

Adabas

Adabas インストールマニュアル (z/OS)

バージョン 8.1.3

June 2008

This document applies to Adabas Version 8.1.3 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 1971-2008. 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 Adabas インストールマニュアル (z/OS)	1
2 サポート環境	3
3 z/OS 環境での Adabas のインストール	5
インストールチェックリスト	6
Adabas インストールの準備	7
リリーステープのインストール	9
Adabas コミュニケーション環境の初期化	11
Adabas データベースのインストール	23
SVC 保全性	29
クロスメモリサービスの要件	29
Global Resource Serialization の要件	31
EXCPVR の使用	31
共有可能な ADARUN の作成	32
16MB を超えるストレージ	33
2GB を超えるストレージ (64 ビット)	33
ZAP の適用	34
Adabas 7 ADALINK について	35
Adabas 8 ADALINK について	38
4 TP モニタを使用する場合の Adabas のインストール	41
IBM プラットフォーム用の Adabas リンクルーチンの準備	42
Adabas 8 配下での IMS/TM を使用する場合の Adabas のインストール	46
CICS を使用する場合の Adabas インストールの全般的な考慮事項	49
Adabas 8 配下での CICS を使用する場合の Adabas のインストール	51
Adabas 8 用 CICS ハイパフォーマンススタブルーチンのインストール	53
Adabas 8 配下での Com-plete を使用する場合の Adabas のインストール	78
バッチ/TSO を使用する場合の Adabas インストールの全般的な考慮事項	81
Adabas 8 配下での Batch/TSO を使用する場合の Adabas のインストール	82
Adabas データベース ID による Adabas SVC ルーティングの確立	85
バージョン 8 のソースメンバのデフォルト (LGBLSET マクロ) の変更点	94
5 UES 対応データベースの接続	109
概要	110
ダイレクト TCP/IP リンク経由の接続	111
TCP/IP リンクのアクティブ化	114
6 デバイスとファイルについて	117
サポートされるデバイスタイプ	118
ECKD デバイス	119
新規デバイスの追加	119
7 AOS デモバージョンのインストール	125
AOS デモのインストール手順	126
Natural Security を使用する場合の AOS のインストール	127
AOS デモバージョンデフォルトの設定	127
8 Recovery Aid (ADARAI) のインストール	131
ADARAI のインストール概要	132

ADARAI のインストール手順	132
9 エラー処理およびメッセージバッファリング機能のインストール	135
10 Adabas ダンプフォーマットングツール (ADAFDP)	137
ADAFDP 機能	138
ADAFDP 出力	138
11 変換テーブル	145
Adabas による EBCDIC から ASCII および ASCII から EBCDIC の変換	146
Entire Net-Work による EBCDIC から ASCII および ASCII から EBCDIC の変換	147
目次	149

1 Adabas インストールマニュアル (z/OS)

このドキュメントは、z/OS システムで Adabas インストールを計画または実行する方、および Adabas データベースシステムの管理または保守を行う方（データベース管理者やシステムプログラミング担当者など）を対象としています。

- サポート環境
- z/OS 環境での Adabas のインストール
- TP モニタを使用する場合の Adabas のインストール
- UES 対応データベースの接続
- デバイスとファイルについて
- AOS デモバージョンのインストール
- *Recovery Aid (ADARAI)* のインストール
- エラー処理およびメッセージバッファリング機能のインストール
- Adabas ダンプフォーマットングツール (ADAFDP)
- 変換テーブル

2 サポート環境

Adabas のインストールを実行する前に、ホストのオペレーティングシステムが最小要件を満たしていることを確認してください。

Adabas バージョン 8.1 は、次のオペレーティングシステム環境で使用できます。

■ 現在 IBM がサポートしているすべてのバージョンの z/OS

Software AG は、各製造元によってサポートされているオペレーティングシステムのバージョンに応じて Adabas のサポートを提供します。通常、製造元が特定バージョンのオペレーティングシステムに対するサポートを停止すると、Software AG もそのバージョンに対するサポートを停止します。

古いオペレーティングシステム上で新しいバージョンの Adabas を実行することは技術的には可能ですが、Software AG は、オペレーティングシステムの製造元によってサポートされていないバージョンに対して、サポートを継続しません。

サポートに関するご質問、または上記以外のオペレーティングシステムのリリース、バージョン、または種類に Adabas をインストールする場合のサポートの可否とサポート条件については、Adabas 技術サポートにお問い合わせください。

3 z/OS 環境での Adabas のインストール

▪ インストールチェックリスト	6
▪ Adabas インストールの準備	7
▪ リリーステープのインストール	9
▪ Adabas コミュニケーション環境の初期化	11
▪ Adabas データベースのインストール	23
▪ SVC 安全性	29
▪ クロスメモリサービスの要件	29
▪ Global Resource Serialization の要件	31
▪ EXCPVR の使用	31
▪ 共有可能な ADARUN の作成	32
▪ 16MB を超えるストレージ	33
▪ 2GB を超えるストレージ (64 ビット)	33
▪ ZAP の適用	34
▪ Adabas 7 ADALINK について	35
▪ Adabas 8 ADALINK について	38

インストールチェックリスト

z/OS システムに Adabas をインストールする手順の概要について説明します。

- !** **Important:** 提供されたすべての Adabas 8 のメンテナンスを提供時の状態のまま適用し、Adabas 8 パッチレベルライブラリ (L00n) を提供時の状態のまま連結したことを確認してください。これにより、Adabas 8 のコードは最新に保たれ、すべての Adabas 8 の機能の拡張とメンテナンスがサポートされます。

