指定すると後続のコマンドにも適用されるため、再度指定する必要はありません。 デフォルト値は NAT <i>v</i> で、 <i>v</i> は現在の Natural バージョンの先頭の数字です。
-------------------	--

SUBSID は「グローバル」パラメータなので、一度指定すると、明示的に指定または上書きするまで、後続のコマンドにも SUBSID が適用されます。

DELCACHE 機能、FSHUT 機能、および SHOWBP 機能の場合は、値 "*" を指定すると、特定の Natural サブシステムのすべてのバッファプールを処理できます。

Natural サブシステムの詳細については、「[Natural サブシステム](#)」（z/OS）または「[Natural サブシステム](#)」（z/VSE）を参照してください。

TYPE - バッファプールのタイプ

このパラメータは、バッファプールのタイプを指定します。可能な値は次のとおりです。

TYPE=NAT	Natural バッファプール（デフォルト）
TYPE=SORT	ソートバッファプール
TYPE=DLI	DL/I バッファプール
TYPE=EDIT	エディタバッファプール
TYPE=MON	Monitor バッファプール
TYPE=RNM	Review Natural Monitor バッファプール

NATBUFFER 指定の例

以下の例では、バッファサイズ、モード、およびテキストブロックサイズの設定に使用される NATBUFFER パラメータを、省略形のパラメータ名 (N) で使用しています。

例1：16 MB より上、サイズ 1 MB、テキストブロックサイズ 1 KB のグローバルバッファプールを割り当てるには、以下のように指定します。

```
N=(1000,,1)
```

または

```
N=(1M,,1)
```

例2：16 MB より上、サイズ 10 MB、テキストブロックサイズ 4 KB のグローバルバッファプールを割り当てるには、以下のように指定します。

```
N=(10000)
```

または

```
N=(10M)
```

例3：16 MB より上、サイズ 256 KB、テキストブロックサイズ 4 KB のグローバルバッファプールを割り当てるには、以下のように指定します。

```
N=(,,)
```

これは、**NATBUFFER** パラメータを省略することと同じで、デフォルト値が適用されます。

48 BS2000/OSD 環境下の Natural グローバルバッファ プール

-
- BS2000/OSD 環境での Natural グローバルバッファプールの使用 320
 - BS2000/OSD 環境下のグローバルバッファプールの設定 320
 - BS2000/OSD 環境下のグローバルバッファプールの管理 321

このドキュメントでは、BS2000/OSD オペレーティングシステム環境での Natural グローバルバッファプール（GBP）の目的と使用方法について説明します。

以下の例で、*vrs* または *vr* は、対応するバージョン、リリース、およびシステムメンテナンスレベル番号を表します。製品バージョンの詳細については、『用語集』の「バージョン」を参照してください。

以下のトピックについて説明します。

BS2000/OSD 環境での Natural グローバルバッファプールの使用

Natural グローバルバッファプールは、BS2000バージョン10.0以降で使用できる共通メモリプールです。

これは、XS31 コンピュータでは、16 MB より下に、または 16 MB より上の拡張アドレススペースに配置できます。XS31 以外のコンピュータでは、クラス4のストレージ（サイズはオペレーティングシステムの生成方法によって異なる）より下のユーザーアドレススペースに配置できます。

グローバルバッファプールは、TIAM 環境の複数の Natural、UTM 環境の Natural、およびバッチアプリケーションで同時に使用できます。オペレーティングシステムごとに複数のグローバルバッファプールを持つことができます。

グローバルバッファプールは、最初の Natural アプリケーションの起動前にアクティブにする必要があります。最後の Natural セッションが終了した後でも、オペレーティングシステムがアクティブである限り、グローバルバッファプールをアクティブのままにしておくことができます。これにより、新しいセッションを開始したときにグローバルバッファプールの内容はまだ使用可能なので、再度バッファプールにロードする必要がありません。

BS2000/OSD 環境下のグローバルバッファプールの設定

グローバルバッファプールは、**CMPSTART** プログラムを起動するバッチジョブを実行することによって設定されます。グローバルバッファプールの名前、サイズ、仮想アドレスなどは、このジョブに指定したパラメータで決まります。

CMPSTART ジョブの例：

```
/SYSFILE SYSOUT.LST.BPvrsGA
/SYSFILE SYSDTA=(SYSCMD)
/EXEC (CMPSTART,$NATvrs.NATvrs.BS2.MOD)
NAME=BPvrsGA,TYPE=NAT,POSI=ABOVE,SIZE=2MB,ADDR=260,PFIX=NO,SCOP=GLOBAL
/SYSFILE SYSDTA=(PRIMARY)
```

パラメータ値が無効であるか BS2000/OSD 環境に適合していない場合は、バッファプールタスクはエラーメッセージを表示して終了します。エラーメッセージには、終了の原因と SVC リターンコード（該当する場合）が含まれます。すべてのエラーメッセージは SYSOUT に出力されます。重大なエラーの場合は、オペレータコンソールにも表示されます。

BS2000/OSD 環境下のグローバルバッファプールの管理

いったんグローバルバッファプールをアクティブにした後は、オペレータコンソールを使用して管理できます。

以下の BS2000/OSD コンソールコマンドを使用できます。 *tsn* はバッファプールタスクの TSN です。

コマンド	機能
/INTR <i>tsn</i> ,DPRM	グローバルバッファプールの現在のパラメータ設定と開始時間を表示します。
/INTR <i>tsn</i> ,SHUT /INTR <i>tsn</i> ,STOP	バッファプールタスクを正常に終了します。
/INTR <i>tsn</i> ,DUMP	バッファプールタスクを異常終了し、ダンプを生成します。

最後の Natural アプリケーションが終了するまで共通メモリプールはアクティブのままなので、バッファプールタスクの終了は必ずしもグローバルバッファプールの終了を意味しません。

このため、グローバルバッファプールをプログラムでも終了できるように、CMPEND プログラムが用意されています。

例：

```
/PROC C
/SYSFILE SYSDTA=(SYSCMD)
/EXEC (CMPEND,NATvrs.MOD)
name /* name of the global buffer pool
/SYSFILE SYSDTA=(PRIMARY)
/ENDP
```


49 Natural スワッププール

このドキュメントでは、以下のいずれかの TP モニタの使用時に利用できる Natural スワッププールについて説明します。

- **CICS** (Natural スワッププールはオプション)
- **UTM** (Natural スワッププールは必須)

Natural スワッププールの動作および機能は、どちらの環境でもほとんど同じです。相違や TP モニタ固有の機能については、本文中に示されています。

- **Natural** スワッププールの目的
- **Natural** スワッププールの操作
- スワッププールの初期化
- スワッププールのダイナミック再構成
- **Natural** スワッププールの定義
- **Natural** ユーザーエリアサイズの考慮事項
- スワッププールデータスペース
- **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
- グローバルスワッププールの終了

関連トピック：

- **CICS** 環境での *Natural* スワッププール (Natural の『TP モニタインターフェイス』ドキュメント)

- CICS 環境での *Natural* スワッププールの使用 (Natural の『TP モニタインターフェイス』ドキュメント)
- UTM 環境の *Natural* スワッププール (Natural の『TP モニタインターフェイス』ドキュメント)
- 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 (『メッセージおよびコード』ドキュメント)

50 Natural スワッププールの目的

■ Natural スワッププールの目的	326
■ Natural スワッププールを使用する利点	326
■ スワッププールの構造	327

このドキュメントでは、Natural スワッププールの目的、利点、および構造について説明します。

以下のトピックについて説明します。

Natural スワッププールの目的

オンラインの Natural ユーザーごとに、Natural ユーザーワークエリアが必要です。このワークエリアのサイズは、マクロ `NTSWPRM` のパラメータ `MAXSIZE` で指定します。ユーザーワークエリアは、ユーザーがダイアログトランザクションを開始するときには常に、コンピュータのメインストレージ内に存在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（または CICS 環境でのロール機能）にロールアウトされ再度ロールインされる頻度を減らすために、Natural スワッププールを設定することも可能です。

詳細については、「[Natural スワッププールの操作](#)」を参照してください。

Natural スワッププールを使用する利点

ユーザーワークエリアは、できる限り圧縮された形式で Natural スワッププールに保持されます。ディスクスワッピングの削減は、スワッププールのサイズ、圧縮された各 Natural ユーザーワークエリアのサイズ、およびオンラインユーザーの人数によって異なります。

すべてのオンラインユーザーのユーザーワークエリアがスワッププールに常駐し続けることが可能な場合は、ディスクスワッピングは実行されません。

TP モニタ :	コメント :
CICS	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 <code>BPI</code> または Natural パラメータモジュール <code>NATPARM</code> の対応するマクロ <code>NTBPI</code> で指定します。これにより、Natural CICS 環境を初期化する Natural セッションに有効な (NT) <code>BPI</code> 設定が使用されます。
UTM	Natural スワッププールのサイズは、マクロ <code>ADDON</code> のキーワードパラメータ <code>SIZE</code> またはプログラム <code>CMPSTART</code> で指定します (スワッププールデータスペース生成用のキーワードパラメータ <code>DATA</code> および <code>DESA</code> も参照)。

スワッププールの構造

物理スワッププールは、以下の要素で構成されます。

- メインディレクトリ
- 論理スワッププール（以下を含む）
 - サブディレクトリ
 - スワッププールスロット

スワッププールメインディレクトリ

スワッププールメインディレクトリは、スワッププール全体を参照します。最大 15 の論理スワッププールを定義できます。

サブディレクトリ

各論理スワッププールには、独自のサブディレクトリが含まれます。

スワッププールスロット

スワッププールスロットでは、Natural ユーザーワークエリアが圧縮形式で保持されます。

スワッププールを初めて初期化するとき、マクロ `NTSWPRM` のパラメータ `SWPSLSZ` を使用して、論理スワッププールの数とそのスロットのサイズを定義し、スワッププールパラメータモジュールを生成することができます。

論理スワッププール

各論理スワッププールには、サブディレクトリとゲストテーブルが含まれます。

使用される各スワッププールディレクトリエントリは、その前後のエントリに連結されています。これは、ゲストテーブルのエントリにも該当します。こうすることで、最新および最も古いスワッププールユーザー/ゲストが常に認識されます。

論理スワッププールのゲストの定義は、以下の例のようになります。

3つの論理スワッププール（LSP）を持つスワッププールがあります。

- LSP 1 のスロットサイズは 62 KB です。
- LSP 2 のスロットサイズは 72 KB です。
- LSP 3 のスロットサイズは 82 KB です。

圧縮 Natural ユーザーワークエリアのサイズは 60 KB であるため、このユーザーワークエリアは LSP 1 のスロットに圧縮する必要があります。この例で、LSP 1 が現在いっぱい、LSP 2 に空きスロットがある場合は、ユーザーワークエリアは LSP 2 に圧縮されます。LSP 2 がいっぱい、LSP 3 に空きスロットがある場合は、ユーザーワークエリアは LSP 3 に圧縮されます。独自の LSP がいっぱいなので、LSP 2 または 3 のユーザーワークエリアはこれらの LSP のゲストになります。

TP モニタ :	コメント :
CICS	CICS 環境では、Natural スワッププールはオプションです。CICS コマンドレベルのオーバーヘッドのため、スワッププールとのスワッピングは高価なロール I/O より高速です。ただし、仮想ストレージがボトルネックの場合は、ページングオーバーヘッドのため、スワッププールのインストールによってパフォーマンスが低下することがあります。Natural の『TP モニタインターフェイス』ドキュメントの「CICS 環境での 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」も参照してください。
UTM	UTM 環境では、Natural スワッププールは必須です。「UTM 環境の Natural スワッププール」も参照してください。

51 Natural スワッププールの操作

- ユーザーが Natural への途中で、セッションが開始されていない場合 330
- ユーザーが Natural から戻ってくる場合 330

以下の状況について説明します。

ユーザーが Natural への途中で、セッションが開始されていない場合

ユーザーのワークエリアがスワッププールに保持されている場合は、対応するスロットが読み取られ、Natural ユーザースレッドに圧縮解除されます。対応するスワッププールディレクトリエントリは、ディレクトリのチェーンからリンク解除され、未使用のエントリとして宣言されます。エントリがゲストの場合は、ゲストテーブルが更新されます。

ユーザーのワークエリアがスワッププールに保持されていない場合は、ワークエリアがデータスペースまたはスワップファイル（または CICS 環境でのロール機能）から読み取られ、Natural ユーザースレッドに圧縮解除されます。

Natural がアクティブになります。

ユーザーが Natural から戻ってくる場合

Natural は、ユーザーワークエリアの圧縮長が論理スワッププールの最大スロットサイズを超えるかどうかをチェックします。

最大スロットサイズを超える場合は、ユーザーワークエリアが圧縮され、非同期的にスワップファイルに書き込まれるか、CICS 環境でのセッションに関連付けられたロール機能にロールされます。

最大スロットサイズを超えない場合は、Natural はユーザーワークエリア独自のスワッププールに未使用のスロットがあるかどうかを確認します。

- 未使用のスロットがある場合は、ユーザーワークエリアがこのスロットに圧縮されます。対応するディレクトリエントリは、最新のエントリとしてディレクトリチェーンにリンクされます。
- 未使用のスロットがない場合は、Natural はユーザーワークエリア独自の論理スワッププールにゲストが存在するかどうかを確認します。

ゲストが 1 つまたは複数存在する場合は、スロットが使用可能にされます。最も古いゲストテーブルエントリがゲストテーブルからリンク解除され、その次に古いエントリが最も古いゲストになります。適切なディレクトリエントリが、ディレクトリチェーンからリンク解除されます。

ゲストが存在しない場合は、スロットが使用可能にされます。最も古いディレクトリエントリがディレクトリチェーンからリンク解除され、その次に古いエントリが最も古いゲストになります。

- ESA データスペースが生成され、未使用のスロットが使用可能な場合は、スレッドがスワップファイルにロールアウトされる前にこのスロットが使用されます。

リンクされていないユーザーの圧縮ユーザーエリアは、書き込みバッファに転送され、スワップファイルに非同期的に書き込まれるか、CICS 環境でのセッションに関連付けられたロール機能に同期的にロールされます。現在のユーザーのワークエリアは、使用可能になったスロットに圧縮されます。対応するディレクトリエントリは、最新のエン트리としてディレクトリチェーンにリンクされます。

スワッププール再構成およびスロットサイズ計算の統計テーブルが更新されます。

52 Natural スワッププールの初期化

- スワッププールの初期化の制御 334
- スワッププール初期化パラメータ 335

このドキュメントでは、Natural スワッププールの初期化を制御する方法について説明し、マクロ NTSWPRM で初期化に使用できるキーワードパラメータの概要を示します。

以下のトピックについて説明します。

スワッププールの初期化の制御

マクロ NTSWPRM のパラメータ SWPINIT は、スワッププールの初期化を制御します。

SWPINIT=AUTO に設定した場合

- スワッププールマネージャは、Natural システムファイル FNAT または FUSER からスワッププール名をキーとしてスワッププール初期化データの読み取りを試みます（マクロ NTSWPRM のキーワードパラメータ SWPFILE を参照）。データが検出された場合は、それらが使用され、マクロ NTSWPRM の対応するパラメータは無視されます。データが検出されなかった場合は、マクロ NTSWPRM のキーワードパラメータ SWPSLSZ のオペランドがスワッププールの初期化に使用されます。
- パラメータ SWPSLSZ が 1 つのスロットサイズ定義のみを含んでいる場合は、スワッププールは 1 つの論理スワッププールで初期化されます。スワッププールマネージャは、指定された時間間隔（マクロ NTSWPRM のパラメータ SWPTIM1 を参照）で、スワッププールの再編成または最適化が必要かどうかを制御します（「スワッププールのダイナミック再構成」も参照）。スワッププールが再構成された場合は、スワッププール用に新しく計算された初期化データは、次の初期化のために Natural システムファイルに保存されます。スワッププールの再構成によって複数の論理スワッププールが生じる場合は、スワッププールのダイナミック再構成はそれ以上行われません。
- スワッププールに複数の論理スワッププールがある場合は、スワッププールをダイナミックに再構成することはできません。
- これ以上のスワッププールの最適化は、以下の Natural SYSTP ユーティリティ機能で明示的に初期化することができます。
 - スロットサイズの計算
 - スワッププールパラメータサービス（Natural システムファイル内のスワッププール初期化データの変更）
 - スワッププールの無効化およびスワッププールの有効化
- スワッププールをダイナミックに再構成または最適化するための論理スワッププールの最大数は、マクロ NTSWPRM のキーワードパラメータ SWPLSWP のオペランドで定義できます。

SWPINIT= に設定した場合

- Natural システムファイル内のスワッププール初期化データは、読み取りも保存も行われません。マクロ NTSWPRM のキーワードパラメータ SWPSLSZ のオペランドが、スワッププールの初期化に使用されます。
- スワッププールをダイナミックに再構成または最適化するためのルールは、初期化データが Natural システムファイルに保存されないことを除き、上記の SWPINIT=AUTO と同じです。

スワッププール初期化パラメータ

以下に、マクロ NTSWPRM で初期化に使用できるキーワードパラメータの概要を示します。

パラメータ	説明
SWPSLSZ	論理スワッププールの数、そのスロットサイズ、およびスロット番号と論理スワッププールとの数値関係を定義します。
SWPINIT	Natural システムファイルでのスワッププール初期化データへのアクセスを定義します。
SWPLSWP	スワッププールをダイナミックに再構成するための論理スワッププールの最大数を定義します。
SWPSDIF	異なる論理スワッププールのスロットサイズ間の最小差（偶数）を定義します。この値は、スロットサイズの計算中とスワッププールのダイナミック再構成中に制御されます。

以下の TP モニタ固有の要件が適用されます。

- **UTM 環境：**
スワッププールのサイズは、マクロ ADDON またはプログラム CMPSTART のキーワードパラメータ SIZE のオペランドで指定する必要があります。
- **CICS 環境：**
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 の対応するマクロ NTBPI で指定します。これにより、Natural CICS 環境を初期化する Natural セッションに有効な (NT) BPI 設定が使用されます。

53 スワッププールのダイナミック再構成

■ スワッププールのダイナミック再構成の要件	338
■ 統計テーブル	338
■ スワッププールの再構成プラステーブル	338
■ スワッププールの再構成マイナステーブル	339
■ スワッププール再構成用のパラメータ	339
■ スワッププール再構成の必要性チェック	340
■ スワッププールのダイナミック再構成の流れ	340
■ スワッププールのダイナミック再構成の開始	341