手順	説明	追加情報
	基本インストール手順	
1	Adabas ライブラリ用に DASD スペースを割り当てます。	ライブラリはインストールテープからリストアします。「 ライブラリのディスクスペース要件 」を参照してください。
2	Adabas データベース用に DASD スペースを割り当てます。	パフォーマンス向上のために、複数のデバイスおよびチャネルにデータベースファイルを分散します。「 データベースのディスクスペース要件 」を参照してください。
3	Adabas ニュークリアスを実行するためのアドレススペースを指定します。	「 Adabas ニュークリアスのアドレススペース要件 」を参照してください。
4	インストールテープから Adabas ライブラリをリストアします。	テープに添付されたテープ位置情報を使用します。「 リリーステープのインストール 」を参照してください。
5	一時的または永続的に AdabasSVC をインストールします。	「 Adabas コミュニケーション環境の初期化 」を参照してください。
	データベースのインストール手順	手順 6~15 では、セットアップ定義の変更が必要になります。詳細については、「 Adabas データベースのインストール 」を参照してください。
6	ADAFRM ユーティリティジョブで Adabas データベースを割り当て、フォーマットします。	
7	ADADEF ユーティリティジョブで、グローバルなデータベース属性を定義します。	
8	ADALODE、ADALODV、および ADALODM ジョブで、デモンストレーションファイルをロードします。	
9	製品ライセンスファイルをインストールします。	
10	ADANUC ジョブで、Adabas ニュークリアスを開始し、Adabas コミュニケーションをテストします。	
11	必要に応じて、CPEXLIST パラメータ付きの MULTI モードで ADAREP を実行し、Adabas アドレススペースコミュニケーションをテストします。	
12	必要に応じて、AOSINPL ジョブを実行し、Natural システムファイルに Adabas Online System (AOS) アドオン製品	

手順	説明	追加情報
	をロードします。または、AOS デモバージョンをインストールします（「 AOS デモバージョンのインストール 」参照）。	
13	OS の MODIFY コマンド (F) で ADAEND オペレータコマンドを使用して Adabas ニュークリアスを終了させます。	
14	ADASAV ユーティリティジョブを実行してデータベースをバックアップします。	
15	DEFAULTS ジョブを実行して ADARUN デフォルトを挿入します。	
	TP モニタのインストール	
16	Adabas に必要な TP リンクルーチンを実行してインストールします。	「 TP モニタを使用する場合の Adabas のインストール 」を参照してください。

Adabas インストールの準備

Adabas インストール準備の主要な手順は次のとおりです。

- システム設置条件が正しいかチェックします。
- ディスクおよびストレージスペースを割り当てます。

このsectionでは、次のトピックについて説明します。

- [ライブラリのディスクスペース要件](#)
- [UES サポートに必要なデータセット](#)
- [内部製品データセットのディスクスペース要件](#)
- [データベースのディスクスペース要件](#)
- [Adabas ニュークリアスのアドレススペース要件](#)

ライブラリのディスクスペース要件

Adabas ライブラリに最低限必要な 3390 ディスクスペースの要件は次のとおりです。

ライブラリ	3390 シリンダ	3390 トラック	ディレトリブロック
ロード	8	120	40
ソース	3	45	20
JCL	1	15	20



Note: ユーザープログラムを Adabas ロードライブラリから独立させるには、マルチユーザーモードでユーザープログラムを実行するために必要なモジュールと、ADAUSER でリンクされたモジュールだけを含むロードライブラリを別に作成します。バージョン 8.1

の場合、ユーザープログラムで必要となるモジュールは、ADAIOR、ADAIOS、ADALNK、ADAMLF、ADAPRF、ADARUN です。

UES サポートに必要なデータセット

ユニバーサルエンコーディングサービス (UES) をサポートするためのデータベースを有効にする場合は、Software AG 内部製品ライブラリ (APS - 移植プラットフォーム) が必要になります。これらのライブラリは、製品ライブラリとは独立して提供されます。

UES をサポートするには、次のライブラリをロードして、STEPLIB 連結に組み込む必要があります。ここで、*nn* はロードライブラリレベル、*vrs* はテープで配布されるそのコードの最新バージョンの番号です。

```
APSVrs.LDnn
```

高いレベル番号を持つライブラリで下位レベルのロードライブラリ (複数の場合あり) を完全に差し替えられない場合は、より高いレベルのライブラリは、steplib 連結で下位番号のライブラリの前に指定する必要があります。



Note: バージョン 7.2.2 より前の Adabas ロードライブラリを使用している場合、それは、下位バージョン番号の内部製品ライブラリを含んでいます。steplib 連結で、現在の内部製品ライブラリの下にこのライブラリを順序付ける必要があります。

さらに、UES サポートをするには、次のライブラリをロードして、セッション実行 JCL に組み込む必要もあります。

```
ADAvrs.ADAvrsCS
```

UES 対応データベースの接続のセットアップについては、「[UES 対応データベースの接続](#)」を参照してください。

内部製品データセットのディスクスペース要件

Adabas バージョン 8.1 で提供される内部製品ライブラリに必要な 3390 ディスクの最小ディスクスペースは次のとおりです (*vrs* はテープで提供されるコードの最新バージョンです)。

ライブラリ	3390 シリンダ	3390 トラック	ディレクトリブロック
ADAvrs.ADAvrsCS	12	180	150
APSVrs.LDnn	5	75	55

データベースのディスクスペース要件

Adabasに必要な実際のデータベースのスペースはユーザーの要件によって異なります。データベースに必要な最小の 3390 ディスクスペースは次のとおりです。

データベースコンポーネント	3390 シリンダ	3390 トラック
ASSOR1 (アソシエータ)	20	300
DATAR1 (データストレージ)	60	900
WORKR1 (ワークスペース)	15	225
TEMPR1 (テンポラリワークスペース)	15	225
SORTR1 (ソートワークスペース)	15	225

Adabas ニュークリアスのアドレススペース要件

標準的な Adabas ニュークリアスでは、動作させるために最低 800~1024KB のアドレススペースが必要です。ADARUNパラメータ設定によっては、必要なニュークリアスのアドレススペースのサイズが大きくなる場合があります。パラメータの設定はユーザーが決定します。

リリーステープのインストール

このセクションでは、次の作業の方法について説明します。

- データセット COPY.JOB をテープからディスクにコピーする方法
- ローカル命名規則に従ってデータセットを変更する方法

変更されたデータセットを使用して、すべてのデータセットをテープからディスクにコピーすることができます。その後、各コンポーネントに対して個別にインストール手順を実行する必要があります。



Note: SMA を使用している場合、*System Maintenance Aid* のマニュアルの SMA を使用する場合の Software AG 製品のインストールに関する章を参照してください。SMA を使用しない場合は、次の指示に従ってください。

- 手順 1a: データセット COPY.JOB をテープからディスクにコピーする (SMA ジョブ番号 T_{nnn})
- 手順 1b: COPY.JOB をローカル命名規則に変更する

■ 手順 1c : COPY.JOB をサブミットする

手順 1a : データセット COPY.JOB をテープからディスクにコピーする (SMA ジョブ番号 Tnnn)

データセット COPY.JOB (ラベル 2) には、その他すべての既存のデータセットをテープからディスクにアンロードするための JCL が含まれています。COPY.JOB をアンロードするには、次のサンプル JCL を使用します。

```
//SAGTAPE JOB SAG,CLASS=1,MSGCLASS=X
//* - - - - -
//COPY EXEC PGM=IEBGENER
//SYSUT1 DD DSN=COPY.JOB,
// DISP=(OLD,PASS),
// UNIT=(CASS,,DEFER),
// VOL=(,RETAIN,SER=<Tnnnnn>),
// LABEL=(2,SL)
//SYSUT2 DD DSN=<hilev>.COPY.JOB,
// DISP=(NEW,CATLG,DELETE),
// UNIT=3390,VOL=SER=<vvvvvv>,
// SPACE=(TRK,(1,1),RLSE),
// DCB=*.SYSUT1
//SYSPRINT DD SYSOUT=*
//SYSIN DD DUMMY
//
```

上記の意味は次のとおりです。

- <hilev> は有効なハイレベル修飾子です。
- <Tnnnnn> はテープ番号です。
- <vvvvvv> はボリューム通番 (VOLSER) の指定です。

手順 1b : COPY.JOB をローカル命名規則に変更する

このジョブをサブミットする前に、3つのパラメータの設定が必要です。

- HILEV に有効なハイレベル修飾子を設定します。
- LOCATION にストレージの場所を設定します。
- EXPDT に有効な失効日を設定します。

手順 1c : COPY.JOB をサブミットする

COPY.JOB をサブミットして、その他すべてのデータセットをテープからディスクにコピーします。

Adabas コミュニケーション環境の初期化

ここでは Adabas ルーター (ADASVC) のインストールについて説明します。ルーターは Adabas ニュークリアスと Adabas ユーザー間のコミュニケーションにクロスメモリサービスを使用します。

Adabas z/OS のクロスメモリコミュニケーションサービスは次の 2 つのモジュールから構成されています。

- Adabas ルーター (ADASVC)
- Adabas サブシステム初期化ルーチン (ADASIR)

ADASIR (IPL 時または Adabas SVC インストールプログラム (ADASIP) によって実行される) はルーターの操作環境、特に ID テーブルを初期化します。

ADASVC 一時的または永続的のどちらでもインストールできます。

- 一時的なインストールは ADASIP の実行によって行われ、次の IPL までの間だけ Adabas SVC が有効になります。



Note: 一度インストールすると、ADASIP REPLACE オプションを使用して Adabas SVC を一時的に再インストールすることができます。ただし、この手続きの間は Adabas ニュークリアスをアクティブにできません。



Note: SVC を初期化するために ADASIP を実行した後、CICS を再起動させる必要があります。

- Adabas SVC は、標準のオペレーティングシステムプロシージャを使用して、永続的にインストールされます。SVC をアクティブにするためには IPL が必要です。

通常、最初は ADASIP を使用して Adabas SVC を一時的にインストールします。これで、再度 IPL を実行しなくても、Adabas をすぐに使用できます。その間、次の IPL での永続的インストールのために準備します。

- Adabas リリース間の SVC の互換性の問題
- APF 認可の要件
- SVC テーブルエントリの割り当て
- サブシステム名の条件
- Adabas SVC のページ固定

- Adabas SVC の初期化
- ルーターのインストール概要
- 一時的インストールのための ADASIP の使用
- ADASIR の使用
- 一時インストール用の SVC の再リンク
- 永続的インストールのための SVC 再リンク

Adabas リリース間の SVC の互換性の問題

Adabas 8 には、新しい Adabas SVC が含まれています。この SVC は、下位互換性が完全に確保されています。つまり、新しい Adabas 8 SVC は、Adabas 7、またはそれ以前のデータベースで使用することができます。