このドキュメントでは、スワッププールのダイナミック再構成の前提条件、処理、制御、および開始について説明します。

以下のトピックについて説明します。

スワッププールのダイナミック再構成の要件

スワッププールをダイナミックに再構成できるのは、物理スワッププールに含まれる論理スワッププールが1つの場合のみです。この場合、スワッププールスロットはすべて同じサイズです。必要に応じて、要件を満たすように論理スワッププール数とスロットサイズを調整できます。スロットサイズは、スワッププールをダイナミックに再構成することにより調整します。

統計テーブル

スワッププールディレクトリの統計エリアには、スワッププールの再構成に使用する以下の2つの統計テーブルが含まれています。

- スワッププールの再構成プラステーブル
- スワッププールの再構成マイナステーブル

スワッププールの再構成プラステーブル

スワッププールの再構成プラステーブルには、圧縮長がスワッププールのスロットサイズを超えるためにスワッププールに配置できない Natural ユーザーエリアに関する情報が含まれていません。

テーブルには、以下の 11 のエントリがあります。

- 最初の 9 つのエントリは、スロットサイズを上回る長さが 1~9 ユニットのユーザーエリアの数を数えます。
- 10 番目のエントリは、スロットサイズを上回る長さが 9 ユニットを超えるユーザーエリアの数を数えます。
- 11 番目のエントリには、10 番目のエントリが数えるユーザーエリアの平均長が含まれます。

スワッププールの再構成マイナステーブル

スワッププールの再構成マイナステーブルには、圧縮長がスワッププールのスロットサイズよりも小さい Natural ユーザーエリアに関する情報が含まれています。

テーブルには、以下の 11 のエントリがあります。

- 最初の 9 つのエントリは、スロットサイズを下回る長さが 1~9 「ユニット」 のユーザーエリアの数を数えます。
 - 10 番目のエントリは、スロットサイズを下回る長さが 9 ユニットを超えるユーザーエリアの数を数えます。
 - 11 番目のエントリには、10 番目のエントリが数えるユーザーエリアの平均長が含まれます。
- 「ユニット」 のサイズは、キーワードパラメータ `SWPFACT` で定義します。

スワッププール再構成用のパラメータ

スワッププールのダイナミック再構成は、マクロ `NTSWPRM` の以下のキーワードパラメータで制御されます。

パラメータ	指定内容
SWPSLSZ	スワッププールの初回初期化用のスロットサイズ。デフォルトのサイズは 62 KB です。
SWPTFIX	スロットサイズを固定にするかダイナミックにするか。スロットサイズを固定にした場合は、スワッププールはダイナミックに再構成されません。スロットサイズを固定に定義しなかった場合は、スワッププールは必要に応じてダイナミックに再構成されます（これがデフォルトです）。
SWPTIM1	スワッププールの再構成が必要かどうかを確認するためにチェックを実行する時間間隔。デフォルトでは、チェックは 30 分ごとに実行されます。
SWPTIM2	スワッププールの再構成が必要かどうかをチェックしてから再構成を開始するまでの経過時間。デフォルトでは、再構成は、再構成が必要であると確認されてから 2 分後に開始されます。
SWPUSER	実際の SWP スロット長には長すぎる圧縮ユーザースレッドの割合 (%)。この値に達し、物理 SWP に 1 つの論理スワッププールしか含まれていない場合は、SWP の再構成が通知されます。
SWPFACT	スワッププールの再構成プラステーブルおよびマイナステーブルの「ユニット」のファクタ。

スロットサイズの最適化が効率的に実行されていない場合以外は、これらのパラメータのデフォルト値を変更する必要はありません。

テストおよび最適化のために、Natural ユーティリティ SYSTP の一部である Natural スワッププールマネージャを使用して、これらのパラメータの値をオンラインでダイナミックに変更できます。

スワッププール再構成の必要性チェック

チェックは以下に基づきます。

- チェック間のダイアログステップの総数
- `SWPUSER` パラメータで定義された比率
- `SWPLSWP` パラメータで定義された論理スワッププールの最大数
- 異なる論理スワッププールのスロットサイズの最小差
- スワッププールの再構成プラステーブルおよびマイナステーブルの値（これらのテーブルは `SWPFACT` パラメータの設定に影響されます）
- 物理スワッププールの合計サイズ

対応するスロットサイズを持つ必要な論理スワッププールの数は、圧縮長が現在のスロットサイズよりも最低1ユニット大きいか小さいユーザーエリアの数がダイアログステップ数より $n\%$ （ n は `SWPUSER` パラメータの値）多い場合に計算されます。

スワッププールが再構成されるときは、新しい論理スワッププールが使用されます。再構成後に物理スワッププールに複数の論理スワッププールが含まれている場合は、スワッププールのダイナミック再構成はそれ以上行われません。

スワッププールのダイナミック再構成の流れ

Natural は、物理スワッププールに含まれる論理スワッププールが1つの場合のみ、スワッププールの再構成が必要かどうかをチェックします。

`SWPTIM1` パラメータで指定された時間が経過すると、スワッププールの再構成が必要かどうかを確認するためのチェックが実行されます。

- スワッププールの再構成が不要な場合は、`SWPTIM1` パラメータで設定されたタイマー（チェック間の時間間隔）が再度アクティブになります。
- スワッププールの再構成が必要な場合は、`SWPTIM2` パラメータで設定されたタイマー（チェックの終了から再構成の開始までの時間間隔）がアクティブになります。それ以上のユーザーエリアをスワッププールに配置することはできません。スワッププールに保持されるユーザーエリアは、引き続き使用でき、ユーザースレッドに読み込むことができます。この2番目の時間間隔が経過すると、スワッププールの再構成が開始されます。

スワッププールのダイナミック再構成の開始

SWPTIM2 パラメータで指定された時間が経過した後、現在のオンラインセッション中にスワッププールが再構成されます。

1. スワッププールに引き続き保持される圧縮ユーザーエリアは、スワップファイル（または CICS 環境でのロール機能）に書き込まれます。
2. スワッププール再構成の統計テーブルの内容は、SYSLST に書き込まれ、テーブルから削除されます。
3. スワッププールは、新しく計算された値で再度初期化されます。
4. SWPTIM1 パラメータで設定されたタイマー（チェック間の時間間隔）が再度アクティブになります。

Natural ユーティリティ SYSTP (Natural 『ユーティリティ』ドキュメントを参照) の一部である Natural スワッププールマネージャを使用して、スワッププールの統計データと Natural バッファおよびユーザースレッドのサイズに関する情報を取得できます。

54 Natural スワッププールの定義

- 環境固有の要件 344
- マクロ NTSWPRM のキーワードパラメータ 344

DSPCONT - データスペースのスロット制御の時間 (分単位)

このパラメータは、ESA データスペースエリアがいっぱいのときにデータスペースの制御が実行されるまでの時間 (分単位) を定義します。この時間が経過すると、データスペース内のスロットがチェックされ、スレッドの存続期間が期限切れになっていないかどうかを確認されます。期限切れになっている場合は、影響を受ける各スロットの圧縮 Natural ユーザースレッドがロールファイルにロールアウトされます。

DSPCONT= <i>nnn</i>	<i>nnn</i> は 1~480 の範囲で指定する必要があります。
DSPCONT=10	デフォルト値は 10 (分) です。

DSPLIFE - ESA データスペース内のスレッドの存続期間 (分単位)

このパラメータは、ESA データスペースのスロット内にある圧縮 Natural ユーザースレッドの存続期間を定義します。データスペースのスロット制御ロジックがアクティブの場合は、存続期間が期限切れになるとスレッドがロールアウトされます。スレッドの存続期間は、スレッドが ESA データスペースに書き込まれたときに開始します。

DSPLIFE= <i>nn</i>	<i>nn</i> は 1~60 の範囲で指定する必要があります。
DSPCONT=5	デフォルト値は 5 (分) です。

SWPFILE - スワッププール初期化データの場所

このパラメータは、機能 `SWPINIT=AUTO` を使用する場合に、スワッププール初期化データを Natural システムファイル FNAT に保存するか FUSER に保存するかを定義します。

SWPFILE=FNAT/FUSER	スワッププール初期化データのファイル定義。
SWPFILE=FNAT	デフォルト値は FNAT です。

MAXSIZE - Natural ユーザースレッドのサイズ

このパラメータは、Natural ユーザースレッドのサイズ *nnn* (KB 単位) を定義します。このサイズを決める方法については、「[MAXSIZE パラメータの使用](#)」を参照してください。

MAXSIZE= <i>nnn</i>	<i>nnn</i> は 64~32768 の範囲で指定する必要があります。
MAXSIZE=400	デフォルト設定は 400 (KB) です。

CICS 環境では、Natural CICS インターフェイスがこのパラメータの最大のスレッドのサイズを自動的に使用するため、このパラメータの指定は無視されます。

SWPFACT - 再構成テーブルのユニットのサイズ

このパラメータで指定するファクタ n は、スワッププールの再構成プラステーブルおよびマイナステーブルの「ユニット」のサイズを決めます。

SWPFACT= n	n に指定できる値は 0~4 です。 n により、「ユニット」のサイズは以下のようになります。
	0 の場合は 2 KB
	1 の場合は 4 KB
	2 の場合は 8 KB
	3 の場合は 16 KB
	4 の場合は 32 KB
SWPFACT=1	デフォルト設定は 4 KB です。

これらのテーブルを使用して、スロットサイズを計算し、スワッププールをダイナミックに再構成し、スワッププール統計を取得します。「[スワッププールのダイナミック再構成](#)」を参照してください。

SWPINIT - スワッププール初期化データへのアクセス

このパラメータは、Natural システムファイルでのスワッププール初期化データへのアクセスを定義します。

SWPINIT=	空白。「スワッププールの初期化の制御」の「 SWPINIT= に設定した場合 」を参照してください。
SWPINIT=AUTO	これはデフォルト設定です。スワッププール初期化データは、Natural システムファイルから読み取られるか、Natural システムファイルに保存されます。「スワッププールの初期化の制御」の「 SWPINIT=AUTO に設定した場合 」も参照してください。

このパラメータの使用方法の詳細については、「[Natural スワッププールの初期化](#)」を参照してください。

SWPLSWP - 論理スワッププールの数

このパラメータは、使用する論理スワッププールの最大数 n を定義します。

SWPLSWP= n	n に指定できる値は 0~15 です。
SWPLSWP=0	以下の注 3 を参照してください。

注：

1. 論理スワッププールの最小サイズは 64 KB です。
2. 定義する値は、パラメータ **SWPSLSZ** で定義したスロットサイズの数値より小さくすることはできません。
3. デフォルト値の 0 を使用すると、スワッププールマネージャによって論理スワッププールの最大サイズが計算されます。
4. このパラメータは、スワッププール初期化データを Natural システムファイルから読み取ることができる場合は無視されます。

SWPPWRD - 管理パスワード

このパラメータを使用して、Natural ユーティリティ SYSTP のスワッププールマネージャサブシステムに、スワッププール再構成制御データおよびバッファ使用統計の管理用パスワードを指定します。

SWPPWRD= <i>password</i>	<i>password</i> の最大長は 4 文字です。
SWPPWRD=ADMI	これはデフォルト値です。

SWPSDIF - スロットサイズの最小差

このパラメータを使用して、複数の論理スワッププール内のスロットサイズの最小差を指定します。

SWPSDIF= nn	nn は偶数である必要があり、キロバイト数 (KB) を指定します。 nn は 2~98 の範囲で指定する必要があります。
SWPSDIF=8	デフォルト値は 8 KB です。



注意: このパラメータは、スワッププール初期化データを Natural システムファイルから読み取ることができる場合は無視されます。

SWPSLSZ - 論理スワッププールの数、スロットサイズ

このパラメータは、論理スワッププールの数、スロットサイズ、および異なる論理スワッププール間のスロット数の関係を指定します。可能な値は次のとおりです。

SWPSLSZ=(<i>nn</i> , <i>f</i> (, <i>nn</i> , <i>f</i> ...)) SWPSLSZ=(<i>nn</i> (, <i>nn</i> ...), <i>f</i> (, <i>f</i> ...)) SWPSLSZ=(<i>nn</i> (, <i>nn</i> ...))	<i>nn</i>	論理スワッププールのスロットサイズをKB単位で指定します（偶数である必要があります）。 <i>nn</i> は20~998の範囲で指定する必要があります。
	<i>f</i>	異なる論理スワッププールのスロット数間の数値ファクタの関係を指定します。 <i>f</i> は1~9の範囲で指定する必要があります。
SWPSLSZ=(62,1)	デフォルトのスロットサイズは62KBです。デフォルトの関係は1です。	

例：

```

SWPLSZ=(44,1,62,2)
/* SWAP POOL SIZE IS 2048 KB
/* THERE WILL BE TWO LOGICAL SWAP POOLS, RELATION BETWEEN THEM IS 1:2
/* 1 LOGICAL SWAP POOL WITH 12 (1) 44-KB SLOTS
/* 1 LOGICAL SWAP POOL WITH 24 (2) 62-KB SLOTS<

SWPLSZ=(64,80,96)
/* SWAP POOL SIZE IS 8 MB
/* THERE WILL BE THREE LOGICAL SWAP POOLS, RELATION BETWEEN THEM IS 1:1:1
/* 1 LOGICAL SWAP POOL WITH 34 (1) 64-KB SLOTS
/* 1 LOGICAL SWAP POOL WITH 34 (1) 80-KB SLOTS
/* 1 LOGICAL SWAP POOL WITH 34 (1) 96-KB SLOTS
    
```

このパラメータは、スワッププール初期化データをNaturalシステムファイルから読み取ることができる場合は無視されます。

SWPTFIX - 固定スロットサイズ

このパラメータは、スワッププールスロットのサイズを固定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。

SWPTFIX=Y	SWPSLSZパラメータ（上記参照）で定義したスロットサイズが固定サイズとして見なされ、スワッププールは再構成されません。
SWPTFIX=N	これはデフォルト値です。SWPSLSZパラメータ（上記参照）で定義したスロットサイズが固定サイズとして見なされず、スワッププールが必要に応じて再構成されます。つまり、スロットのサイズが、実際の要件を満たすように動的に調整されます。

 **注意:** このパラメータは、物理スワッププールに複数の論理スワッププールが含まれている場合は無視されます。

SWPTIM1 - 再構成チェックの時間間隔

このパラメータを使用して、スワッププールの再構成が必要かどうかを確認するためにチェックを実行する時間間隔 *nnn* を指定します。可能な値は次のとおりです。

SWPTIM1= <i>nnn</i>	<i>nnn</i> は 1~540（分）の範囲で指定する必要があります。
SWPTIM1=(<i>nnn</i> , RESET)	スワッププール再構成の統計テーブルの内容は、チェックの後に削除されます（通常はスワッププールの再構成後にのみ削除されます）。
SWPTIM1=30	デフォルト値は 30（分）です。

チェックとスワッププールの再構成が実行される仕組みについては、「[スワッププールのダイナミック再構成](#)」を参照してください。

 **重要:** パラメータ SWPTFIX が Y に設定されているか、物理スワッププールに複数の論理スワッププールが含まれる場合は、SWPTIM1 パラメータは適用されません。

SWPTIM2 - 再構成の開始までの経過時間

このパラメータを使用して、スワッププールの再構成が必要かどうかをチェックしてから実際の再構成を開始するまでの経過時間 *nn* を指定できます。

SWPTIM2= <i>nn</i>	<i>nn</i> は 1~99（分）の範囲で指定する必要があります。
SWPTIM2=2	デフォルト値は 2（分）です。

この時間中は、ユーザーエリアをそれ以上スワッププールに配置することはできませんが、スワッププール内に引き続き保持されているユーザーエリアは Natural ユーザースレッドで使用したり読み取ったりすることができます。

チェックとスワッププールの再構成が実行される仕組みについては、「[スワッププールのダイナミック再構成](#)」を参照してください。

パラメータ SWPTFIX が Y に設定されているか、物理スワッププールに複数の論理スワッププールが含まれる場合は、SWPTIM2 パラメータは適用されません。

SWPUSER - スワッププール再構成の条件

このパラメータを使用して、スワッププールの再構成を実行するために満たす必要のある条件を定義できます。

SWPUSER= <i>nn</i>	<i>nn</i> は 1～99 の範囲で指定する必要があります。
SWPUSER=20	デフォルト値は 20 (%) です。

圧縮ユーザーエリアの長さが現在のスロットサイズよりも 1 ユニット以上大きい（または 1 ユニット以上小さい）すべてのユーザーのダイアログステップの比率を決めるパーセント値 *nn* を定義できます。この比率に達したことがチェックで確認されると、スワッププールの再構成が実行されます。

チェックが実行される仕組みについては、「[スワッププールのダイナミック再構成](#)」を参照してください。

パラメータ SWPTFIX が Y に設定されているか、物理スワッププールに複数の論理スワッププールが含まれる場合は、SWPUSER パラメータは適用されません。

NOVPA - 完了した非同期書き込みの待機数

このパラメータは、完了した非同期書き込みの待機数を指定します。

NOVPA= <i>nnn</i>	<i>nnn</i> は 1～999 の範囲で指定する必要があります。
NOVPA=20	デフォルト値は 20 (待機数) です。

NOVPW - ロックされていないスワッププールの待機数

このパラメータは、ロックされていないスワッププールの待機数を指定します。

NOVPW= <i>nnn</i>	<i>nnn</i> は 1～999 の範囲で指定する必要があります。
NOVPW=15	デフォルト値は 15 (待機数) です。

WAITMS - ロックされていないスワッププールの待機時間

このパラメータは、ロックされていないスワッププールの1回の待機時間をミリ秒単位で指定します。

WAITMS= <i>nnn</i>	<i>nnn</i> は 1~999 の範囲で指定する必要があります。
WAITMS=5	デフォルト値は 5 (ミリ秒) です。

WRITMS - 完了した非同期書き込みの待機時間

このパラメータは、完了した非同期書き込みの1回の待機時間をミリ秒単位で指定します。

WRITMS= <i>nnn</i>	<i>nnn</i> は 1~999 の範囲で指定する必要があります。
WRITMS=10	デフォルト値は 10 (ミリ秒) です。

55 Natural ユーザーエリアサイズの考慮事項

▪ MAXSIZE パラメータの使用	354
▪ 個別の Natural バッファのサイズの定義	354
▪ 可能性のあるエラーメッセージ	354
▪ すべてのバッファの集計サイズの表示	355
▪ 最大サイズの計算	355

このドキュメントでは、Natural ユーザーエリアのサイズと個別の Natural バッファのサイズを管理する方法について説明します。

以下のトピックについて説明します。

MAXSIZE パラメータの使用

Natural ユーザーエリアの全体的なサイズは、スワッププールパラメータモジュールの **MAXSIZE** パラメータによって決まります。そのため、MAXSIZE は、Natural で必要とされる、またサブシステム（Con-nect、TRS など）で使用される可能性のある、すべてのバッファの集計サイズを含めるのに十分な大きさに設定する必要があります。Natural およびサブシステムのバッファ要件は、TP ドライバで満たされます。Natural アプリケーションが開始されると、サイズが MAXSIZE のユーザースレッドが作成されます。これは、オペレーティングシステムへの物理要求メモリによって行われます。