ただし、Adabas 8 データベースを使用している以前の Adabas リリースから Adabas SVC を使用することは、できません。これを行おうとすると、Adabas 8 のデータベースは正常に初期化されません。

Adabas 8 SVC では、パフォーマンスが向上され、エラーリカバリルーチンが改善されています。新しい SVC では効率的なオペレーティングシステムインターフェイスが使用されており、特にコマンド完了時のユーザーへの通知が改善されています。これにより、SRB モードのルーチンから TCB モードのルーチンに、またユーザープログラムと Adabas ニュークリアスの間でワークが切り替えられます。Adabas 8 SVC のパフォーマンスを分析する際は、この点を考慮に入れてください。新しい SVC では、SRB モードのオーバーヘッドが大幅に削減され、TCB モードのオーバーヘッドが若干増加しましたが、結果的には全体的な SVC のパフォーマンスは向上しています。

APF 認可の要件

Adabas 8 SVC では、Adabas ニュークリアスおよびその他の MPM サーバー（Entire Net-Work、Natural グローバルバッファプールなど）が APF 認可されている必要があります。この APF 認可により、ADASVC0 コール不正使用を防止します。Software AG では、セキュリティリスクを軽減するため、APF 認可の実行を強くお勧めします。ただし、Software AG では、ご要望に応じてこの要件を省略する ZAP（AY811031）を提供しています。



Note: 一部のアドオン製品では、制限付き z/OS サービスを使用するために APF 認可が必要となる場合があります。このような場合でも、APF 認可が必要です。

SVC テーブルエントリの割り当て

選択したインストール手順に関係なく、有効な SVC テーブルエントリを Adabas ルーター (ADASVC) に割り当てる必要があります。SVC テーブルエントリは SYS1.PARMLIB のメンバ IEASVCxx に定義します。

オペレーティングシステムの SVC テーブルエントリは、ADASVC に対応して次の情報を含んでいる必要があります。

オフセット	ラベル	説明
0	SVCEP	SVC エントリポイントアドレス。
4	SVCATTR1	タイプ 2 SVC (フラグビット SVCTP2 に X'80' を設定)、もしくはタイプ 3 または 4 SVC (フラグビット SVCTP34 に X'C0' を設定) を指定する必要があります。ADASIR は、タイプ 1、5、6 のいずれかの SVC をタイプ 2 に変更します。 この SVC に APF 認可が必要であることを設定できます (フラグビット SVCAPF に X'08' を設定)。設定した場合、すべてのターゲットおよびユーザーの APF 認可が義務付けられます。
6	SVCLOCKS	すべて 0 にする必要があります。ADASIR によって SVCLOCKS がゼロに設定されます。

サブシステム名の条件

ADASVC のオフセット X'28' の 4 文字フィールド SUBSYS に含まれるサブシステム名 (デフォルトは ADAB) は SYS1.PARMLIB のメンバ IEFSSN_{xx} に指定された名前と必ず同じにする必要があります。異なっていると、ADASIR は ADAS12 メッセージとコンディションコード 2 で終了し、Adabas は使用不可になります。

Adabas SVC のページ固定

Adabas SVC を固定 LPA に配置する場合、SYS1.PARMLIB のメンバ IEAFIX_{xxx} にエントリを追加してください。

Adabas SVC の初期化

Adabas の全ニュークリアスを完全に機能させるには、Adabas SVC を ADASIP または ADASIR によって初期設定しておく必要があります。

ルーターのインストール概要

- 一時ルーターのインストール (SMA ジョブ番号 I011)
- 永続ルーターのインストール (SMA ジョブ番号 I010)

一時ルーターのインストール (SMA ジョブ番号 I011)

Adabas インストールテープをリストアしてから、ローカルエディタを使用してジョブ JCLLINK (ADASIR、ADASIP、ADASVC のリンクに使用) を次のように修正してください。

▶ **手順 3.1. 一時ルーターをインストールするには、次の手順に従います。**

- 1 ADASIP を認可モジュールとして APF 認可ライブラリにリンクします。
- 2 ADASIR および ADASVC を APF 認可ライブラリにリンクします。
 - ADASIP を実行するために APF 認可ライブラリに ADASVC を挿入します。
 - SYS1.LINKLIB に連結された APF 認可ライブラリに ADASIR を挿入します。SYS1.LINKLIB は、SYS1.PARMLIB にあるソースメンバ LNKLST_{xx} で定義されています。
- 3 ADASIP を実行して SVC をインストールします。

IPL を行わずに Adabas SVC をダイナミックに追加するには、ADASIP ジョブを修正して実行します。

永続ルーターのインストール (SMA ジョブ番号 I010)

▶ **手順 3.2. 永続ルーターをインストールするには、次の手順に従います。**

- 1 SVC ルーチンの名前変更規則 (例: タイプ 3 SVC は IGC00_{nnn} という名前にする。 *nnn* は符号付き 10 進数の SVC 番号) に従って名前を変更された Adabas SVC (ADASVC) を ADASIR に対する永続的なステップとして SYS1.LPALIB にリンクします。
- 2 ADASIR を SYS1.LINKLIB にリンクします。または SYS1.PARMLIB の LNKLST_{xx} メンバで SYS1.LINKLIB に連結された APF 認可ライブラリにリンクします。



Note: ADASIR はリエントラントではありません。したがって、SYS1.LPALIB にリンクしないでください。

- 3 ジョブ JCLUPDT を修正して実行し、正しいフォーマットで新しいエントリを追加します。
- 4 CLPA オプションを指定して z/OS の IPL を実行し、Adabas コミュニケーション環境をインストールして初期化します。