Natural から TP ドライバへのバッファ要求では、「論理的な」GETMAIN しか発生しません。つまり、Natural ユーザースレッドはその後、「論理」ユニットの Natural バッファに分割されません。

個別の Natural バッファのサイズの定義

個別の Natural バッファのサイズは、Natural パラメータモジュールのパラメータ **ESIZE**（ユーザーバッファ拡張エリアのサイズ）や **CSIZE**（Con-nect バッファエリアのサイズ）などで明示的に定義されるか、パラメータ **PS**（Natural レポートのページサイズ）や **LS**（ラインサイズ）などの定義によって暗黙的に決定されます。

Natural バッファの最大サイズは、Natural ユーティリティ **SYSTP** のバッファ使用統計機能で表示できます。SYSTP は、特定のアプリケーションのすべてのユーザーに使用される全体的な最大 Natural バッファサイズを確認する機能も提供します。

可能性のあるエラーメッセージ

Natural エラーメッセージ「NOT ENOUGH MEMORY」または「BUFFER SIZES EXCEED MAXSIZE」が表示された場合、これは MAXSIZE パラメータの値が十分な大きさに定義されていないことを示しています。

すべてのバッファの集計サイズの表示

Naturalによって要求されるすべてのバッファの集計サイズ（つまり、アプリケーションのユーザーによって実際に使用される MAXSIZE の量）は、SYSTP ユーティリティの *Natural* スワップ情報機能を使用して取得できます。

最大サイズの計算

MAXSIZE の標準的な計算方法は以下のとおりです。

明示的に定義されたすべてのバッファサイズ（ESIZE など）と 40 KB（内部 Natural バッファサイズの合計）を足します。

これにより、MAXSIZE に必要なサイズの概数を得ることができます。

56 スワッププールデータスペース

- 追加の ESA データスペースの使用 358
- ESA データスペーススロットサイズの調整 358

このドキュメントでは、ESA データスペースを生成することで Natural スワッププールの容量を拡張する方法について説明します。

以下のトピックについて説明します。

追加の ESA データスペースの使用

スワップ I/O オペレーションをさらに減らすために、**CMPSTART** プログラムのキーワードパラメータ **DATA** および **DESA** を使用して、ESA データスペースを生成することにより、Natural スワッププールの容量を拡張できます。このデータスペースは、スワッププールでスペースが不足したときに圧縮 Natural ユーザーレッドを保存するために使用できます。

このデータスペースも消費されると、ユーザーレッドの存続期間が終了したためにユーザーレッドをデータスペースからロールファイルに書き込む必要があるかどうかをチェックされます（マクロ **NTSWPRM** のキーワードパラメータ **DSPCONT** および **DSPLIFE** を参照）。

データスペースに空きストレージスペースがない場合は、スワッププールロジックによって最も古い非アクティブのユーザーレッドがスワッププールからロールファイルに書き込まれます。

ESA データスペーススロットサイズの調整

生成された ESA データスペースは、同じサイズのスロットに分割されます。

- TP モニタ UTM を使用している場合は、**NATUTM** マクロキーワードパラメータ **ROLLTSZ** を適切に設定することでスロットサイズを定義できます。
- TP モニタ CICS を使用している場合は、データスペースのスロットサイズは最長のスレッドのサイズを自動的に取得します。

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** の対応するマクロ **NTBPI** で指定します。これにより、Natural CICS 環境を初期化する Natural セッションに有効な (NT) **BPI** 設定が使用されます。

57 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

-
- UTM 環境下の Natural グローバルスワッププールの目的 360
 - UTM 環境下の Natural グローバルスワッププールのインストール 360
 - UTM 環境下の Natural グローバルスワッププールの開始 361
 - グローバルスワッププールに関する情報の表示 361

このドキュメントでは、UTM 環境の Natural で Natural グローバルスワッププールをインストールおよび操作する方法について説明します。

以下のトピックについて説明します。

UTM 環境下の Natural グローバルスワッププールの目的

UTM 環境の Natural アプリケーションのすべてのタスクが異常終了した場合は、ローカル Natural スワッププールの内容が削除されます。そのため、タスクが再開されると、新しいスワッププールが初期化され、異常終了の影響を受けたすべてのユーザーが Natural セッションを再度開始する必要があります。

この状況を回避するために、グローバル（つまり再スタート可能な）スワッププールを使用できます。UTM 環境の Natural アプリケーションが異常終了した後、ユーザーがアプリケーションに再度ログオンするときに、終了前に表示された最後の画面が再度送信され、ユーザーは中断されたポイントからセッションを再開できます。

UTM 環境下の Natural グローバルスワッププールのインストール

グローバルスワッププールをインストールするには、以下の前提条件が満たされている必要があります。

グローバルスワッププールを使用する場合は、グローバルバッファプールも使用する必要があります。UTM 環境の Natural アプリケーションを再起動する前に、グローバルバッファプールを初期化する必要があります。つまり、少なくとも 1 人のユーザーが新しい Natural セッションを正常に開始してこのバッファプールを使用している必要があります。

異常終了した UTM 環境の Natural アプリケーションを再起動する前に、新しいグローバルバッファプールを開始する場合は、新しいグローバルスワッププールも開始する必要があります。ただし、新しいグローバルスワッププールを開始する場合は、新しいグローバルバッファプールも開始する必要はありません。

スワッププールとスワップファイルとの関係は次のとおりです。最初の UTM タスクでは新しく開始されたスワッププールを使用し、スワップファイルは OPEN 'OUTIN' で開かれるので、スワップファイルの内容は削除されます。後続の UTM タスクではすでに使用された（初期化された）スワッププールを使用し、スワップファイルは OPEN 'INOUT' で開かれるので、スワップファイルの内容は引き続き使用できます。

UTM 環境下の Natural グローバルスワッププールの開始

Natural グローバルスワッププールは、プログラム `CMPSTART` で開始する必要があります。これは、UTM 環境の最大 5 つの Natural アプリケーションから使用できます。

グローバルスワッププールに関する情報の表示

グローバルスワッププールの現在のパラメータ設定およびその開始日時に関する情報を取得するには、次の手順に従います。

以下のコンソールコマンドを発行します。

```
/INTRtsn,DPR
```


58 UTM 環境下のグローバルスワッププールの終了

- コンソールコマンドを使用した終了 364
- ダンプ付きの異常終了 364
- プログラムによる終了 365

このドキュメントでは、UTM 環境で Natural グローバルスワッププールを終了する方法について説明します。

以下のトピックについて説明します。

- ❗ **注意:** スワッププールを終了する前に、それを使用している UTM 環境の Natural アプリケーションを終了する必要があります。

コンソールコマンドを使用した終了

▶手順 58.1. グローバルスワッププールを正常に終了するには

- 1 以下のコンソールコマンドを発行します。

```
/INTRtsn,STOP
```

- 2 または、以下のコンソールコマンドを発行します。

```
/INTRtsn,END
```

ダンプ付きの異常終了

▶手順 58.2. グローバルスワッププールを異常終了してダンプを生成するには

- 以下のコンソールコマンドを発行します。

```
/INTRtsn,DUMP
```

スワッププールが異常終了し、ダンプが生成されます。

プログラムによる終了

▶手順 58.3. プログラム **CMPEND** を使用してグローバルスワッププールを正常に終了するには

- 以下のコマンドを発行します。

```
/SYSFILE SYSDTA=(SYSCMD)  
/EXEC (CMPEND,NAT230,MOD) name
```


59 Natural 3GL CALLNAT インターフェイス

このドキュメントでは、Naturalで3GLプログラムを使用してNaturalサブプログラムを呼び出して実行できるようにするNatural 3GL CALLNAT インターフェイスについて説明します。

- [Natural 3GL CALLNAT インターフェイス - 目的、前提条件、制限](#)
- [Natural 3GL CALLNAT インターフェイス - 使用方法、例](#)

60 Natural 3GL CALLNAT インターフェイス - 目的、前提条件、制限

▪ 3GL CALLNAT インターフェイスの目的	370
▪ 前提条件	371
▪ 制限	372

このドキュメントでは、3GL CALLNAT インターフェイスの目的、前提条件、および制限について説明します。

以下のトピックについて説明します。

3GL CALLNAT インターフェイスの目的

Natural は 3GL CALLNAT インターフェイスを使用することにより、3GL プログラムで Natural サブプログラムを呼び出して実行できるようになります。

3GL には、標準のリンケージコールインターフェイスをサポートする任意のプログラミング言語を使用できます。ほとんどの場合、3GL は COBOL プログラムですが、機能は PL/1、FORTRAN、C、アセンブラプログラムなどでも使用できます。

CMS 環境の Natural では、CALLNAT 機能を使用して Rexx プログラムの Natural サブプログラムを実行します。詳細については、「[VM/CMS 環境の Natural](#)」を参照してください。

可用性

このインターフェイスは、以下のオペレーティングシステム上でバッチモードで使用できます。

- z/OS
- z/VSE
- z/VM
- BS2000/OSD

また、以下の TP モニタ環境にも使用できます。

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

前提条件

このセクションでは、内部 CALLNAT ステートメントを使用して 3GL プログラムから Natural サブプログラムを実行するための前提条件について説明します。目的の機能を得るには、3GL プログラムから CALLNAT インターフェイスを実行する前に、Natural の環境を設定する必要があります。

スペース要件

Natural プログラムのパラメータアドレッシングのメカニズムでは、渡されるパラメータが Natural によって割り当てられたエリア、つまりそのサイズのいずれかに常駐している必要があります。しかし、3GL プログラムは、変数用のストレージをタスクのアドレススペース内のどこかに割り当てます。そこで、アドレッシングが有効に機能するように、まだ Natural エリアに存在していない変数に対して「値渡し」というメカニズムが使用されます。これは、Natural サブプログラムを呼び出す前に、渡されるパラメータが DATSIZE バッファという Natural エリアに転送されるということです。

変数の内容に使用されるストレージに加えて、パラメータ数に応じて追加のストレージが必要です。サブプログラム呼び出しプログラムが Natural でコーディングされている場合は、必要な合計スペース量は DATSIZE バッファで必要なスペースとほぼ同じです。

リンク

Natural サブプログラムを呼び出すには、3GL プログラムで CALLNAT インターフェイスを呼び出す必要があります。3GL プログラミング言語のコールインターフェイスの性能と機能に基づいて、CALLNAT インターフェイスは、ダイナミックロード用のアクセス可能なロードライブラリに配置されるか、3GL プログラムにリンクされます。ほとんどの場合は、3GL プログラムをインターフェイスモジュール (NATXCAL など。下記参照) にリンクする必要があります。

サンプルの XNATGC2 と XNATGCP2 ではそれぞれ、ダイナミックロードと、COBOL または PL/I からの CALLNAT インターフェイス呼び出しの手法を説明しています。

 **注意:** 環境に最適な方法については、担当のシステムプログラマに確認してください。

環境の依存関係

外部 3GL モジュールは、CSTATIC モジュールとして Natural にリンクした後でブランチとリンク命令を使用して呼び出すことも、ダイナミックにロードして TP 依存リンクメソッドを使用して呼び出すこともできます。

後者の場合は、3GL モジュールは TP 固有の方法で作成されるので、それに応じて CALLNAT インターフェイスを調整する必要があります。この目的のために、複数の TP 固有インターフェイスモジュールが用意されています。

NATXCAL	3GL モジュールを Natural にリンクするか、ダイナミックにロードしてから、ブランチおよびリンク命令を使用して呼び出す場合に使用します (バッチ、CMS、Com-plete、IMS/TM、TIAM、TSO、UTM)。
NATXCAL4	3GL モジュールを CALL ステートメントの INTERFACE4 オプションを使用して呼び出す場合に使用します。INTERFACE4 Natural Callnat インターフェイスと INTERFACE4 コールバック機能を提供します。INTERFACE4 の機能の詳細については、CALL ステートメントのドキュメントを参照してください。
NCIXCALL	CICS 環境で、3GL モジュールが EXEC CICS LINK を使用して呼び出し済みの場合に使用します。NCIXCALL は CICS マクロでコンパイルされるソースコードで提供されます。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」も参照してください。
NCIXCPRM	CICS 環境で、後続の EXEC CICS LINK コマンドの COMMAREA として使用されるパラメータアドレスリストを作成するために使用します。

制限

Natural サブプログラムの終了

呼び出された Natural サブプログラムは、呼び出し元プログラムに戻して終了する必要があります。

使用できない Natural ステートメント

以下のステートメントは使用できません。

- FETCH
- RUN
- STOP
- TERMINATE

呼び出された Natural サブプログラムでこれらのステートメントが使用されていると、該当する Natural ランタイムエラー (NAT0967) が発生します。

3GL プログラムから渡されるパラメータ値

3GL プログラムから渡されるパラメータ値は、書き込み禁止ストレージエリアに配置することはできません。

ダイナミック配列

ダイナミック範囲の配列は使用できません。

TP モニタ固有の制限

■ CICS 環境

CICS 環境では、Natural 3GL CALLNAT インターフェイスを使用する 3GL プログラムは会話型モード用に作成する必要があります。3GL プログラムは第 2 CICS プログラムレベルで実行されるので、擬似会話型プログラム手法は使用できません。

■ IMS/TM および UTM 環境

IMS/TM および UTM 環境で実行される Natural で 3GL CALLNAT インターフェイスを使用できるのは、3GL プログラムと Natural サブプログラムのどちらも端末 I/O を発行していない場合のみです。呼び出された Natural サブプログラムで DISPLAY、INPUT、および WRITE が使用されていると、該当する Natural ランタイムエラー (NAT0967) が発生します。

61 Natural 3GL CALLNAT インターフェイス-使用方法、 例

▪ 使用方法	376
▪ 環境の例	379

このセクションでは、3GL CALLNAT インターフェイスの使用方法を説明し、3GL CALLNAT 環境の例をいくつか示します。

以下のトピックについて説明します。

使用方法

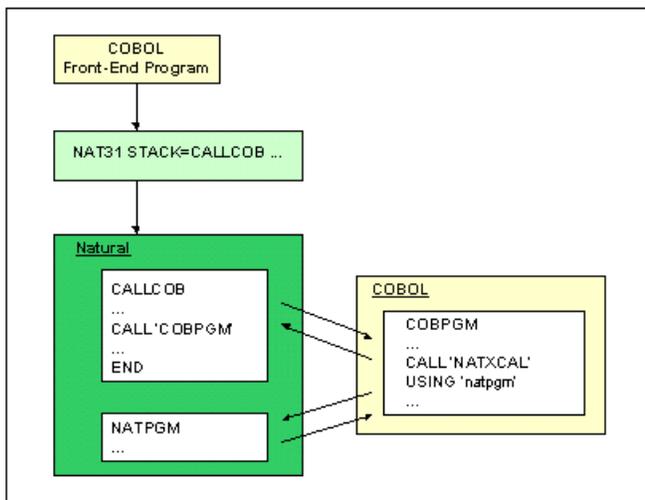
以下のトピックについて説明します。

- 概要
- 呼び出しの構造
- パラメータ処理

概要

Natural サブプログラムを 3GL プログラムから呼び出すときは、Natural セッションがアクティブである、つまり 3GL プログラム自体が Natural によって呼び出される必要があります。

そのため、Natural 層を表示したくない場合は、特別な予防措置を取る必要があります。以下の図に、このような場合に、Natural3GLCALLNAT インターフェイスを使用するアプリケーションをどのように設計するかについて概要を示します。



Natural 起動プログラムの最初の呼び出しによって、必要な環境が設定されます。その後で、この起動プログラムは Natural CALL ステートメントを使用して、CALLNAT インターフェイスを呼び出す場所から 3GL プログラムを呼び出すことができます。

呼び出しの構造

Natural のメインプログラムは極めて単純で、COBOL などのプログラムを呼び出すだけです。

```
.....
CALL 'cobpgm'
END
.....
```

3GL プログラミング言語（COBOL など）の CALL ステートメントは、Natural 3GL CALLNAT インターフェイスにアクセスできる必要があります、それによって Natural サブプログラムを呼び出します。

```
.....
CALL 'interface' USING natpgm p1 ... pn
.....
```

パラメータ *interface* は環境依存（NATXCAL など）で、呼び出し元プログラムにリンクされます。パラメータ *natpgm* は、8 バイトの英数字変数で、呼び出される Natural サブプログラムの名前を含む必要があります。パラメータ *p1 ... pn* は Natural サブプログラムに渡されます。

例（CICS を除くすべての環境）：

COBOL プログラム *cobpgm* に、以下のようなコーディングを含めることができます。

```
.....
MOVE 'FINDNPGM' TO natpgm
CALL 'interface' USING natpgm number name
IF natpgm NE 'FINDNPGM'
THEN GOTO error_handling_1
.....
```

呼び出される Natural サブプログラム *natpgm* では、ファイル EMPLOYEES 内で、COBOL プログラムから渡された値と同じ *name* の人の数を計算します。

```
DEFINE DATA
PARAMETER
1 pnumber (P10)
1 pname (A20)
LOCAL
1 emp VIEW OF employees
END-DEFINE
*
RESET presp
FIND NUMBER emp WITH name=pname
MOVE *NUMBER TO pnumber
ESCAPE ROUTINE
```

サブプログラムの実行中にエラーが発生した場合は、このエラーに関する情報が変数 *natpgm* に *NAT $nnnn$ フォーマットで返されます。*nnnn* は対応する Natural エラー番号です。

例 (CICS のみ) :

CICS 環境では、COBOL などからの Natural サブルーチンの呼び出しは、以下のように指定する必要があります。

```
...  
WORKING STORAGE SECTION  
...  
01 PARM-LIST PIC X(132).  
01 NATPGM PIC X(8).  
01 NUMBER PIC 9(10) comp-3.  
01 NAME PIC X(20).  
...  
PROCEDURE DIVISION  
...  
MOVE 'FINDNPGM' TO NATPGM  
CALL 'NCIXCPRM' USING PARM-LIST NATPGM NUMBER NAME ...  
EXEC CICS LINK PROGRAM('NCIXCALL')  
COMMAREA(PARM-LIST) LENGTH(132) END-EXEC.  
...
```

呼び出されるサブルーチン NCIXCPRM では、後続の EXEC CICS LINK コマンドで COMMAREA として使用されるパラメータアドレスリストを作成します。