一時的インストールのための ADASIP の使用

- [ADASIP 機能](#)
- [ADASIP パラメータ](#)
- [ADASIP の実行](#)

ADASIP 機能

ADASIP は次の機能を実行します。

- 指定された CSA サブプールで、Adabas SVC およびサブシステムコミュニケーションベクターテーブル (SSCT) 用の CSA メモリを取得します。
- 取得した CSA スペースに Adabas SVC をロードします。
- 指定された Adabas SVC の SVC テーブルエントリを変更します。
- 同じサブシステム名の SSCT を SSCT チェーンから削除します (オプション)。
- 新しい SSCT を SSCT チェーンに追加します。
- ADASIR プログラムを実行します。
- 以前にインストールされた SVC によって取得された CSA を解放します。

何らかのエラーが検出された場合、ADASIP は、完了したすべてのアクティビティをバックアウトし、そのエラーを示すユーザーアベンドで終了します。

現在 ADASVC によって使用されている SVC 番号を使用して、ADASVC のインスタンスを再インストールするときは、サブシステム名を現在使用中のものと同じにする必要があります。これにより、正常に動作しないコンフィグレーションを避けることができます。詳細については、「[SVC 保全性](#)」を参照してください。

バージョン 8 の ADASIP/ADASIR を使用して、バージョン 7 の Adabas SVC をインストールすることができますが、バージョン 8 の SVC をインストールするには、バージョン 8 の ADASIP/ADASIR が必要です。バージョン 8 を使用して、すべてのサポートされる SVC のリリースをインストールすることをお勧めします。以前のバージョンの ADASIP/ADASIR を使用して、後のバージョンの SVC を置き換えた場合、共通ストレージの一部のエリアが解放されない可能性があります。

次の JCL によって、ADABAS.Vvrs.LOAD の ADASIP が APF 認可ライブラリに認可モジュールとしてリンクされます。

```
//LNKSIP EXEC PGM=IEWL
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD SPACE=(CYL,(1,1)),UNIT=SYSDA
//ADALIB DD DSN=ADABAS.Vvrs.LOAD,DISP=SHR
//SYSLMOD DD DSN=apflibname,DISP=SHR
//SYSLIN DD *
INCLUDE ADALIB(ADASIP)
SETCODE AC(1)
NAME ADASIP(R)
```

ADASIP パラメータ

ADASIP パラメータの構文は次のとおりです。

```
CONSNAME=c, IDTSPL=i, LEAVE=l, NRIDTES=n, REPLACE=r, SUBSYS=su,
SVCNR=svcn, SVCSP=svcs
```

上記の意味は次のとおりです。

<i>c</i>	オペレータメッセージが書き込まれるコンソールの名前です。省略した場合、メッセージは、ROUTCDE=2、マスターコンソール情報を使用して発行されます。
<i>i</i>	ID テーブルサブプールです。詳細については、ADASIR IDTSPL パラメータを参照してください。
<i>l</i>	ADASIR がメッセージ ADAS11 または ADAS12 をオペレータコンソールに表示するかどうかを示します。詳細については、ADASIR LEAVE パラメータを参照してください。
<i>n</i>	ID テーブルエントリの数です。詳細については、ADASIR NRIDTES パラメータを参照してください。
<i>r</i>	同じサブシステム名の既存の SSCT を置き換えるかどうかを示します。Y (YES) または N (NO) を指定します (デフォルトは N)。どのようなタイプの Adabas SVC の置換にもこのオプションを使用します (例: 新規 SVC バージョンをインストール時)。
<i>su</i>	サブシステム名。このパラメータは必須です。Adabas SVC の各インスタンスは固有のサブシステム名を持っている必要があります。
<i>svcn</i>	Adabas SVC 番号です。詳細については、ADASIR SVCNR パラメータを参照してください。
<i>svcs</i>	Adabas SVC および SSCT サブプールです。固定 CSA の場合は 228、ページング可能な CSA の場合は 241 です (デフォルトは 241)。

次の表に有効な ADASIP パラメータの省略形を示します。

パラメータ	省略形
CONSNAME=	C=
IDTSPL=	I=
LEAVE=	L=
NRIDTES=	N=
REPLACE=	R=
SUBSYS=	SU=
SVCNR=	SVCN=
SVCSPL=	SVCS=

SUBSYS と SVCNR 以外のパラメータはすべてオプションです。指定した場合、パラメータ IDTSPL、LEAVE、NRIDTES、SUBSYS、SVCNR は検証されずに、ADASIR に渡されます。

ADASIP の実行

次のような JCL で ADASIP を実行します。

```
// EXEC PGM=ADASIP,PARM=parameters
//STEPLIB DD ...
//SVCLIB DD ...
//SIRLIB DD ...
```

STEPLIB DD ステートメントで定義するデータセットは、APF 認可プログラム ADASIP を含む APF 認可ライブラリにする必要があります。ADASIP はリエントラントでもリフレッシュ属性でもないので、データセットを SYS1.LPALIB にすることはできません。

SVCLIB DD ステートメントで定義するデータセットには、Adabas SVC（その名前または別名 ADASVC）を含む APF 認可ライブラリを指定する必要があります。

SIRLIB DD ステートメントで定義するデータセットは ADASIR プログラムを含んでいる必要があります。ADASIR はリエントラントでもリフレッシュ属性でもないので、データセットを SYS1.LPALIB にすることはできません。