パラメータ処理

フォーマットおよび長さの確認は行いません。正しいパラメータリストを渡すのは呼び出し元の責任です。パラメータの数値、フォーマット、および長さは、呼び出される Natural サブプログラムによって定義されます。

パラメータを渡す場合は、グループ配列を渡してはなりません。グループ配列は個別の配列として解決されるからです。

無効な構文の例:

```
.....  
01 GROUP (1:2)  
02 F1  
02 F2  
.....  
.....  
CALL ..... F1 F2  
.....
```

有効な構文の例：

```
.....
01 F1 (1:2)
01 F2 (1:2)
....
.....
CALL ..... F1 F2
.....
```

ダイナミック範囲の配列は使用できません。

環境の例

以下に示す 3GL CALLNAT 環境の例では、特定の TP モニタシステムまたはバッチモード環境で COBOL ルーチンから Natural サブプログラムを呼び出す方法を説明し、このような環境を作成するためのシステム固有の手順を示します。

以下のトピックについて説明します。

- [CICS の環境例](#)
- [その他のサンプル](#)
- [その他のサポートされている環境用のサンプル](#)

CICS の環境例

CICS でサンプル Natural 3GL CALLNAT 環境を作成するには、次の手順に従います。

手順 1：環境の初期化の作成

- 3GL CALLNAT 環境を初期化するフロントエンドプログラムをセットアップします。
- Natural/CICS ソースライブラリの COBOL フロントエンド XNCIFRCX を使用します。これは、Natural を起動し、LOGON YOURLIB をスタックし、Natural 3GL CALLNAT 環境を初期化する TSTCOB プログラムを実行します。
- ソースコードで文字列 NCvr を見つけ、Natural 用の有効なトランザクション ID と置き換えます。
- COBOL プログラムをコンパイルおよびリンクエディットし、CEDA DEFINE PROGRAM を使用してプログラムを CICS に定義します。

手順 2：サンプル COBOL 呼び出しのインストール

Natural/CICS ソースライブラリ NCI.SRCE にサンプルメンバ XNCI3GC1 がある場合は、その中に Natural サブプログラム MYPROG へのデフォルト呼び出しが含まれています。

- テストに使用するために、SYSTEM ライブラリに以下のプログラムを作成し格納します。

```
WRITE 'BEFORE PGM EXECUTION'
CALL 'COBNAT'
WRITE 'AFTER PGM EXECUTION'
END
```

- XNCI3GC1 ソースを調べ、CALL と LINK を確認します。以下の CICS INCLUDE 指示を使用して COBNAT としてコンパイルおよびリンクするか、ジョブ例 NCTI070 の手順 2 を使用します。

```
INCLUDE CICSLIB(DFHECI)
INCLUDE XNCI3GC1           <= output from translator and compiler
INCLUDE NCILIB(NCIXCPRM)
ENTRY XNCI3GC1
NAME COBNAT(R)
```

手順 3：サンプル Natural サブプログラムの作成

デフォルトでは、ソースメンバ XNCI3GC1 は YOURLIB ライブラリの Natural サブプログラム MYPROG を呼び出すように設定されています。前述したように、TSTCOB は Natural サブプログラム MYPROG への実際の呼び出しを含む COBNAT を呼び出すことによってプロセスを起動します。

- Natural サブプログラムの実行をデモンストレーションするサブプログラム MYPROG を作成します。

```
DEFINE DATA PARAMETER
  01 PARM1 (A18)
  01 PARM2 (A18)
  01 PARM3 (A18)
END-DEFINE
*
  MOVE 'PARAM01' TO PARM1
  MOVE 'PARAM02' TO PARM2
  MOVE 'PARAM03' TO PARM3
END
```

手順 4 : CICS リソースの確認

- CICS リソース (PPT および PCT) を定義するためのガイドとして、ジョブ NCII005 を使用します。
- 必要な CICS リソース (PPT および PCT) を定義します。

手順 5 : 環境のテスト

NCYC デフォルトトランザクションを使用して環境をテストします。CEDF を使用して、プログラム制御をモニタし、使用中のデータエリアをモニタします。

 **重要:** Natural は CICS プログラム階層の最上位にあるため、端末 I/O を発行する COBOL サブプログラムは会話型モードで実行する必要があります。擬似会話型のプログラムは変更する必要があります。Natural 3GL CALLNAT インターフェイスを使用した新しい開発は会話型モードで行う必要があります。

その他のサンプル

XNCI3GC2	XNCI3GC1 と同じ機能の COBOL サンプルですが、呼び出し元の Natural プログラムからのパラメータを受け入れます。
XNCI3GP1	COBOL サンプル XNCI3GC1 と同じ機能の PL/I サンプルです。
XNCI3GP2	XNCI3GC1 と同じ機能の PL/I サンプルですが、呼び出し元の Natural プログラムからのパラメータを受け入れます。

その他の非 CICS サンプル

XNAT3GC2	CICS サンプル XNCI3GC2 と同じ機能の COBOL サンプルです。
XNAT3GP2	CICS サンプル XNCI3GP2 と同じ機能の PL/I サンプルです。

その他のサポートされている環境用のサンプル

サンプル Natural 3GL CALLNAT を作成するには、次の手順に従います。

手順 1 : ASMNAT のアセンブルとリンク

サンプルアセンブラルーチン XNAT3GA1 には、CALLNAT インターフェイスにアクセスするための基本例が含まれています。このプログラムのソースには、レジスタ呼び出し規則が含まれています。

NATXCAL と XNAT3GA1 をエン트리ポイント ASMNAT でリンクします。

手順 2 : Natural セッションの開始

実際に Natural サブルーチン ASMNAT を呼び出す ASMNAT プログラムを呼び出すプログラムをスタックして Natural セッションを開始します。

62 Software AG Editor の使用

このドキュメントでは、Software AG Editor の操作方法について説明します。

Software AG Editor は、Natural に含まれている基本機能の代表的機能で、いくつかの Natural サブ製品やその他の Software AG 製品でのみ使用します。

- [エディタワークファイル](#)
- [エディタバッファプール](#)

以下の項目も参照してください。

- [SYSEDIT ユーティリティ](#) (Natural の『[ユーティリティ](#)』ドキュメント)
- [Software AG Editor のインストール](#) (Natural の『[インストール](#)』ドキュメント)
- [Software AG Editor](#) (Natural の『[エディタ](#)』ドキュメント)

63 エディタワークファイル

■ エディタワークファイルの構造	386
■ z/OS、z/VSE、および BS2000/OSD でのエディタワークファイル	387
■ バッチフォーマットユーティリティの使用	388
■ 初期化中のフォーマット	388
■ z/OS および z/VSE でのエディタワークファイルの管理	388
■ BS2000/OSD でのエディタワークファイルの管理	389
■ VM/CMS でのエディタワークファイル	390
■ Complete/SMARTS でのエディタワークファイル	390

このドキュメントでは、各種オペレーティングシステムでのエディタワークファイルの構造、使用、および管理について説明します。以下のトピックについて説明します。

以下の項目も参照してください。

- *SYSEDT* ユーティリティ (Natural の『ユーティリティ』ドキュメント)
- *Software AG Editor* のインストール (Natural の『インストール』ドキュメント)
- *EDBP - Software AG エディタバッファプール定義* (Natural の『パラメータリファレンス』)
- *Software AG Editor* (Natural の『エディタ』ドキュメント)

エディタワークファイルの構造

エディタワークファイルは、固定長レコードの相対レコードデータセットです。以下の3つの部分に分かれています。

- 制御レコード
- ワークレコード
- リカバリレコード

 **注意:** プロファイルパラメータ *EDPSIZE* で定義されるエディタ補助バッファプールを使用する場合は、エディタワークファイルは不要です。

制御レコード

制御レコードには、バッファプールパラメータを含むバッファプール制御情報が含まれています。

ワークファイルの初回初期化時、またはバッファプールコールドスタート (エディタバッファプールパラメータ *COLD* によりトリガ) 時に、エディタバッファプールのサブパラメータ *EDBP* や対応するマクロ *NTEDBP* で定義された値がワークファイル制御レコードに保存されます。

制御レコードを変更するには、*SYSEDT* ユーティリティの生成パラメータ機能を使用します。

バッファプールウォームリスタート時に、パラメータが制御レコードから読み込まれます。

ワークレコード

ワークレコードには、フリーのバッファプールブロックが不足したためにバッファプールから移動した論理ファイルレコードが含まれています。

論理ワークファイルレコードは、バッファプールの再起動時または論理ファイルのタイムアウト時に失われます。

リカバリレコード

リカバリレコードには、エディタセッションのチェックポイント情報が保持されます。システムが異常終了した場合に、エディタリカバリ機能がこの情報を使用して論理ファイルを回復します。リカバリレコードは、バッファプールのコールドリスタート時に失われます。

リカバリ機能は、Natural ISPF でのみ使用されます。この製品を使用しない場合は、エディタバッファプールサブパラメータ PWORK=100 を定義することで、リカバリ部分なしで実行できます。

z/OS、z/VSE、および BS2000/OSD でのエディタワークファイル

1つのエディタワークファイルは1つのエディタバッファプールに対応します。グローバルエディタバッファプールを使用する場合は、そのグローバルエディタバッファプールを使用するすべてのユーザー間でエディタワークファイルを共有する必要があります。

エディタワークファイルは、すべてのユーザーのエディタセッションを収容できる大きさである必要があります。エディタユーザーごとに最低100レコード分を用意することをお勧めします。ワークファイルのレコード長は、必ず固定で、504~16384バイトで定義でき、8の倍数にする必要があります。

 **注意:** Natural ISPF で編集されるデータセットまたは PDS メンバのレコード長は、このエディタワークファイルのレコード長を超えることはできません。

ワークファイルレコードのサイズは、z/OS および z/VSE ではエディタワークファイルの割り当て時に指定され（デフォルトサイズは4088）、BS2000/OSD ではバッファプールパラメータマクロで定義することにより指定されます（デフォルトサイズは4096）。

エディタワークファイルレコードの総数は、エディタワークファイルに割り当てられたデータセットのスペースによって異なります。

エディタワークファイルのフォーマットには、以下の2通りの方法があります。

- バッチフォーマットユーティリティを使用してオフラインで実行する
- バッファプールの初期化中にオンラインで実行する。

バッチフォーマットユーティリティの使用

この方法ではオンラインユーザーがフォーマットの終了を待つ必要がないため、この方法をお勧めします。必要に応じて、Natural パラメータモジュールをアセンブルしバッチフォーマットユーティリティにリンクして、NTEDBP マクロを使用してエディタバッファプールパラメータを指定できます。指定しない場合は、デフォルトのパラメータ値が適用されます。

ただし、再フォーマット中は、ワークファイルを使用しないでください。対応するバッファプールを使用しているシステムは再フォーマットの前に終了しています。

初期化中のフォーマット

エディタバッファプールが非初期化状態または終了状態で、以下のいずれかに当てはまる場合は、Software AG Editor を初めて使用するセッション中に "バッファプールコールドスタート" が実行されます。

1. ワークファイルがフォーマットされていない。
2. 「コールドスタート」が制御レコードで指定されている（またはエディタバッファプールサービスユーティリティ SYSEDT を使用して指定されている）。
3. バッファプールサブパラメータ COLD=ON が指定されている。

それ以外の場合は、バッファプールの初期化中に有効な制御レコードが見つかったら、バッファプールウォームスタートが実行されます。この場合は、バッファプールパラメータはすべてワークファイル制御レコードから取得され、レコードはフォーマットされません。

z/OS および z/VSE でのエディタワークファイルの管理

エディタワークファイルのサイズを変更する際は（例えば、サイズが小さすぎる場合）、エディタワークファイルバッチユーティリティの COPY 機能を使用することで、バッファプールコールドスタートを回避できます。コールドスタートが実行されると、リカバリーレコードが失われます。

既存のエディタワークファイルをコピーするには、次の手順に従います。

1. SYSEDT ユーティリティを使用して、バッファプールパラメータを修正します。例えば、ファイル内のワークレコードの比率を変更する場合は、PWORK を使用します。
2. SYSEDT ユーティリティのシステム管理機能を使用して、エディタバッファプールを終了します。バッファプールが終了したら、エディタを使用している Natural セッションがないことを確認します。

3. エディタワークファイルを閉じ（必要な場合）、割り当てを解除します。
4. VSAM ユーティリティ IDCAMS (ALTER コマンド) を使用して、エディタワークファイルの名前を変更します。
5. 新しいエディタワークファイルを、オリジナルの名前で定義します。サイズは必要に応じて変更できますが、レコード長は同じにします。
6. 以下を追加してから、新しいファイルを使用してエディタワークファイルバッチユーティリティを実行します。

```
PARM=COPY
```

上記を、EXEC JCL カードに追加します。

```
//CMCOPY DD...    under z/OS or  
//DLBL CMCOPY...  under z/VSE
```

上記を、新しいワークファイル CMEDIT にコピーされる名前変更したエディタワークファイル CMCOPY のカードに追加します。

7. 潜在的なエラーがないかジョブログを確認します。
8. エディタワークファイルを再配置し、（必要な場合）開きます。
9. SYSEDT ユーティリティ (Natural の『ユーティリティ』ドキュメント参照) を使用して、バッファプールとワークファイルが正常に再起動したかどうかを確認します。

 **重要:** バッファプールの再起動後に Natural セッションでエディタを使用する場合は、すべての Natural セッションを再スタートする必要があります。

BS2000/OSD でのエディタワークファイルの管理

エディタワークファイルのサイズを変更する場合は、次の手順に従って、新しいエディタワークファイルをフォーマットし、リカバリレコードを古いワークファイルから新しいワークファイルにコピーします。

1. エディタを使用するシステムをすべてシャットダウンします。
2. エディタバッファプールを終了します。
3. 現在のエディタワークファイルの名前を変更します。
4. 新しいエディタワークファイルを、オリジナルの名前で作成します。
5. エディタワークファイルフォーマットプログラム (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」も参照) を、FORMAT 機能の代わりに COPY 機能を使用して実行します。

例：

```
/CAT NATEDT.WORKFILE,NATEDT.COPYFILE
/FILE NATEDT.WORKFILE,LINK=CMEDIT,SPACE=nnn
/LOGON
/FILE NATEDT.COPYFILE,LINK=CMCOPY
/FILE NATEDT.WORKFILE,LINK=CMEDIT
/SYSFILE SYSLST=LST.NATEDFM2
/SYSFILE SYSDDTA=(SYSCMD)
/EXEC (NATEDFM2,NATvrs.MOD)
COPY
/LOGOFF N
```

VM/CMS でのエディタワークファイル

1つのエディタワークファイルは1人のVM/CMSユーザーに対応します。ワークファイルはエディタの初回初期化時に作成され、デフォルト名はCMEDIT DATA A1です。

エディタワークファイルは、VM/CMSユーザーのエディタセッションを収容できる大きさである必要があります。名前、サイズ、およびワークファイルレコード数は、Naturalパラメータモジュール [NATPARM](#) のエディタバッファプールパラメータ EDBP またはマクロ NTEDBP で指定されます。最適なパフォーマンスを得るために、800の倍数にすることをお勧めします。

Complete/SMARTS でのエディタワークファイル

SMARTS ワークファイルは、SMARTS ポータブルファイルシステム内にあります。パスは SMARTS 環境変数 \$NAT_WORK_ROOT で指定する必要があります。エディタワークファイルの名前は EDBP サブパラメータ DDNAME で指定する必要があります。

エディタワークファイルのフォーマットは、バッファプールの初期化中（オンライン）に限り可能です。現在、SMARTS 環境でエディタワークファイルをオフラインでフォーマットするツールはありません。

64 エディタバッファプール

■ エディタバッファプールの目的	392
■ フリーブロックの取得	393
■ エディタバッファプールの初期化	393
■ エディタバッファプールの再起動	394
■ エディタバッファプールパラメータ	394
■ マルチユーザー環境でのバッファプールの初期化	394

このドキュメントでは、Software AG Editor で使用する中間メインストレージエリアであるエディタバッファプールの目的、使用、および操作について説明します。

以下のトピックについて説明します。

エディタバッファプールの目的

エディタバッファプールは、エディタバッファ (SSIZE) の拡張と見なすことができます。これは、Software AG Editor で論理ファイルを保持するために使用する中間メインストレージエリアです。

論理ファイルは 1 つまたは複数の論理レコードで構成され、エディタによって保持されるオブジェクトのデータ (ファイルメンバなど) を含みます。ユーザーは複数のオブジェクトを同時に使用することができ、ユーザーごとに複数の論理ファイルを同時に存在させることができます。

論理ファイル数は (エディタワークファイルのリカバリレコードの比率と同じように) バッファプールパラメータマクロで定義されます。

エディタバッファプールは、ローカルバッファプールやグローバルバッファプール (z/OS および BS2000/OSD のみ) または補助バッファプール (EDPSIZE) として定義できます。マルチユーザー環境 (CICS、IMS/TM、UTM) の場合は、エディタバッファプールは同じリージョン (ローカルプール) または複数リージョン (グローバルプール) のどちらかのすべてのエディタユーザーによって共有されます。CMS 環境では、バッファプールは常にローカルプールです。グローバルバッファプールは、Com-plete 環境では SD エディタワークファイルが異なるため、Com-plete とその他のリージョンとの間では共有できません。

エディタバッファプールには、以下のさまざまな制御テーブルと多数のデータブロックが含まれます。

エリア	サイズ
メインコントロールブロック	500 バイト
論理ファイルテーブル	論理ファイル当たり 20 バイト
ワークファイルテーブル	レコード当たり 4 バイト
リカバリファイルテーブル	レコード当たり 16 バイト
バッファプールブロックテーブル	ブロック当たり 28 バイト
バッファプールブロック	下記の本文参照

バッファプールブロックのサイズはワークファイルレコードのサイズと同じで、1 つのバッファプールブロックに 1 つの論理ファイルレコードを含めることができます。

バッファプールはエディタの最初のユーザーによって初期化されます。ウォームスタートでのバッファプールの初期化中に、すべてのリカバリレコードがチェックされてリカバリファイルテーブルが構築されます。

バッファプールにアクセスする機能は複数あります（例えば、レコードの割り当て、読み取り、書き込み、または削除の各機能）。

フリーブロックの取得

バッファプールがいっぱいになったら、外部データセットであるエディタワークファイルにバッファプールブロックを移動して、フリーブロックを取得する必要があります。

このような場合は、エディタによってすべての論理ファイルのタイムアウト値がチェックされ、指定時間内にアクセスされていない論理ファイルが削除されます。これにより、そのバッファプールブロックとワークファイルレコードがすべて解放され、論理ファイルが失われます。