パラメータの入力に誤りがある場合、ADA SIP は U0481 アベンドで終了します。

IBM ジョブコントロール規則では PARM パラメータの継続は次のように指定します。

```
// EXEC PGM=ADASIP,PARM=('parameters ....', X  
// 'parameters')
```

72 カラム目の X は継続文字です。次の制限も、JCL ステートメントに適用されます。

- 継続する行の終了引用符の後にコンマが必要です。
- 継続する各行の 72 カラム目に空白以外の文字が必要です。また、継続行は 4~16 カラムで開始する必要があります。
- 継続する行の最後のパラメータと終了引用符の間にコンマは指定できません。これは、継続文字列を結合するときに、JCL が自動的にパラメータ間にコンマを挿入するためです。

```
// ...PARM=('CONSID=3', X  
// 'SUBSYS=ADAB', X  
// 'SVCNR=249')
```

上記の結果は次と同等です。

```
CONSID=3,SUBSYS=ADAB,SVCNR=249
```

ADASIR の使用

- [ADASIR 機能](#)
- [ADASIR の再リンク](#)
- [ADASIR パラメータ](#)
- [ADASIR の実行](#)

ADASIR 機能

ADASIR プログラムは次のいずれかによって起動されます。

- ADASIP プログラムで起動して、Adabas SVC を一時的にインストールします。
- z/OS で起動して、Adabas SVC を永続的にインストールします。

ADASIR はマスタスケジューラの初期化中または ADASIP 実行中にコントロールを受け取りません。ZAP またはアセンブルに無効な値があれば、確認メッセージが表示されます。(特定のメッセージについては『*Adabas メッセージおよびコードマニュアル*』を参照してください)。IEFSSN_{xx} メンバに指定されたパラメータまたは ADASIP によって渡されたパラメータの処理中にエラーが見つかった場合、オペレータはすべての値の確認メッセージが表示されます。

SVC テーブルエントリが無効な場合、ADASIR によってエントリの変更の承認を求めるメッセージが表示されます (SVCTAB=P (デフォルト) が指定された場合)。エラーが見つかった場合、

必ず修正し、Adabas SVC を使用する前に、IPL を実行するか、ADASIP を再実行する必要があります。

ADASIR の再リンク

ADASIR モジュールは APF 認可ライブラリにリンクする必要があります。

次の JCL は ADABAS.Vvrs.LOAD の ADASIR を SYS1.LINKLIB にリンクします。

```
//LNKSIR EXEC PGM=IEWL
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD SPACE=(CYL,(1,1)),UNIT=3350
//ADALIB DD DSN=ADABAS.Vvrs.LOAD,DISP=SHR
//SYSLMOD DD DSN=SYS1.LINKLIB,DISP=SHR
//SYSLIN DD *
INCLUDE ADALIB(ADASIR)
NAME ADASIR(R)
```

ADASIR パラメータ

ADASIR パラメータの構文は次のとおりです。

```
IDTSPL=i, LEAVE=l, NRIDTES=n, SVCNR=svcn, SVCTAB=svct
```

変数	説明
<i>i</i>	ID テーブルサブプールです。固定 CSA の場合は 228、ページング可能な CSA の場合は 241 です（デフォルトは 241）。
<i>l</i>	メッセージ ADAS11 または ADAS12 をオペレータコンソールに表示するかどうかを示します。Y（YES）または N（NO）を指定します（デフォルトは N）。
<i>n</i>	ID テーブルエントリカウントです。範囲は 1 から最大値の z/OS ニュークリアスの CSECT IEAVESVT のオフセット X'146' で定義された値までです（「 クロスメモリサービスの要件 」参照）。
<i>svcn</i>	Adabas SVC 番号（200～255）
<i>svct</i>	オペレータに SVC テーブルエントリ更新の承認を要求するかどうかを指定します。要求する場合は P（デフォルト）、要求しない場合は N を指定します。SVC テーブルエントリが ADASIR の予期しないものになる可能性がある場合、P を指定してください。

次の表に有効な ADASIR パラメータの省略形を示します。

パラメータ	省略形
IDTSPL=	I=
LEAVE=	L=
NRIDTES=	N=
SVCNR=	SVCN=
SVCSPL=	SVCS=

ADASIR の実行



Note: ADASIR モジュールは APF 認可ライブラリにリンクする必要があります。

SVC の永続的インストール準備のため、SYS1.PARMLIB の新規メンバまたは既存のメンバ IEFSSN_{xx} にエントリを作成する必要があります。このエントリは 80 文字のレコードで、次のような形式です。

```
SUBSYS SUBNAME(cccc) CONSNAME(consname) INITRTN(ADASIR)
INITPARM('parameters') comments
```

上記の意味は次のとおりです。

<i>cccc</i>	1~4 文字のサブシステム名。この名前は Adabas SVC のオフセット X'28' に指定された名前と同じにする必要があります。SVC に指定された名前は ADAB です。他の名前の場合、 <i>cccc</i> を指定する前に最初に SVC に ZAP する必要があります。
<i>consname</i>	ADASIR がメッセージを表示するコンソールの名前です。省略した場合、メッセージは、ROUTCDE=2、マスターコンソール情報を使用して発行されます。
<i>'parameters'</i>	ADASIR パラメータ複数のパラメータを指定する場合、一重の引用符で囲み、パラメータ間にコンマを挿入する必要があります。
<i>comments</i>	コメントはオプションですが、前に最低 1 つの空白を挿入する必要があります。