それでも使用できるバッファプールブロックがまだない場合は、指定のタイムアウトパラメータ値（Natural の『ユーティリティ』ドキュメントの「SYSEDT ユーティリティ - エディタバッファプールサービス」の「生成パラメータ」を参照）に従って、最も古いブロックがワークファイルに移動されます。

エディタバッファプールの初期化

初期化されていないエディタバッファプールは、Software AG Editor が初めて呼び出されたときに初期化されます。その後、さまざまなコントロールブロックが作成されます。バッファプールとワークファイルの初期化には2つモードがあります。「コールドスタート」と「ウォームスタート」です。

バッファプールコールドスタート

バッファプールコールドスタートは、エディタバッファプールサブパラメータ COLD またはエディタバッファプールサービスユーティリティ SYSEDT によってトリガされるか、自動的にトリガされます（エディタワークファイルが未フォーマットの場合）。

バッファプールコールドスタートでは、エディタバッファプールパラメータ EDBP または対応するマクロ NTEDBP の値がワークファイル制御レコードに保存され、すべてのワークファイルリカバリレコードがクリアされます。

バッファプールウォームスタート

バッファプールウォームスタートでは、バッファプールパラメータがワークファイル制御レコードから読み込まれ、すべてのワークファイルリカバリレコードが読み取られて、バッファプールにリカバリファイルテーブルが構築されます。

エディタバッファプールの再起動

SYSEDT ユーティリティを使用してエディタバッファプールを終了することで、エディタバッファプールを非初期化状態にできます。これにより、TP システムやグローバルバッファプールの再起動を回避できます。

バッファプールに問題があって SYSEDT を使用できない場合は、プログラム BPTERM を使用してバッファプールを終了できます。

 **重要:** バッファプールの再起動後に Natural セッションでエディタを使用する場合は、すべての Natural セッションを復元する必要があります。

エディタバッファプールパラメータ

エディタバッファプールを操作するパラメータを定義するには、エディタバッファプールパラメータ EDBP、または Natural パラメータモジュール NATPARM の対応するマクロ NTEDBP が必要です。

エディタワークファイルがフォーマットされると、これらのパラメータはワークファイル制御レコードに保存され、その他のレコードはすべてクリアされます。したがって、以前使用したワークファイルを再フォーマットすると、すべてのエディタチェックポイントとリカバリ情報が失われます。

これらのパラメータの一部は、エディタバッファプールサービスユーティリティ SYSEDT を使用して、バッファプールの実行中にダイナミックに変更できます。

マルチユーザー環境でのバッファプールの初期化

バッファプールの初期化中に、すべてのリカバリレコードがエディタワークファイルから読み込まれます。エディタバッファプールの初期化が完了するまで、最初のユーザーは長時間待たねばならず、タイムアウトメッセージが表示されることもあります。

このため、最初のユーザーがアクティブになる前にバッファプールの初期化をトリガする特別な Natural プログラムが用意されています。このプログラムは、TP モニタの起動中に有効にするか、グローバルバッファプールの使用時にバッチジョブで有効にすることができます。

その後、以下のセッションパラメータを使用してセッションを開始する必要があります。

```
STACK=(LOGON SYSEDT, user, password;BPINIT;FIN)
```

CICS環境の場合：セッションを非同期的に実行している場合は、初期化中に出力されるエラーメッセージを取得するために `SENDER=CONSOLE` を指定する必要があります。ソースプログラム `FRONTPLT` が、`PLTPI` 経由で **CICS** 起動中に非同期 Natural セッションを開始する方法を示すサンプルプログラムとして提供されています。

65

Natural Net Data インターフェイス NATNETTO

■ Natural Net Data ドライバの機能説明	398
■ 全体的なメッセージレイアウト	399
■ ヘッダーのレイアウト	399
■ フォーマットバッファのレイアウト	403
■ バリュースタックのレイアウト	408
■ 属性バッファ	409

このドキュメントでは、Natural Net Data インターフェイスとネットデータプロトコル定義について説明します。

『*Installation*』ドキュメントの「Natural Net Data インターフェイスのインストール」も参照してください。

Natural Net Data ドライバの機能説明

Natural Net Data ドライバ NATNETTO は、EntireX CICS 3270 Bridge や、メッセージ指向のサーバー環境（つまり TP モニタ）での類似のクライアント/サーバーソリューションをサポートするために導入されたコンポーネントです。

NATNETTO が実装しているプロトコルドライバは、ネットデータプロトコルを使用して、Natural（レガシー）アプリケーションによるクライアントアプリケーションからのプログラム間通信を可能にします。代表的な例は、CICS、IMS/TM、UTM などの TP モニタ環境で実行される Natural アプリケーションにアクセスするデスクトップクライアント（Natural for Windows または VBA など）で構築）です。

"ネットデータ"とは、プロトコルにテキスト定数などのフォーマットデータやデバイス依存の制御シーケンスなどが一切含まれていないという意味です。すべてのデータは出力可能なフォーマットで通信されます。これは、結局、非英数字フィールドに必要なマーシャリングとアンマーシャリングはクライアントが行う必要があることを暗黙に示します。

基本的に、プロトコルは以下の2つの部分で構成されます。

- ヘッダーまたはコントロールブロック、および未加工のネットデータを含むバリュースタック。この部分は必須です。ヘッダーには、制御、環境、およびセッション情報が含まれ、データスタックの他の部分へのポインタが保持されます。バリュースタックには、クライアントとサーバーの間で交換される実際のネットデータが含まれます。
- オプションの変動部分。フォーマットスタックや属性スタックがあります。オプションのフォーマットスタックには、バリュースタックのフィールドごとに1エントリの記述データが含まれます。属性スタックには、バリュースタックのフィールドごとに1バイトのプリセット値0が含まれます。該当するオプションが設定されている場合は、クライアントは変更された各フィールドのこの値を1に切り替える必要があり、このようにして mdt ビットの設定をエミュレートします。

ヘッダー、バリュースタック、および属性スタックは、送受信メッセージに含まれます。フォーマットスタックは、送信メッセージにのみ含まれます。ヘッダーは、クライアントがフロー制御のためにミラー化する必要があるトランザクション番号を保持します。ほとんどのレガシーアプリケーションはブロックモードの端末から起動するように設計されているので、プロトコルは PF キーやカーソル位置など 3270 に類似の機能をサポートします。

全体的なメッセージレイアウト

全体的なメッセージレイアウトの以下の部分は必須です。

- ヘッダー（以下の表の最初の 2 行）
- バリュースタック

以下の部分はオプションです。

- フォーマットバッファ
- 属性バッファ

"FSCB"	バリュースタックオフセット	フォーマットバッファオフセット	...
	属性バッファオフセット	補助文字	カーソル位置 ...
バリュースタック			
フォーマットバッファ			
属性バッファ			

レイアウトの各部分の詳細については、表 1 を参照してください。

ヘッダーのレイアウト

表 1: コントロールブロック - 固定部分

フィールド	フォーマット	範囲	意味
Eyecatcher	A4	FSCB	アイキャッチャー
Product code	A3	-	製品 ID
Protocol version	N2	01 - 99	特定の製品のバージョン
Value buffer offset	N10	算出	メッセージの先頭からのバリュースタックのオフセット
Format buffer offset	N10	算出	メッセージの先頭からのフォーマットバッファのオフセット
Total message length	N10	算出	すべてのバッファを累積した長さ
Message number	N6	呼び出しごとに 1 ずつ増加	通信相手によってエコーされます。
Block number	N5	01~99（通常は 01）	1 つのメッセージ内の分割ブロック用
Number of parameters	N5	算出	VB 内のパラメータ数

フィールド	フォーマット	範囲	意味
Session token	A32		セキュリティトークン
Message format	A1	表 2 を参照	バリュースタック内のフィールド区切りのモード
Delimiter character	A1	-	
Architecture	A2	表 3 を参照	送信相手のアーキテクチャ
Call type	A2	表 4 を参照	現在のコールのタイプ
Response code	N4	0001 - 9999	クライアントからのレスポンスコード
Block status	A1	L または N	ブロックは最後の msg または次の msg
Server name/TAC	A8	-	TP トランザクションコードまたはサーバーの名前
Aid character	A2	表 5 を参照	クライアントで押されたか生成された補助文字
Cursor line	N3	1-クライアント上の最大物理行	カーソル行または 000 *
Cursor column	N3	1-クライアント上の最大物理列	カーソル列またはカーソルフィールド番号 *
Attribute buffer offset	N10	算出	メッセージの先頭からの属性バッファのオフセット
Timestamp	A16	一般	クロック値（マップ格納時間、16進数、出力可能）を保存
DBID	N5	1 - 32767	サーバー上の FNAT の DBID
File number	N5	1 - 32767	サーバー上の FNAT のファイル番号
Date form	A1	I、G、E、U	Natural に従った日付フォーマット
Decimal character	A1	-	サーバー上の Natural デリミタ文字
Input delimiter char.	A1	-	Natural INPUT 区切り文字（サーバー）
Control character	A1	-	Natural 制御文字（サーバー）
Language code	N2	01 - 99	Natural 言語コード（サーバー）
Application ID	A8	-	Natural アプリケーション ID
Program name/map	A8	-	実行中のプログラム/マップまたはフォーマット名
Error number	N5	00001 - 99999	Natural エラー番号
Line number	N4	0001 - 9999	現在の I/O ステートメントの行番号
Error state	A1	-	ステータスバイト
Error program	A8	-	エラーの原因となったオブジェクト
Error level	N2	01 - 15	エラーオブジェクトのサブルーチンレベル
Message type	A1	表 6 を参照	メッセージのタイプ
Option flag 1	A1	表 7 を参照	コントロールフラグ

フィールド	フォーマット	範囲	意味
Option flag 2	A1	表7を参照	コントロールフラグ
Option flag 3	A1	表7を参照	コントロールフラグ
Option flag 4	A1	表7を参照	コントロールフラグ
Option flag 5	A1	表7を参照	コントロールフラグ
Option flag 6	A1	表7を参照	コントロールフラグ
Option flag 7	A1	表7を参照	コントロールフラグ
Option flag 8	A1	表7を参照	コントロールフラグ

*NATCONFIGにカーソルフィールド番号表記が設定されている場合は、カーソル行は常に000で、カーソル列はフィールドの絶対番号を含み、ここにカーソルが配置される（送信）か送信時に配置済み（受信）です。

 **注意:** 現在、すべてのヘッダーフィールドが使用されているわけではありません。

表2：フィールド区切りのモード

フォーマット A1

値	意味
D	デリミタモード
F	固定フォーマットモード
L	フィールドの前に長さフィールド (N3) を配置

表3：送信相手のアーキテクチャ

フォーマット A2 のマスク

値	意味
- 1	下位バイト 1 番目のマスク (Vax)
- 2	未使用
- 4	EBCDIC アーキテクチャのマスク
- 8	ASCII 8 アーキテクチャのマスク
1 -	浮動小数点表記 VAX のマスク
2 -	浮動小数点表記 IEEE のマスク

表 4：コールタイプ

フォーマット A2

通信のタイプ	値	意味
Natural ネットデータ/3GL	MD	マップデータ (フォーマットを使用したネットデータ)
	ND	ネットデータ
	CM	コマンドモード (サーバー)
	FD	マップフォーマットダウンロード
	IP	通常の入力ステートメント
	CS	セッションクローズ終了メッセージ

表 5：補助文字表

フォーマット A2

補助文字	PF キー
EN	Enter
CL	Clear
P1	PA1
P2	PA2
P3	PA3
01	PF1
02	PF2
03	PF3
...	...
47	PF47
48	PF48
CS	Close Session



注意: CS (Close Session) を使用すると、クライアントはサーバーセッションを直ちに強制的に閉じることができます。したがって、これは実際には PF キーではなく、サーバーのコマンドコードです。

表 6：メッセージタイプ

値	意味
D	ダイアログメッセージ
A	非同期メッセージ
P	出力メッセージ

表 7：Natural ネットデータ通信のオプションフラグ

すべてのフラグはフォーマット A1 です。

フラグ	値	意味
Option 1	F	メッセージにはフォーマットバッファが含まれます（フォーマットバッファオプション）。
Option 2	S	ネットデータは画面バッファから生成されます。
	P	ネットデータはページバッファから生成されます。
Option 3	A	メッセージには属性バッファが含まれます（属性バッファオプション）。
Option 4	P	VB 内のデータは表示フォーマット（出力可能）です。
	I	VB 内のデータは送信元の内部フォーマットです。
	A	データは英数字に変換された内部フォーマットです。
Option 5	M	送信メッセージにはオーバーレイ部分が含まれます。
Option 6	1	拡張フォーマットバッファオプション 1。
	2	拡張フォーマットバッファオプション 2。
Option 7		将来的に使用される予定。
Option 8		将来的に使用される予定。

フォーマットバッファのレイアウト

基本部分

フォーマットバッファの各エントリは、以下の 4 つの要素から成る可変長の文字列です。

- 識別子
- 保護インジケータ
- フォーマットインジケータ
- 出力可能フィールド長

表 8：フォーマットバッファエントリ

要素	値	意味
識別子	F	フィールド
	S	サブフィールド
保護	M	変更可能フィールド
	O	出力専用フィールド、保護されたフィールド
フォーマット	A	アルファベット
	N	数値
フィールド長	L~LLL,L	Natural 標準に従った長さ指定

例：

FMA20 フィールド、変更可能、フォーマットはアルファベット、フィールド長は 20
 SMN12,4 サブフィールド、変更可能、フォーマットは数値、フィールド長は 12.4
 O 出力専用フィールド、保護されたフィールド

 **注意:** 数値長の精度部分は、デリミタおよび小数点文字プロファイルパラメータの現在の値に関係なく、常にコンマ (,) で区切られます。アルファベットタイプのフィールドについては、精度部分は省略されます。

サブフィールドを使用して、Naturalのダイナミック属性機能によって基本フィールドから分離されたフィールドを指定します。フィールドがさまざまなサブフィールドにダイナミックに分割される場合は、次のようにマークされます。

最初のサブフィールドは通常どおり識別子Fでマークされ、その他のすべてのサブフィールドはSで識別されます。

拡張 1

以下の図に、データセクションの一部を示します。これは、Naturalの内部画面属性バッファのレイアウトを示します。フォーマットバッファの拡張オプション 1 が設定されている場合は、属性バイト (PATTR1~PATTR4) の各フィールドは出力可能なフォーマットになり、適切なフィールドフォーマットバッファエントリに追加されます。拡張は基本フォーマットエントリから"/" (スラッシュ) で区切られます。

PATTR1	DS	X		ATTRIBUTE BYTE 1
P1TMP	EQU	X'80'	1000 0000	TEMPORARY PROTECTED (ONLY PAGE)
P1EXTLNG	EQU	X'80'	1000 0000	EXTENDED LENGTH (ONLY SCREEN)
P1RPA	EQU	X'40'	0100 0000	FIELD CAN BE REPEATED
P1PROT	EQU	X'20'	0010 0000	FIELD IS PROTECTED
P1NUM	EQU	X'10'	0001 0000	FIELD IS NUMERIC
P1SKIP	EQU	P1PROT+P1NUM (X'30')		FIELD WILL BE SKIPPED AUTOMATICALLY
P1HIGH	EQU	X'08'	0000 1000	FIELD IS HIGHLIGHTED
P1BLINK	EQU	X'04'	0000 0100	FIELD IS BLINKING
P1NOND	EQU	P1HIGH+P1BLINK (X'0C')		FIELD IS NON-DISPLAY
P1NHC	EQU	X'02'	0000 0010	FIELD MAY NOT BE PRINTED
P1CURS	EQU	X'01'	0000 0001	SET CURSOR HERE (ONLY UNPROT)
	SPACE			

PATTR2	DS	X		ATTRIBUTE BYTE 2
P2ITAL	EQU	X'80'	1000 0000	ITALIC/CURSIVE
P2MAND	EQU	X'40'	0100 0000	INPUT MANDATORY
P2MFILL	EQU	X'20'	0010 0000	MANDATORY FILL
P2LC	EQU	X'10'	0001 0000	DO NOT TRANSLATE (LOWER CASE)
P2CS2	EQU	X'08'	0000 1000	SECOND CHARACTER SET
P2UL	EQU	X'04'	0000 0100	UNDERLINED
P2RVID	EQU	X'02'	0000 0010	REVERSED VIDEO
P2RL	EQU	X'01'	0000 0001	RIGHT-LEFT
	SPACE			
PATTR3	DS	X	COLOR ATTRIBUTE	ATTRIBUTE BYTE 3
P3TP	EQU	X'80'	1000 0000	TERMINAL PROGRAM AVAILABLE
P3PFK	EQU	X'40'	0100 0000	*COM FIELD
P3NUM	EQU	X'20'	0010 0000	NUMERIC FIELDS

P3HELPR	EQU	X'10'	0001 0000	HELP ROUTINE AVAILABLE
P3FRAME	EQU	X'08'	0000 1000	FRAME ATTRIBUTE
P3NEUTR	EQU	X'07'	0000 0111	NEUTRAL
P3YELL	EQU	X'06'	0000 0110	YELLOW
P3TURQ	EQU	X'05'	0000 0101	TURQUOISE
P3GREEN	EQU	X'04'	0000 0100	GREEN
P3PINK	EQU	X'03'	0000 0011	PINK
P3RED	EQU	X'02'	0000 0010	RED
P3BLUE	EQU	X'01'	0000 0001	BLUE
*		FBI (DB)		(FIELD PROCESSING INFORMATION)
	SPACE			
PATTR4	DS	X		INTERNAL PROCESSING ATTRIBUTES
P4TEXT	EQU	X'80'	1000 0000	FIELD IS TEXT CONSTANT
P4SAME	EQU	X'40'	0100 0000	SAME ATTRIBUTE AS BEFORE

P4NATTR	EQU	X'20'	0010 0000	FIELD NEW ATTRIBUTE
*				PAGE BUFFER, DYNAMIC ATTRIBUTE
P4OVL	EQU	X'10'	0001 0000	FIELD BELONGS TO OVERLAY BUFFER
P4MDT	EQU	X'08'	0000 1000	FIELD HAS BEEN MODIFIED
P4MDTH	EQU	X'04'	0000 0100	UPDATE FROM HELP (PAGE BUFFER)
P4NFLD	EQU	X'04'	0000 0100	FIELD NEW ON SCREEN
*				IF SET FOR OVL, NEW LINE
P4CONT	EQU	X'02'	0000 0010	FIELD IS CONT OF BEFORE
P4LAST	EQU	X'01'	0000 0001	LAST ATTRIBUTE IN BUFFER
**P4HELP	EQU	P4TEXT+P4MDT		HELP REQUEST FOR THIS FIELD

例：

拡張フォーマットバッファエントリ 18820300 は、フィールドが数値で、反転表示モードでは斜体で強調表示されることを示します。フィールドの色はピンクです。

バリュースタックのレイアウト

バリュースタック構造には、以下の3つのモードが可能です。

- 固定フォーマット
すべてのパラメータは区切られずに単純に連結されます。つまり、単一のパラメータは、フォーマットバッファ内のフォーマット記述に従って、またはC構造を適用して、データエリアまたはデータセクションに区切る必要があります。
- デリミタ区切りフォーマット
パラメータは、設定可能なデリミタ文字で区切られます。
- 長さ前置フォーマット
各パラメータの先頭に、フォーマット N3 の長さフィールドが配置されます。長さの表記は明示的です。

属性バッファ

属性バッファはオプションです。これは、パラメータフィールドごとに1バイトの、mdt フラグを示すエントリで構成されます。mdt は、クライアントが各変更可能フィールドに設定する必要があります。このフラグの値は "0" または "1" です。1 の値は、mdt が設定されていることを表します。

例：

この例では、図 1 に 3270 フォーマットの画面イメージを、図 2 に同じフォーマットの生成されたネットデータストリームを示します。Natural マップの名前は NETM002 です。

```

                                TESTMAP NWI

AL20.0 ABCDEFGHIJKLMNOPQRST
NL20.0 1234567890
NL10.4 0000001234.5678
AL20C AAAAABBBBBCCCCDDDDDDZ
N20.0 99999999999999999999

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12-
      Help                -      +                        %%

```


66 サーバーとしての Natural

このドキュメントでは、バッチモードの z/OS および z/VSE オペレーティングシステムと TP モニタ CICS でのサーバーとしての Natural の使用について説明します。

- **z/OS 環境のサーバーとしての Natural** バッチモードの z/OS のクライアント/サーバー環境で Natural がサーバーとして機能できる仕組みについて説明します。
- **z/VSE 環境のサーバーとしての Natural** バッチモードの z/VSE のクライアント/サーバー環境で Natural がサーバーとして機能できる仕組みについて説明します。
- **CICS 環境のサーバーとしての Natural** TP モニタ CICS のクライアント/サーバー環境で Natural がサーバーとして機能できる仕組みについて説明します。サーバー環境での Natural CICS インターフェイスのインストールおよび機能と、この環境に適用される制限についても説明します。

67 z/OS 環境のサーバーとしての Natural

■ 機能	414
■ サーバー環境での Natural ニュークリアスのインストール	415
■ サーバー環境における外部データセットでの出力ファイルおよびワークファイルの処 理	415

このドキュメントは、z/OS 環境にのみ適用されます。以下のトピックについて説明します。

機能

Natural は、プログラミング言語である以外に、クライアント/サーバー環境でサーバーとして機能できます。Natural サブプログラムの実行などのサービスを提供します。サーバー機能の一部は、拡張バッチドライバです。クライアント/サーバー通信の基盤となるプロトコルが多数あり、DB2 のストアードプロシージャの実行やリモートプロシージャコールの実行などがあります。『*Natural* リモートプロシージャコール (RPC)』ドキュメントを参照してください。

Natural サーバースタブ

サーバーとしての Natural は、独立したリージョン内で動作するか、DB2 ストアドプロシージャのようなサーバーサブシステムリージョン内で動作します。Natural をサーバーとして実行するには、サービス固有のサーバースタブが必要です。このサーバースタブは、サーバー製品の一部として提供され、すべてのサービス要求を制御するとともに、Natural サーバーフロントエンドの唯一のインターフェイスとして機能します。

DB2 用や Natural RPC 用など、他にも異なるサーバースタブがあります。

Natural バッチドライバ

Natural バッチドライバ (z/OS 環境の NATOS など) は、拡張されて環境固有のインターフェイスコンポーネントとして機能するようになりました。Natural サーバーセッションを管理し、Natural に環境固有のサービスを提供します。これは、サーバースタブモジュールにリンクするか、個別のモジュールとしてサーバースタブでロードすることができます。

バッチドライバは、スレッドストレージの圧縮、圧縮解除、および外部ストレージデバイスへのロールアウトの機能を持つストレージスレッドを使用して、複数のセッションを作成および制御できます。

サーバー初期化中にバッチドライバがサーバースタブによって初めて呼び出されると、ストレージスレッドがメインストレージに作成されます。ストレージスレッドの数とサイズは、サーバースタブによって決まります。その後、スタティックな Natural セッションが初期化されます。これには、プロファイルパラメータの評価およびスタティックなストレージバッファの割り当てが含まれます。作成された初期化済みストレージスレッドは、メインストレージに個別に保存されます。新しい Natural セッションごとに、この最初の "セッションクローン" がスレッドにコピーされます。

サーバースタブの決定により、セッションはロールアウトされて後で再開されることがあります。**Natural** **ロールサーバー** がドライバによって使用されて、セッションの圧縮スレッドストレージが保存されます。別の方法としては、メインストレージを使用して、圧縮スレッドストレージを保存できます。この場合は、ロールアウト状態のセッションの数はリージョンサイズによって制限されます。

サーバー環境での Natural ニュークリアスのインストール

Natural ニュークリアスおよびそのバッチドライバは、サーバー環境とサーバー以外の環境の両方をサポートするように設計されています。サーバー固有の定義および要件については、該当のドキュメント（『Natural リモートプロシージャコール (RPC)』ドキュメントや『Natural for DB2』ドキュメントなど）を参照してください。

セッション数が少数に制限されておらず、セッションのロールアウトがサポートされるサーバータイプの場合は、サーバーを初期化する前に **Natural ロールサーバー** をインストールし起動する必要があります。これを行うには、Natural パラメータモジュールの SUBSID パラメータが正しい値に設定されていることを確認します。サーバーでは、サーバーの他に、ADALNK もリクエストであるように Adabas リンクインターフェイス (ADALNK) を生成する必要があります。

ローカルまたはグローバルの **Natural バッファプール** を使用できます。ローカルバッファプールを定義すると、サーバーリージョン内のすべてのセッションで共有されます。

論理出力ファイルまたはワークファイル番号をサーバーセッション内で処理に使用する場合は、その番号をセッションの開始時にアクセスメソッドに関連付ける必要があります。これを行うには、マクロ NETWORK および NTPRINT で NATPARM を使用します。すべての出力ファイルおよびワークファイル番号の全範囲を使用可能にする場合は、以下の例のように指定します。

```
NTPRINT (1-31),AM=STD,OPEN=ACC,DEST=*
NETWORK (1-32),AM=STD,OPEN=ACC,DEST=*
```

サブパラメータ DEST=* は、最初の DEFINE WORK FILE または DEFINE PRINTER ステートメントの OUTPUT 節中に一般的な DD 名の生成を定義します（下記参照）。サブパラメータ OPEN=ACC は、プログラム起動時にファイルの事前オープンを回避します。オープンはファイルの初回アクセス時に発行されます。

サーバー環境における外部データセットでの出力ファイルおよびワークファイルの処理

1つのリージョンで多数の同時セッションを実行している場合は、外部出力ファイルおよびワークファイルとのリソースの競合が発生することがあります。出力ファイルまたはワークファイル用の論理名 (DD 名) は、マクロ NTPRINT または NETWORK のサブパラメータ DEST か、同等のダイナミックパラメータ PRINT または WORK によって定義されます（デフォルトは CMPRTnn または CMWKFnn）。通常の Natural バッチ処理では、これらのファイルは論理 (DD) および物理データセット名によって JCL に定義されます。

ただし、DD 名は、各セッションの1つのタスクで排他的に使用するためにオペレーティングシステムによって予約されています。つまり、1つのセッションで処理のために CMWKF01 が開

かれた場合は、閉じられるまで他のセッションではこのファイルを使用できません。他のセッションでこれを開こうとすると、エラーが発生します。

サーバー環境では、すべての出力ファイルおよびワークファイルの要求は専用のI/Oサブタスクによって処理されます。これにより、データセットの安全性が確保され、リソースの競合が回避されます。そのため、Naturalセッションの境界を超えて出力ファイルおよびワークファイルの共有使用が可能になります。つまり、複数のセッションで同じファイルに同時にアクセスできます。これは、CMで始まるDD名を持つ出力ファイルおよびワークファイルにのみ有効です。他のファイルはすべて、排他的で共有不可と見なされます。

出力ファイルおよびワークファイルの排他的使用のため、Natural では以下の2つの機能でサーバー環境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（どちらもサーバー環境用のNatural アプリケーションプログラム内に特別に実装する必要があります）。

- DEFINE WORK FILE または DEFINE PRINTER ステートメントの OUTPUT 節
- データセットのダイナミックアロケーション（アプリケーションプログラミングインターフェイス USR2021N。「SYSEXT-Natural アプリケーションプログラミングインターフェイス」を参照）。

DEFINE WORK FILE および DEFINE PRINTER ステートメントの OUTPUT 節は、以下のいずれかの場合に使用できます。

- ワークファイルまたは出力ファイル用の論理 DD 名を定義する場合
- 物理データセット名を定義する場合
- 出力スプールクラスを定義する場合

DD名が指定された場合は、アクセスメソッドによって、データセットが割り当てられているかどうかチェックされます。割り当てられていないと、エラーが発行されます。データセットは、ライブラリ SYSEXT で提供された USR2021N サブプログラムを使用する任意の Natural プログラムで割り当てることができます。

物理データセット名またはスプールファイルクラスが指定された場合は、DEFINE ... ステートメントの実行中にアクセスメソッド自体によってデータセットがダイナミックに割り当てられます。確実に一意の DD 名が使用されるように、DEST=* を NATPARM ファイルに事前に定義する必要があります。これにより、DD 名の競合が回避されます。

アプリケーションがアプリケーションプログラミングインターフェイス USR2021N を使用している場合は、DD 名変数にアスタリスク値を指定して、アクセスメソッドから一意の DD 名を戻します。この DD 名は、後続の DEFINE ... ステートメントに使用できます。

デフォルトでは、サーバージョブのアクセスプロパティは出力ファイルおよびワークファイルに使用されます。一部のサーバータイプ（Natural 開発サーバーや Natural RPC など）では、偽装がサポートされます。つまり、個別のクライアントアカウントのアクセスプロパティが、排他的な出力ファイルおよびワークファイルに使用されます。詳細については、サーバーのドキュメントの該当セクションを参照してください。

68 z/VSE 環境のサーバーとしての Natural

■ 機能	418
■ サーバー環境での Natural ニュークリアスのインストール	419
■ サーバー環境における外部データセットでの出力ファイルおよびワークファイルの処 理	419

このドキュメントは、z/VSE 環境にのみ適用されます。以下のトピックについて説明します。

機能

Natural は、プログラミング言語である以外に、クライアント/サーバー環境でサーバーとして機能できます。Natural サブプログラムの実行などのサービスを提供します。サーバー機能の一部は、拡張バッチドライバです。クライアント/サーバー通信の基盤となるプロトコルが多数あり、DB2 のストアードプロシージャの実行やリモートプロシージャコールの実行などがあります。『*Natural* リモートプロシージャコール (RPC)』ドキュメントを参照してください。

Natural サーバースタブ

サーバーとしての Natural は、独立したリージョン内で動作するか、DB2 ストアドプロシージャのようなサーバーサブシステムリージョン内で動作します。Natural をサーバーとして実行するには、サービス固有のサーバースタブが必要です。このサーバースタブは、サーバー製品の一部として提供され、すべてのサービス要求を制御するとともに、Natural サーバーフロントエンドの唯一のインターフェイスとして機能します。

DB2 用や RPC 用など、他にも異なるサーバースタブがあります。

Natural バッチドライバ

Natural バッチドライバ (z/VSE 環境の NATVSE など) は、拡張されて環境固有のインターフェイスコンポーネントとして機能するようになりました。Natural サーバーセッションを管理し、Natural に環境固有のサービスを提供します。これは、サーバースタブモジュールにリンクするか、個別のモジュールとしてサーバースタブでロードすることができます。

バッチドライバは、スレッドストレージの圧縮、圧縮解除、および外部ストレージデバイスへのロールアウトの機能を持つストレージスレッドを使用して、複数のセッションを作成および制御できます。

サーバー初期化中にバッチドライバがサーバースタブによって初めて呼び出されると、ストレージスレッドがメインストレージに作成されます。ストレージスレッドの数とサイズは、サーバースタブによって決まります。その後、スタティックな Natural セッションが初期化されます。これには、プロファイルパラメータの評価およびスタティックなストレージバッファの割り当てが含まれます。作成された初期化済みストレージスレッドは、メインストレージに個別に保存されます。新しい Natural セッションごとに、この最初の "セッションクローン" がスレッドにコピーされます。

サーバースタブの決定により、セッションはロールアウトされて後で再開されることがあります。ロールキャッシュがドライバによって使用されて、セッションの圧縮スレッドストレージが保存されます。

サーバー環境での Natural ニュークリアスのインストール

Natural ニュークリアスおよびそのバッチドライバは、サーバー環境とサーバー以外の環境の両方をサポートするように設計されています。サーバー固有の定義および要件については、該当のドキュメント（『Natural リモートプロシージャコール (RPC)』ドキュメントや『Natural for DB2』ドキュメントなど）を参照してください。

ローカルまたはグローバルの **Natural バッファプール** を使用できます。ローカルバッファプールを定義すると、サーバーリージョン内のすべてのセッションで共有されます。

論理出力ファイルまたはワークファイル番号をサーバーセッション内で処理に使用する場合は、その番号をセッションの開始時にアクセスメソッドに関連付ける必要があります。これを行うには、マクロ NETWORK および NTPRINT で NATPARM を使用します。すべての出力ファイルおよびワークファイル番号の全範囲を使用可能にする場合は、以下の例のように指定します。

```
NTPRINT (1-31),AM=STD,OPEN=ACC,DEST=*
NETWORK (1-32),AM=STD,OPEN=ACC,DEST=*
```

サブパラメータ DEST=* は、最初の DEFINE WORK FILE または DEFINE PRINTER ステートメントの OUTPUT 節中に一般的な DD 名の生成を定義します（下記参照）。サブパラメータ OPEN=ACC は、プログラム起動時にファイルの事前オープンを回避します。オープンがファイルの初回アクセス時に発行されます。

サーバー環境における外部データセットでの出力ファイルおよびワークファイルの処理

1つのリージョンで多数の同時セッションを実行している場合は、外部出力ファイルおよびワークファイルとのリソースの競合が発生することがあります。出力ファイルまたはワークファイル用の論理名（DD 名）は、マクロ NTPRINT または NETWORK のサブパラメータ DEST か、同等のダイナミックパラメータ PRINT または WORK によって定義されます（デフォルトは CMPRT nn または CMWKF nn ）。通常の Natural バッチ処理では、これらのファイルは論理ファイルおよび物理データセット名によって JCL に定義されます。

ただし、DD 名は、各セッションの1つのタスクで排他的に使用するためにオペレーティングシステムによって予約されています。つまり、1つのセッションで処理のために CMWKF01 が開かれた場合は、閉じられるまで他のセッションではこのファイルを使用できません。他のセッションでこれを開こうとすると、エラーが発生します。

サーバー環境では、すべての出力ファイルおよびワークファイルの要求は専用の I/O サブタスクによって処理されます。これにより、データセットの安全性が確保され、リソースの競合が回避されます。そのため、Natural セッションの境界を超えて出力ファイルおよびワークファイル

の共有使用が可能になります。つまり、複数のセッションで同じファイルに同時にアクセスできます。

出力ファイルおよびワークファイルの排他的使用のため、Naturalでは以下の機能でサーバー環境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（どちらもサーバー環境用のNaturalアプリケーションプログラム内に特別に実装する必要があります）。

■ DEFINE WORK FILE または DEFINE PRINTER ステートメントの OUTPUT 節

これらのステートメントの OUTPUT 節は、以下のいずれかの場合に使用できます。

■ ワークファイルまたは出力ファイル用の論理ファイル名を定義する場合

■ 物理データセット名を定義する場合

物理データセット名またはスプールファイルクラスが指定された場合は、DEFINE ... ステートメントの実行中にアクセスメソッド自体によってデータセットが動的に割り当てられます。確実に一意のファイル名が使用されるように、DEST=* を NATPARM ファイルに事前に定義する必要があります。これにより、ファイル名の競合が回避されます。

69 CICS 環境のサーバーとしての Natural

■ 機能	422
■ サーバー環境での Natural CICS インターフェイスのインストール	422
■ 制限	423

このドキュメントは、CICS 環境にのみ適用されます。以下のトピックについて説明します。

以下の項目も参照してください。

- CICS 環境での Natural
- Natural リモートプロシージャコール (RPC)

機能

サーバーとしての Natural

Natural は、プログラミング言語である以外に、クライアント/サーバー環境でサーバーとして機能できます。Natural サブプログラムの実行などのサービスを提供します。クライアント/サーバー通信の基盤となるプロトコルが多数あり、DB2 のストアードプロシージャの実行やリモートプロシージャコールの実行などがあります。『Natural リモートプロシージャコール (RPC)』ドキュメントを参照してください。

Natural サーバースタブ

サーバーとしての Natural は、独立したリージョン内で動作するか、DB2 ストアドプロシージャのようなサーバーサブシステムリージョン内で動作します。Natural をサーバーとして実行するには、サービス固有のサーバースタブが必要です。このサーバースタブは、サーバー製品の一部として提供され、すべてのサービス要求を制御するとともに、Natural サーバーフロントエンドの唯一のインターフェイスとして機能します。

DB2 用や RPC 用など、他にも異なるサーバースタブがあります。

サーバー環境での Natural CICS インターフェイスのインストール

Natural CICS インターフェイスをインストールするときに、Natural サーバー環境として機能させるために定義する必要のある事項は特にありません。スレッドタイプまたはロールタイプ (CICS ロール機能またはロールサーバー) に要件はありません。

実際には、Natural サーバーセッションでは、端末内の Natural セッションのような "標準" のセッションと、CICS 環境の Natural を共有できます。違いは、Natural サーバーセッションの場合は、Natural CICS インターフェイスは端末やプリンタなどの基本的な機能には対応しませんが、サーバースタブには対応することです。CICS に関しては、Natural サーバーセッションは一連の非同期 CICS タスクであり、セッションコンテキスト (セッション再スタートデータ) はサーバースタブによって 8 バイトの一意的セッション ID で管理されます。