サブシステム名が一致しない場合、ADASIR は ADAS12 メッセージとコンディションコード 2 でアペンドします。Adabas z/OS コミュニケーション環境は初期化されません。必要に応じて、SSN=xx を指定して、z/OS の IPL を再実行します。これがタイプ 3 または 4 の Adabas SVC で最初の IPL の場合、SET パラメータの 1 つとして CLPA を指定します。

IEFSSN_{xx} から取得したパラメータまたは ADASIP から渡されたパラメータを処理しているときにエラーになった場合（メッセージ ADAS05）、オペレータはパラメータをすべて再入力するように要求されます。そのとき、SVC テーブルエントリが誤っている場合（メッセージ ADAS09）、SVCTAB パラメータの値によって、オペレータは SVCTAB エントリ値を変更する許可を求められる（メッセージ ADAS10）か、単に変更されます（メッセージ ADAS15）。

バージョン 8 の ADASIR を使用して、バージョン 7 の Adabas SVC をインストールすることができますが、バージョン 8 の SVC をインストールするには、バージョン 8 の ADASIR が必要です。バージョン 8 を使用して、すべてのサポートされる SVC のリリースをインストールすることをお勧めします。

ADASIR のメッセージおよびその意味については『*Adabas* メッセージおよびコードマニュアル』を参照してください。

一時インストール用の SVC の再リンク

Adabas SVC をその名前または別名 ADASVC で APF 認可ライブラリにリンクします。ADASVC を AMODE=31 および RMODE=24 (デフォルト) でリンクする必要があります。

次に、SVC をリンクする方法の例を示します。

```
// (job card)
//LKED EXEC PGM=IEWL,
// PARM='XREF,LIST,NCAL,LET,MAP,RENT,REFR,REUS'
//SYSPRINT DD SYSOUT=X
//SYSUT1 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSLMOD DD DSN=SYS1.LINKLIB,DISP=SHR ---Target loadlib
//ADALIB DD DSN=user.loadlib,DISP=SHR ---ADASVC loadlib
//SYSLIN DD *
MODE AMODE(31),RMODE(24)
INCLUDE ADALIB(ADASVC)
NAME ADASVC(R)
/*
```



Note: 永続的インストールの準備をしているときに、ADASVC 以外の名前で SVC をリンクする場合、SVC の別名は必ず ADASVC にします。Adabas SVC をダイナミックにロードするとき、ADASIP は SVCLIB DD ステートメントに指定されたライブラリのモジュール名 ADASVC を検索します。

永続的インストールのための SVC 再リンク

Adabas SVC にタイプ 3 または 4 の SVC を使用することをお勧めします。

SVC タイプ 1 および 6 はサポートされていません。

- タイプ 2 の SVC

■ タイプ 3 または 4 の SVC

タイプ 2 の SVC

Adabas SVC をタイプ 2 にする場合、システムニュークリアス IEANUC0x として SVC を SYS1.NUCLEUS にリンクします。

このニュークリアスには、有効にしたタイプ 2 SVC の SVC テーブルエントリが含まれている必要があります。これは SYSGEN 時に定義する必要があります。

ニュークリアスのリンクに必要なステートメントに次のようなリンケージエディタコントロールステートメントを追加します。

```
CHANGE ADASVC(IGCnnn) ---> nnn is the SVC number in decimal
INCLUDE ADALIB(ADASVC) ---> ADALIB contains the Adabas SVC
```

タイプ 3 または 4 の SVC

Adabas SVC をタイプ 3 または 4 としてインストールするには、Adabas SVC を適切な名前ですYS1.LPALIB にリンクします。ADASVC は AMODE=31 および RMODE=24 (デフォルト) でリンクする必要があります。

次に、SVC を再リンクする方法の例を示します。

```
// (job card)
//LKED EXEC PGM=IEWL,
// PARM='XREF,LIST,NCAL,LET,MAP,RENT,REFR,REUS'
//SYSPRINT DD SYSOUT=X
//SYSUT1 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSLMOD DD DSN=SYS1.LPALIB,DISP=SHR <--Target Loadlib
//ADALIB DD DSN=ADABAS.Vvrs.LOAD,DISP=SHR <--ADASVC loadlib
//SYSLIN DD *
MODE AMODE(31),RMODE(24)
CHANGE ADASVC(IGC00nnp) <- where nn are the first two digits of the SVC in decimal and
INCLUDE ADALIB(ADASVC) p is the character corresponding to the x'Cn'
NAME IGC00nnp(R) ----> (n is the last digit of the SVC number in decimal)
```

*
/*

Adabas データベースのインストール

Adabas インストールテープをリストアし、ADASVC をインストールした後に、次のいずれかを実行できます。

- 既存の Adabas データベースを新しいバージョンに移行します。
- 新しいバージョンの Adabas データベースをインストールします。

インストール中に発行されたメッセージやコードについては『Adabas メッセージおよびコードマニュアル』を参照してください。ユーティリティの詳細については、『Adabas ユーティリティマニュアル』を参照してください。

- [既存のデータベースの移行](#)
- [データベースのインストール手順](#)

既存のデータベースの移行

既存のデータベースを Adabas の新しいリリースに移行するには、ADACNV ユーティリティを使用します (SMA ジョブ番号 I051)。詳細については、『Adabas ユーティリティマニュアル』を参照してください。