制限

CICS 環境でサーバーとして Natural を使用する場合は、以下の制限が適用されます。

1. CICS 環境での Natural サーバーセッションは、擬似会話型モードでのみ実行可能です。
Natural サーバーセッションは会話型モードでは実行できず、Natural CICS インターフェイスは常にサーバスタブに制御を渡す必要があるため、CICS 環境での Natural サーバーセッションには PSEUDO=ON が強制されます。同様の理由により、TYPE=GETM スレッドを使用している Natural サーバーセッションには RELO=ON が強制されます。
2. Natural によって呼び出された 3GL プログラムは、Natural サーバーセッションが CICS で非同期に実行されているので CICS 端末 (TCTTE) が利用不可であることを認識する必要があります。
3. プロファイルパラメータ ADAMODE は、1 または 2 に設定する必要があります。そうでないと、Adabas が Natural サーバーセッションのダイアログステップごとに異なる UQE ID を生成します。
4. プロファイルパラメータ PROGRAM または Natural による同等のバックエンドプログラム設定は適用されません。これは、セッション終了時の Natural CICS インターフェイスからサーバスタブへのロジックフローが、潜在的なバックエンドプログラムによって中断されたり改ざんされたりしないようにするためです。
5. Natural 出力ファイルおよびワークファイルのファイル名設定で NCIPARM 端末 ID 変数 &TID を使用する場合は注意が必要です。Natural サーバーセッションは非同期に実行されるため、挿入する (一意の) 端末 ID またはその他の 4 文字の一意なセッション ID はありません。CICS/TS 1.3 以降では、このような Natural 出力ファイルおよびワークファイルの CICS 一時ストレージを処理する場合、CICS インターフェイスは内部的に QNAME オプションを使用します。つまり、内部的には 16 バイトの一時的ストレージキュー名が使用されます (8 バイトの一意なサーバーセッション ID がファイルの DEST の指定に追加されます)。このような CICS 一時ストレージキューは、元のセッションでしかアクセスできません。

70 Natural の実行 - その他のトピック

このドキュメントでは、Natural の実行に関する全般的な情報を示します。

- 非同期処理
- ダブルバイト文字セット
- 入力/出力デバイス
- バックエンドプログラムの呼び出し規則
- **Natural 31** ビットモードのサポート
- **LE** サブプログラム
- 外部 **SORT**

このドキュメントで使用されている用語については、『用語集』を参照してください。

71 非同期処理

- 非同期 Natural セッションの識別 428
- 非同期 Natural セッションの出力処理 428
- 予期しない／望ましくない入力の処理 429
- その他のプロファイルパラメータの考慮事項 429

このドキュメントでは、Natural でサポートされているすべての TP モニタ環境で使用できる、非同期の Natural 処理について説明します。

非同期 Natural セッションはいずれの端末にも関連付けられていないセッションなので、端末ユーザーと対話することはできません。これを使用して、時間のかかるタスクを「バックグラウンド」で実行し、ユーザーがタスクの終了を待つ必要がないようにします。

以下のトピックについて説明します。

関連トピック：

- CICS 環境での Natural の非同期処理
- Com-plete/SMARTS 環境での Natural の非同期処理
- UTM 環境での非同期トランザクション処理

非同期 Natural セッションの識別

セッションが非同期であることを識別するために、Natural システム変数 *DEVICE に値 ASYNCH が割り当てられます。

 **注意:** *DEVICE の値は Natural プロファイルパラメータ TTYPE と任意の SET CONTROL 'T=XXXX' ステートメントで変更できます。『パラメータリファレンス』ドキュメントのプロファイルパラメータ TTYPE と『端末コマンド』ドキュメントの端末コマンド %T= を参照してください。

非同期 Natural セッションの出力処理

非同期セッションはいずれの端末にも関連付けられていないため、セッションで生成された出力を単純に画面表示できません。明示的に出力先を指定する必要があります。この出力先は、Natural の起動時に Natural プロファイルパラメータ SENDER を使用して指定します。SENDER の出力先は、ハードコピー出力とプライマリレポートに適用されます。追加のレポートは、同期オンラインセッションの場合とまったく同様に、DEFINE PRINTER ステートメントで指定された出力先に送信されます。

非同期セッションでも Natural エラーが発生することがあるため、Natural エラーメッセージの送信先も指定する必要があります。この指定には、Natural プロファイルパラメータ OUTDEST を使用します。このパラメータには、エラーメッセージをオペレータコンソールに送信するオプションもあります。Natural は、エラーメッセージを送信した後、非同期セッションを終了します。

非同期 Natural セッションによる予期しない出力を処理できるようにプロファイルパラメータ SENDER および OUTDEST を設定する必要があります。そうしないと、場合によっては非同期 Natural セッションがアベンドすることもあります。

予期しない／望ましくない入力の処理

非同期 Natural セッションでは、実行する Natural プログラムおよび Natural システムコマンドの名前の入力に Natural スタックしか使用できません。Natural プログラムまたは Natural システムコマンドが未処理の Natural エラーによって失敗した場合、または Natural スタック全体の残量がなくなり NEXT モードが入力された場合は、非同期 Natural セッションは終了メッセージ NAT9943 を表示して終了します。

デフォルトでは、INPUT 要求に対して、使用中の TP モニタおよび TTYPE 設定に応じて CLEAR キーまたは EOF インジケータのどちらかが Natural に戻されます。この処理は、意図せずプログラムによって INPUT ステートメントが実行されたときにエラーラップ状態になるのを防ぐのに役立ちます。ENTER キーインジケータを戻すには、INPUT ステートメントの前に SET CONTROL 'N' ステートメントを発行します。



ヒント: システム変数 *SCREEN-IO を適切に評価することにより、使用しているアプリケーションと非同期セッションを互換することができます。

その他のプロファイルパラメータの考慮事項

非同期 Natural セッションの場合は、以下の Natural プロファイルパラメータについて考慮する必要があります。

プロファイルパラメータ	コメント
AUTO	非同期セッションにアルファベット以外のユーザー ID が使用されていることがあります。この場合、AUTO=ON は失敗します。
CM	Natural セッションが誤って NEXT レベルになった場合、望ましくない入力状況が発生する場合があります。CM=OFF を設定すると、そのような状況でセッションは直ちに終了します。
ENDMSG	NAT9995（正常終了メッセージ）は、ENDMSG=OFF を指定することで非表示にできます。
IMSG	Natural 初期化エラーメッセージおよび警告は、IMSG=OFF を指定することで非表示にできます。
MENU	非同期セッションにはコマンド入力用に Natural スタックしかないため、MENU=OFF を指定し、ダイレクトコマンドを使用して Natural 内を移動することをお勧めします。

非同期処理

PLOG	すべてのパラメータを1行ずつSENDERの宛先に送信することによって、ダイナミックパラメータロギングが実行されます。
PROGRAM	標準バックエンドプログラム/トランザクションがインストールで定義される場合は、これが非同期で実行可能か、または端末結合セッションでのみ処理するのが望ましいかを確認する必要があります。PROGRAM=0を指定すると、バックエンドロジックはバイパスされます。

72 ダブルバイト文字セット

▪ Natural プロファイルパラメータ SOSI	432
▪ 出力フォーマット指定	432
▪ DBCS サポートのためのパラメータ定義	432
▪ エディタプロファイルオプション	433
▪ 入力データのチェック	433
▪ 出力データの調整	434
▪ Natural スタックデータ	434
▪ DBCS 処理用のアプリケーションプログラミングインターフェイス	435

このドキュメントは、ダブルバイト文字セット（DBCS）を使用するアジア諸国のみを対象としています。DBCS 端末およびプリンタをサポートするために Natural に実装されているすべての機能について説明します。以下のトピックについて説明します。

Natural プロファイルパラメータ SOSI

SBCS 文字と DBCS 文字が混在する英数字フィールドでは、DBCS 文字列と SBCS 文字列は SO（シフトアウト）および SI（シフトイン）と呼ばれるシフトコードで分離されます。Natural プロファイルパラメータ SOSI は、現在の環境で使用されているシフトインコードおよびシフトアウトコードの値を Natural に渡すために使用されます。

内部的には IBM 文字 X'0E' および X'0F' を使用することを強くお勧めします。この手法を使用すると、すべてのアプリケーションとデータを互換性のある方法で処理でき、ネットワークでサポートしているさまざまなメインフレームタイプで同じ Natural アプリケーションを使用し、同じデータを処理することができます。

このパラメータの詳細については、SOSI を参照してください。

出力フォーマット指定

Natural セッションパラメータ PM=D を使用して、DBCS 専用フィールドを定義します。DBCS 専用フィールドで使用できるのは有効な DBCS 文字のみで、シフトアウト/シフトイン文字（SO/SI）は使用できません。セッションパラメータ PM=D が指定されているフィールドを表示するために、IBM 端末には画面属性 X'43F8' が追加されます。富士通端末の場合は、必要なシフトアウト/シフトイン文字（SO/SI）でフィールド内容が囲まれます。

DBCS サポートのためのパラメータ定義

ダブルバイト文字セットをサポートするには、Natural のセットアップで以下のパラメータを指定する必要があります。

パラメータ	説明
TS=ON	ラテン文字の小文字を使用できない場合、すべての Natural システム出力は NATCONFIG モジュールの NTTABL マクロで定義された変換テーブルを使用して変換されます。
SOSI=(0E,0E,0F,0F,1)	IBM ハードウェアの DBCS シフトアウトおよびシフトインの値を定義します。
SOSI=(28,28,29,29,0)	富士通ハードウェアの DBCS シフトアウトおよびシフトインの値を定義します。

パラメータ	説明
LC=ON	いずれの入力データも大文字に変換されないため、DBCS入力データが再度破壊される可能性があります。

TS=ONの他に、メッセージを大文字に変換するためのパラメータが複数のNaturalコンポーネントによって提供されています。詳細については、「TS - システムライブラリでプログラムからの出力を変換」で、大文字に変換するための他のパラメータに関する説明を参照してください。

エディタプロファイルオプション

DBCSまたは半角カタカナ文字をNaturalエディタのいずれかに入力する場合、エディタプロファイルに以下のエディタの全般的なデフォルトオプションを設定して、DBCSまたは半角カタカナ文字を含む文字定数またはフィールド名が意図せず大文字に変換されるのを回避する必要があります。

オプション	値	説明
Editing in Lower Case	Y	ソースコード中の小文字は自動的に大文字に変換されません。このオプションは、DBCSまたは半角カタカナ文字を使用する場合に必要です。
Dynamic Conversion of Lower Case	N	ソースコードは入力されたときのままです。このオプションは、半角カタカナ文字を使用する場合に必要です。

エディタの全般的なデフォルトオプションの詳細については、「全般的なデフォルト設定」を参照してください。エディタプロファイルの詳細については、『エディタ』ドキュメントの「エディタプロファイル」を参照してください。すべてのユーザーがこれらのオプションを変更しなくてもすむように、ユーザー出口ルーチンUSR0070Pによって、インストールのデフォルトプロファイルを変更できます。このユーザー出口ルーチンではDBCSもサポートされます。「Naturalの構成」セクションの「[USR0070P - エディタプロファイルのユーザー出口](#)」を参照してください。

入力データのチェック

フィールドにセッションパラメータPM=Dが設定されている場合は、入力データについて以下の点がチェックされます。

- 偶数のバイト数を含んでいる。
- 有効なDBCS文字のみを含んでいる。
- シフトアウト/シフトイン文字(SO/SI)を含んでいない。

非 DBCS 文字の検出には ICU が必要なので、ICU を使用できない場合（つまり、プロファイルパラメータ `CFICU=OFF` が設定されている場合）は、このチェックは行われません。

出力データの調整

ユーザー操作のためにウィンドウが表示される場合に、すでに表示されている DBCS 文字にウィンドウがオーバーレイしたり、ウィンドウサイズに合わせて切り捨てられた DBCS 文字がウィンドウ自体に含まれたりすることがあります。オーバーレイは、INPUT ステートメントで `NO ERASE` オプションを使用している場合も発生することがあります。オーバーレイなどの場合に画面が乱れないようにするためには、以下の対処法を実行し、必要に応じて出力データを調整します。

- フィールドにセッションパラメータ `PM=D` が設定されている場合は、孤立バイト（DBCS 文字の部分的なオーバーレイの結果として表示される、データの先頭または末尾に残された1バイト）は属性によって置き換えられます。この操作によって、確実に、有効な DBCS 文字のみが表示されます。
- プロファイルパラメータ `SOSI` が設定されている場合は、`PM=D` が指定されていない英数文字フィールドの内容がシフトアウト/シフトイン文字（`SO/SI`）についてチェックされます。対応するシフトイン文字（`SI`）がないシフトアウト文字（`SO`）が見つかった場合は、出力データの最後の文字がシフトイン文字（`SI`）に置き換えられるか、最後の2文字がシフトイン文字（`SI`）とそれに続く空白に置き換えられます。対応するシフトアウト文字（`SO`）がないシフトイン文字（`SI`）が見つかった場合は、出力データの最初の文字がシフトアウト文字（`SO`）に置き換えられるか、最初の2文字が空白とそれに続くシフトアウト文字（`SO`）に置き換えられます。この操作によって、確実に、DBCS文字がシフトアウト/シフトイン文字（`SO/SI`）で正しく囲まれます。

Natural スタックデータ

DBCS 文字が意図せずデリミタ文字または制御文字として解釈されるのを避けるには、Natural スタックに配置されるデータに DBCS 文字が含まれる場合は、STACK ステートメントの `FORMATTED` オプションを使用する必要があります。

STACK ステートメントの詳細については、『ステートメント』ドキュメントを参照してください。

Natural スタックの詳細については、『プログラミングガイド』を参照してください。

DBCS 処理用のアプリケーションプログラミングインターフェイス

DBCS 処理をサポートするために、以下のユーザーアプリケーションプログラミングインターフェイス (API) を使用できます。

- USR4211N - DBCS 文字の取得
- USR4213N - DBCS サポートの文字列操作

これらの API は、Natural ライブラリ SYSEXT にサブプログラムとして含まれています。API の使用方法の詳細情報は、対応するテキストメンバ (USRxxxxT) に含まれています。『ユーティリティ』ドキュメントの「SYSEXT ユーティリティ-Natural アプリケーションプログラミングインターフェイス」も参照してください。

USR4211N - DBCS 文字の取得

アプリケーションプログラミングインターフェイス USR4211N を使用すると、DBCS サポートの可用性および定義されている SOSI 文字に関する情報を取得できます。

USR4213N - DBCS サポートの文字列操作

アプリケーションプログラミングインターフェイス USR4213N を使用すると、以下の機能を実行できます。

- 通常のラテン文字列を対応する DBCS 文字列に変換します。
- ラテン文字データのみを含む DBCS 文字列をシングルバイト文字列に変換します。
- 文字列の先頭と末尾に現在のシフトコードを追加します。
- 文字列から先頭と末尾のシフトコードを削除します。

最後の 2 つの機能は、ネイティブ DBCS 文字列の生成またはネイティブ DBCS 文字列以外の混在モードデータの生成に使用できます。

73 入力／出力デバイス

■ 端末サポート	438
■ ライトペンサポート	438
■ プリンタのサポート	440

このドキュメントでは、Naturalでサポートされる入力／出力デバイスに関する追加情報を示します。

以下のトピックについて説明します。

端末サポート

Naturalは、IBMおよびSiemensメインフレームコンピュータで使用できるように多種多様の端末タイプをサポートします。TP モニタ環境では端末タイプ情報が自動的にNaturalに提供されないため、Naturalプロファイルパラメータ `TTYTYPE` を使用して、Naturalが適切な変換ルーチンをアクティブにして特定のタイプの端末を操作できるようにします。

関連トピック：

- [NTDVCE - 端末デバイス指定テーブル](#)
- [端末通信](#)
- [Natural コンフィグレーションテーブル](#) (さまざまな I/O トランザクションに関するトピック)
- [Natural でサポートされる Siemens 端末タイプ](#)
- [端末コマンド](#)

ライトペンサポート

ライトペンのサポートは、端末コマンド `%RM` によって拡張されています。このコマンドは、画面のすべてのライトペン対応フィールドを書き込み禁止にします。ユーザーはライトペンでフィールドを選択できますが、フィールドの内容を上書きすることはできません。

ライトペン対応のフィールドは、強調表示 (セッションパラメータ `AD=I`) または点滅表示 (`AD=B`) され、フィールドの先頭文字がライトペン指示文字 (下記参照) である必要があります。フィールドをライトペンで選択すると指示文字が変わるので、ライトペンで選択したフィールドの処理を指示文字の値に基づいて行うことができます。

以下の指示文字を使用できます。

文字	意味
?	Enter キーを押す前に複数のフィールドを選択できます。
>	選択したフィールドをもう一度選択すると、この文字は疑問符?に変わります。?と>は相互に切り替わります。
&	1つのフィールドのみ選択でき、Enter キーはフィールドと MDT (変更可能データタグ)の両方に適用されます。
' ' (空白)	1つのフィールドのみ選択でき、MDTのみが表示されます。

指示文字で、選択フィールド (?、>) とアテンションフィールド (&、空白または空) を区別する必要があります。選択フィールドでは、データ転送はすぐに行われないので、複数のフィールドを選択できます。アテンションフィールドでは、直ちに処理が行われます。

SELECT CURSOR キーは、ライトペン選択をエミュレートします。選択するフィールドにカーソルを移動し、SELECT CURSOR を押すと、そのフィールドが選択されます。

ライトペン使用のサンプル Natural プログラム

```

RESET #FIELD-1 (A8)
  #FIELD-2 (A8) #FIELD-3 (A8) #CV-1 (C) #CV-2 (C) #CV-3 (C)
SET KEY ALL
/* SET CONTROL 'RM' IS A TOGGLE. AFTER IT IS EXECUTED ONCE MAKE IT A
/* COMMENT, SO THAT YOU DO NOT TOGGLE IT 'OFF'.
**SET CONTROL 'RM'
REPEAT
  IF *PF-KEY NOT = 'ENTR' AND *PF-KEY NOT = 'PEN' ESCAPE BOTTOM
  MOVE (AD=I CD=YE) TO #CV-1
  MOVE (AD=I CD=RE) TO #CV-2
  MOVE (AD=I CD=BL) TO #CV-3
  MOVE ' FIELD-1' TO #FIELD-1
  MOVE '&FIELD-2' TO #FIELD-2
  MOVE '?FIELD-3' TO #FIELD-3
  INPUT (SG=OFF IP=OFF)
    01/01 #FIELD-1 (CV=#CV-1 AD=M)
    03/01 #FIELD-2 (CV=#CV-2 AD=M)
    05/01 #FIELD-3 (CV=#CV-3 AD=M)
  WRITE 'PF-KEY =' *PF-KEY
  IF #CV-1 MODIFIED WRITE '#CV-1 MODIFIED' #FIELD-1
  IF #CV-2 MODIFIED WRITE '#CV-2 MODIFIED' #FIELD-2
  IF #CV-3 MODIFIED WRITE '#CV-3 MODIFIED' #FIELD-3
LOOP
END

```

プリンタのサポート

以下のトピックについて説明します。

- **プリンタ送り制御文字**
- **Natural によるレーザープリンタのサポート**

プリンタ送り制御文字

プリンタ送り制御文字は、DEFINE PRINTER ステートメントを以下のように使用して、Natural プログラム内で生成できます。

```
....
DEFINE PRINTER (n) OUTPUT 'name'
DEFINE PRINTER (n+1) OUTPUT 'CCONTROL'
....
```

プリンタ (n) のすべての Natural 出力が通常の Natural レポート出力ルールに従い、プリンタ (n+1) のすべての Natural 出力もプリンタ (n) に出力されるように、両方の DEFINE PRINTER ステートメントがともに動作します。Natural は、このレポート用のプリンタ送り制御文字は生成しません。したがって、出力変数の最初の文字は制御文字です。

このメソッドを使用すると、レーザープリンタシステムの制御文字とラインプリンタのチャンネル送り文字を通常の Natural 出力レポートでマージすることができます。

プリンタ送り制御文字のサンプル Natural プログラム

```
....
DEFINE PRINTER (1) OUTPUT 'CMPRT01'
DEFINE PRINTER (2) OUTPUT 'CCONTROL'
WRITE (1) 'TEST '
WRITE (2) NOTITLE '+TEST'
MOVE H'5A' TO A(A1) WRITE (2) A (PM=C) '....'
....
```

列 0 から始まるスプールファイル内の対応する 16 進数データは、以下のとおりです。

```
I..I..I..I..I..I..I..I..I..I..I..I..I..I..I..I
F1 E3 C5 E2 E3
1 T E S T
4E E3 C5 E2 E3
+ T E S T 5A . . .
. .
Üe . . .
```

CCONTROL は、プリンタ $n-1$ に関連付けられた特殊なプリンタ制御テーブルの名前です。これは変更できません。

Natural によるレーザープリンタのサポート

Natural では IBM 3800 レーザープリンタシステムがサポートされます。

3800 プリンタシステム用のレポートの制御と割り当てには、DEFINE PRINTER ステートメントを使用します。このステートメントを使用して、レポート 1 の Natural 出力が 3800 プリンタシステムにルーティングされるように指定できます。

```
DEFINE PRINTER (1) OUTPUT 'LAS3800'
  I I => 1-31 for CMPRT01 to CMPRT31
  . . . .
```

INTENS パラメータの設定に基づいて、Natural は各行を最大 4 回繰り返し、Natural 属性 AD=D、AD=I、AD=C、および AD=V を認識します（セッションパラメータ AD を参照）。

1 行目の最初の列には ASA 制御コードが含まれ、2 番目の列には最初のフォントの 3800 フォント制御文字（16 進数の F0）が含まれます。2 から nnn までの列には、属性 AD=I、AD=C、または AD=V のフラグが設定されていない出力データが含まれます。

2 行目の最初の列には ASA 制御コード + （行送りなしの出力）が含まれ、2 番目の列には 2 番目のフォントの 3800 フォント制御文字（16 進数の F1）が含まれます。2 から nnn までの列には、AD=I のフラグが設定された出力データが含まれます。

3 行目の最初の列には ASA 制御コード + （行送りなしの出力）が含まれ、2 番目の列には 3 番目のフォントの 3800 フォント制御文字（16 進数の F2）が含まれます。2 から nnn までの列には、AD=C のフラグが設定された出力データが含まれます。

4 行目の最初の列には ASA 制御コード + （行送りなしの出力）が含まれ、2 番目の列には 4 番目のフォントの 3800 フォント制御文字（16 進数の F3）が含まれます。2 から nnn までの列には、AD=V のフラグが設定された出力データが含まれます。

4 未満の値で INTENS を指定している場合は、サポートされていないすべてのフォントは 16 進数の F0 で出力されます。

レーザープリンタ使用のサンプル Natural プログラム

```

.....
DEFINE PRINTER (1) OUTPUT 'LAS3800'
WRITE (1) 'FIRST' 'SECOND' (AD=I) 'THIRD' (AD=C) 'FOURTH' (AD=V)
.....

```

列 0 から始まるスプールファイル内の対応する 16 進数データは、以下のとおりです。

```

I..I..I..I..I..I..I..I..I..I..I..I..I..I..I..I..I..I..I..I..I..I..I..I..I..I..I..I..I..I..I..I
40 F0 C6 C9 D9 E2 E3 40 40 40 40 40 40 40 40 40 40 40 40 40 40 40 40 40 40 40 40 (hex)
   0 F I R S T
4E F1 40 40 40 40 40 40 40 40 E2 C5 C3 E4 D5 C4 C4 40 40 40 40 40 40 40 40 40 40 (hex)
+ 1                               S E C O N D
4E F2 40 40 40 40 40 40 40 40 40 40 40 40 40 40 E3 C8 C9 D9 D4 40 40 40 40 (hex)
+ 2                               T H I R D
4E F3 40 40 40 40 40 40 40 40 40 40 40 40 40 40 40 40 40 40 40 40 C5 (hex)
+ 3                                                                    F

```

レーザープリンタ使用のサンプル JCL

```

.....
//xxxx JOB xxxxx,.....
.
//xxxxx EXEC PGM= XXXXXX;.....
.
// PARM='INTENS=4,XXXX,.....
.
.
//OUT1 OUTPUT PAGEDEF=XXXX,FORMDEF=XXXX,TRC=ON
.           I           I
.           I           I => 3800 form definition
.           I
.           I => 3800 page definition .
//CMPRT01 DD SYSOUT=Y
//          DCB=(RECFM=FBA,LRECL=133),OUTPUT=*,OUT1
//          CHARS=(WWW,XXXX,YYYY, ZZZZ)
.           I
.           I => IBM font names
.....

```

74 バックエンドプログラムの呼び出し規則

■ バックエンドプログラムの呼び出し規則（バッチモード）	444
■ CICS 固有の考慮事項	445
■ IMS/TM 固有の考慮事項	445
■ サンプルバックエンドプログラム	445

このドキュメントでは、バックエンドプログラムの呼び出しに適用される規則について説明します。

注：

- このセクションは BS2000/OSD には適用されません。Natural の『TP モニタインターフェイス』ドキュメントの「UTM 環境の Natural」で「Natural 以外のプログラムの呼び出し」および「チェーンで結ばれた部分的な UTM プログラムの呼び出し」を参照してください。
- z/OS 環境でのバッチモードを除いて、Natural セッションが Natural 開発サーバーで実行中の場合は、指定されたバックエンドプログラムは起動されません。

以下のトピックについて説明します。

バックエンドプログラムの呼び出し規則（バッチモード）

プロファイルパラメータ PROGRAM を指定している（または SYSEXTP ライブラリのサブプログラム CMPGMSET を呼び出して Natural セッション中にダイナミックに設定する）場合は、セッションが正常終了していても異常終了していても、バックエンドプログラムが起動されます。バックエンドプログラムは標準の OS リンケージ規則を使用して呼び出し、制御を呼び出し元に返す必要があります。

バックエンドプログラムが使用可能な場合は、セッション終了メッセージは表示されません。Natural TERMINATE ステートメントの *operand1* を使用して指定された 0 以外のユーザーリターンコードは、Natural エラーメッセージ NAT9987 によって示されます。

以下の情報を含むパラメータエリアがバックエンドプログラムに渡されます。

- Natural システムまたはユーザーのリターンコードを保持するフルワード
- 72 文字の Natural 終了メッセージ
- Natural 終了データの長さ（または 0）を保持するフルワード
- TERMINATE ステートメントの *operand2* によって渡された終了データ（存在する場合）

バックエンドプログラムのパラメータエリアの長さは 80 バイト以上です。バックエンドプログラムのパラメータエリアの DSECT レイアウトを含む NAMBCKP マクロは、Natural ソースライブラリで提供され、アセンブラのバックエンドプログラムで使用されます。

CICS 固有の考慮事項

CICS 環境では、バックエンドプログラムのパラメータデータは COMMAREA と TWA で渡されます。TWA では、リターンコードとメッセージを含む 80 バイトのみが渡され、長さフィールドにはバックエンドプログラムのパラメータエリア全体をポイントするアドレスが含まれます。Natural が EXEC CICS LINK を使用して起動されている場合も、同じ TWA が提供されます。Natural の『TP モニタインターフェイス』ドキュメントの「CICS 環境での Natural」および「LINK を介して起動されたフロントエンド」も参照してください。

Natural/CICS 生成マクロ (NCIPARM) でパラメータ COMAMSG=NO を設定している場合は、終了データのみが COMMAREA で渡されます。

IMS/TM 固有の考慮事項

IMS/TM のダイアログ指向の環境では、バックエンドプログラムの呼び出し規則が異なります。バックエンドプログラムはプログラム間の切り替えによって呼び出され、バックエンドプログラムの名前が IMS/TM トランザクションコードとして使用されます。この場合は、プログラム間の切り替えが行われる前に Natural 環境が終了します。Natural の『TP モニタインターフェイス』ドキュメントの「IMS/TM 環境の Natural - 環境」で「Natural プロファイルパラメータ PROGRAM のサポート」を参照してください。

サンプルバックエンドプログラム

以下に、いくつかのサンプルプログラムを示します。

COBOL によるバッチおよび **TSO** 環境用のサンプルバックエンドプログラム：

```
LINKAGE SECTION
  01  BACKEND-PARM-AREA.
  02  TERMINATION-RETURN-CODE          PIC S9(8) COMP.
  02  TERMINATION-MESSAGE              PIC X(72).
  02  TERMINATION-DATA-LENGTH          PIC S9(8) COMP.
  02  TERMINATION-DATA                 PIC X(100)
```

```
...
PROCEDURE DIVISION USING BACKEND-PARM-AREA
```

アセンブラによるバッチおよび **TSO** 環境用のサンプルバックエンドプログラム：

```
BACKPROG CSECT
          SAVE   (14,12)
          LR     11,15
          USING  BACKPROG,11
          L     2,0(1)
          USING  BCKPARAM,2
...
          RETURN (14,12)
BCKPARAM NAMBCKP
          END
```

アセンブラによる **CICS** 用のサンプルバックエンドプログラム：

```
L         2,DFHEICAP
USING    BCKPARAM,2
...
BCKPARAM NAMBCKP
          END
```

バッチモード (**z/OS** および **z/VSE**) 用のサンプルバックエンドプログラム **XNATBACK**：

バッチモード用のサンプルプログラムは、XNATBACKとしてNaturalソースライブラリで提供されます。このプログラムは、SYSPRINT (z/OS) /SYSLST (z/VSE) とオペレータコンソールの両方にNatural終了メッセージを発行します。潜在的な終了データはSYSPRINT/SYSLSTにダンプフォーマットで出力されます。

75 Natural 31 ビットモードのサポート

通常、Natural は以下の設定で実行されます。

```
AMODE=31
```

```
RMODE=ANY
```

これ以外の設定については、該当する環境のドキュメントを参照してください。

76 LE サブプログラム

■ IBM LE サブプログラムのサポート	450
■ Natural による LE サブプログラムサポートの有効化	450
■ LE ランタイムオプションの引き渡し	451
■ LE アベンド処理	452

このドキュメントは、z/OS バッチモード、z/VSE バッチモード、IMS/TM、および TSO に適用されます。Natural で IBM 言語環境 (LE) サブプログラムをサポートする方法について説明します。

以下のトピックについて説明します。

IBM LE サブプログラムのサポート

IBM 言語環境 (LE) サブプログラムをサポートするには、CALL ステートメントで LE サブプログラムを呼び出せるように Natural を準備する必要があります。LE サブプログラムは、スタティック (プロファイルパラメータ CSTATIC および RCA) でも、Natural のダイナミックサブプログラムでもかまいません。

Natural のダイナミックサブプログラム (LE および LE 以外) は、LE サービス (CEEFETCH または CEELoad マクロ) を使用してロードされます。Natural セッション中にロードされたすべてのダイナミックサブプログラムは、LE 環境の終了時、つまり Natural セッションの終了時に削除されます。プロファイルパラメータ DELETE はまったく影響しません。

Natural による LE サブプログラムサポートの有効化

Natural から LE サブプログラムを呼び出すには、以下が必要です。

1. Natural のインストール時に、対応するドライバが LE370=YES オプションで生成される必要があります。CICS 環境の Natural での LE の有効化については、Natural の『TP モニタインターフェイス』ドキュメントの「CICS 環境での Natural」および「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。
2. リンケージエディタの手順中に、IBM LE ランタイムモジュールが IBM LE ライブラリから自動的に組み込まれる必要があります。「CEE」で始まる未解決の外部条件があってはなりません。リンケージエディタオプションの NCAL を z/OS に設定したり、NOAUTO を z/VSE に設定したりしないでください。
3. z/OS バッチ、IMS/TM、および TSO 環境でも Natural は LE メインプログラムを呼び出すことができますが、ダイナミックサブプログラムとしてのみ呼び出すことができます。LE メインプログラムをダイナミックに呼び出す場合は、CALL ステートメントの前に SET CONTROL 'P=L' を指定して、そのことを示す必要があります。そうしないと、Natural によって作成された LE 環境が LE メインプログラムによって終了します。

LE ランタイムオプションの引き渡し

- z/OS バッチおよび TSO 環境での LE ランタイムオプションの引き渡し
- z/VSE バッチ環境での LE ランタイムオプションの引き渡し
- IMS/TM 環境での LE ランタイムオプションの引き渡し

z/OS バッチおよび TSO 環境での LE ランタイムオプションの引き渡し

以下の 2 つのオプションがあります。

1. JCL で PARM= パラメータを使用して、LE ランタイムオプションを渡すことができます。以下のルールが適用されます。
 - メインルーチンに渡されるランタイムオプションは、その後にスラッシュ (/) を使用して Natural パラメータと分離する必要があります。
 - Natural パラメータ内でスラッシュを使用する場合は、Natural パラメータをスラッシュで始める必要があります。

例：

```
PARM='/ID=/,...'
```

2. JCL で CEEOPTS 入力データセットを使用して、LE ランタイムオプションを渡すことができます。CEEOPTS を使用すると、LE ランタイムオプションをすべてのサブタスクで使用できます。特に、バッチモードの Natural RPC サーバーでは CEEOPTS を使用する必要があります。

例：

```
//CEEOPTS DD *  
POSIX(ON)  
/*
```

z/VSE バッチ環境での LE ランタイムオプションの引き渡し

JCL で PARM= パラメータを使用して、LE ランタイムオプションを渡すことができます。以下のルールが適用されます。

- メインルーチンに渡されるランタイムオプションは、その後にスラッシュ (/) を使用して Natural パラメータと分離する必要があります。
- Natural パラメータ内でスラッシュを使用する場合は、Natural パラメータをスラッシュで始める必要があります。

例：

```
PARM='/ID=/,...'
```

IMS/TM 環境での LE ランタイムオプションの引き渡し

STEPLIB 連結でリージョン固有のランタイムオプションロードモジュール CEEROPT を使用して、LE ランタイムオプションを渡すことができます。その他に、LE ライブラリルーチン保存期限初期化ルーチン CEELRRIN が、リージョン JCL の PREINIT リストに含まれている必要があります。

以下に、AMODE(24) サブプログラムを実行できるようにする CEEROPT ロードモジュールの定義例を示します。

```
CEEROPT CSECT
CEEROPT AMODE ANY
CEEROPT RMODE ANY
          CEEXOPT ALL31=((OFF),OVR),                                X
          STACK=((128K,128K,BELOW,KEEP,512K,128K),OVR)
END      CEEROPT
```

LE アベンド処理

Natural では、LE 固有のユーザーエラー処理をサポートします。つまり、LE サブプログラムでユーザーエラーハンドラを定義している場合は、サブプログラムでアベンドやプログラムチェックなどの LE エラー状態が発生すると、そのハンドラが制御を取得します。LE ユーザーエラーハンドラを定義していない場合は、Natural は DU プロファイルパラメータの設定に従って対処します。

この場合は、特別なエラーメッセージ（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。また、対応する LE エラーメッセージが CEEMSG に発行され、LE ランタイムオプション TERMTHDACT に応じて LE スナップダンプが CEEDUMP に書き込まれます。



注意: DU=FORCE の場合は、Natural のアベンド処理は無効になり、アベンド時に LE サブプログラムがアクティブでなくても LE エラー処理が実行されます。このような場合は、LE ランタイムオプション TERMTHDACT(UAIMM) を指定して、必要なすべての診断情報を取得することを強くお勧めします。

77 外部 SORT

■ 外部 SORT のサポート	454
■ z/OS 固有の考慮事項	454
■ z/VSE 固有の考慮事項	454
■ BS2000/OSD 固有の考慮事項	455

このドキュメントでは、Natural による外部 SORT プログラムの使用について説明します。

以下のトピックについて説明します。

外部 SORT のサポート

Natural SORT ステートメントは、実際のソートを実行する外部 SORT プログラムをオプションで呼び出します。外部 SORT プログラムは、NTSORT マクロのキーワードサブパラメータ EXT が ON に設定されている場合に使用されます。

Natural では、SORT インターフェイスに準拠したすべての外部 SORT プログラムをサポートします。SORT インターフェイスについては、関連する IBM マニュアル (z/OS、z/VSE、および CMS) と Siemens マニュアル (BS2000/OSD) を参照してください。

要件 (スペースやデータセットなど) は、オペレーティングシステムの SORT プログラムを呼び出す 3GL (COBOL や PL/I など) アプリケーションプログラムの実行要件と同じで、使用中の外部 SORT プログラムによって異なる場合もあります。

外部 SORT プログラムとの通信には、E15 および E35 ユーザー出口ルーチンを使用します。そのため、データセット SORTIN および SORTOUT は不要です。

z/OS 固有の考慮事項

拡張パラメータリストをサポートするすべての外部 SORT プログラムを使用できます。

z/VSE 固有の考慮事項

外部 SORT プログラムは、パーティションプログラムエリアにロードされます。そのため、EXEC ステートメントの SIZE パラメータで指定される Natural バッチニュークリアスのサイズ要件に、約 200 KB の余分のストレージを追加する必要があります。

例：

```
// EXEC <natural>,SIZE(<natural>,200K)
```

<natural>は Natural フェーズの名前です。

BS2000/OSD 固有の考慮事項

外部 SORT プログラムは、レベル1のインターフェイスを使用して呼び出されます。つまり、Natural はすべての SORT 制御ステートメントを外部 SORT プログラムに渡すため、データセット SYSDTA は入力に使用されません。

外部 SORT プログラムは、以下のライブラリで検索されます。

- BLSLIB チェーンで連結されたユーザー TASKLIB (ユーザー TASKLIB を指定した場合)
- BLSLIB チェーンで連結されたシステム TASKLIB (\$TSOS.TASKLIB)

索引