データベースのインストール手順

- 手順 1: Adabas データベースの割り当てとフォーマット (SMA ジョブ番号 I030)。
- 手順 2: グローバルなデータベース属性の定義 (SMA ジョブ番号 I030)
- 手順 3: デモンストレーションファイルのロード (SMA ジョブ番号 I050)
- 手順 4: 製品ライセンスファイルのインストール (SMA ジョブ番号 I007)
- 手順 5: Adabas ニュークリアスの開始および Adabas コミュニケーションのテスト (SMA ジョブ番号 I040)
- 手順 6: Adabas アドレススペースコミュニケーションのテスト (該当する場合)
- 手順 7: Adabas Online System アドオン製品のロード (該当する場合) (SMA ジョブ番号 I061)
- 手順 8: Adabas ニュークリアスの終了
- 手順 9: データベースのバックアップ
- 手順 10: ADARUN デフォルトの挿入

■ **手順 11：Adabas に必要な TP リンクルーチンのインストール**

手順 1：Adabas データベースの割り当てとフォーマット（SMA ジョブ番号 I030）。

ADAFRM ユーティリティジョブを修正して実行し、Adabas データベースの割り当てとフォーマットを行います。次の項目を修正する必要があります。

- データベースとライブラリのデータセット名
- ライブラリのボリュームとデータベースのデータセット
- データベースのデータセットのスペース割り当て
- Adabas SVC 番号、データベース ID、データベースデバイスタイプ
- 各 ADAFRM ステートメントのデータセットのサイズ

手順 2：グローバルなデータベース属性の定義（SMA ジョブ番号 I030）

ADADEF ユーティリティジョブを修正して実行し、データベースのグローバルな属性を定義します。次の項目を修正する必要があります。

- データベースとライブラリのデータセット名
- Adabas SVC 番号、データベース ID、データベースデバイスタイプ
- ADADEF パラメータ

手順 3：デモンストレーションファイルのロード（SMA ジョブ番号 I050）

次のジョブを修正し、実行します。

- ADALODE：サンプルデモファイル EMPL をロードします。
- ADALODV：サンプルデモファイル VEHI をロードします。
- ADALODM：サンプルデモファイル MISC をロードします。

各ジョブについて、次の項目を修正する必要があります。

- データベースとライブラリのデータセット名
- Adabas SVC 番号、データベース ID、データベースデバイスタイプ
- ADALOD パラメータ

手順 4：製品ライセンスファイルのインストール（SMA ジョブ番号 I007）

製品ライセンスファイルは個別のカスタマイズインストールテープ、または別に送信される電子メールの添付ファイルで提供されます。ライセンスファイルがインストールテープで提供されている場合は、この手順の最後あたりの指示に従ってライセンスファイルをインストールしてください。ライセンスファイルが電子メールの添付ファイルとして提供されている場合、次の指示に従ってライセンスを z/OS に転送する必要があります。

ライセンスファイルの z/OS への転送

製品ライセンスファイルの電子メール添付ファイルを Windows から z/OS に転送するには、次の手順に従います。

1. 製品ライセンスファイルの電子メール添付ファイルをハードドライブの任意の場所に保存します。
2. [コマンドプロンプト] ウィンドウを開きます。
3. [コマンドプロンプト] ウィンドウで、製品ライセンスファイルを保存したディレクトリに移動します。
4. z/OS ホストとの間で FTP セッションを開始します。

```
ftp host-name
```

host-name は z/OS ホストの名前です。

5. z/OS ホストのログイン ID とパスワードを入力します。
6. FTP セッションが確立されたら、ライセンスファイルの z/OS ファイルシステムとデータセットの接頭辞 ("ADABAS" などの使いやすい接頭辞) を指定します。

```
ftp>cd 'ADABAS'
```

7. バイナリデータモードに切り替えます。転送中、ライセンスファイルのフォーマットは ASCII のまま保持される必要があります。

```
ftp>binary
```

8. ライセンスファイルのデータセットが RECFM=FB および LRECL=80 で書き込まれるように指定します。

```
ftp>quote site RECFM=FB LRECL=80 BLKSIZE=4000
```

製品ライセンスファイルでは RECFM=F もサポートされています。

9. ライセンスをデータセットとして z/OS システムに書き込みます。例えば、ライセンスファイル名が ada81.xml の場合、次のように入力します。

```
ftp>put ada81.xml
```

このコマンドによって、"*prefix*.ADA81.XML" という名前のデータセットが作成されます。*prefix* は手順 6 で指定したデータセット接頭辞です。手順 6 で接頭辞として "Adabas" を指定した場合、データセット名は "ADABAS.ADA81.XML" になります。データセットは RECFM=FB、LRECL=80 になり、ライセンス情報は ASCII フォーマットでそのデータセットに格納されます。

10. FTP セッションを終了します。

```
ftp>quit
```

ライセンスファイルのインストール

製品ライセンスファイルをインストールするには、次の手順に従います。

1. ライセンスファイルは Adabas ソースライブラリ上に格納されていることを確認してください (RECFM=F または FB および LRECL=80)。また、フォーマットは ASCII が保持されるように注意してください。
2. サンプルジョブ ASMLICAM を確認して、次のように変更します。
 - STEPLIB DD ステートメントが Adabas ロードライブラリを指すように変更します。
 - SYSUT1 DD ステートメントが Adabas ライセンスファイルを含むデータセットを指すように変更します。
 - L.SYSLMOD DD ステートメントの適切なユーザーロードライブラリを指定します。



Note: このユーザーロードライブラリは、Adabas ニュークリアス (ADANUC) の STEPLIB 連結にも組み込まれている必要があります。

3. サンプルジョブ ASMLICAM をサブミットします。このジョブは、プログラム ADALICAM を実行し、ライセンステキストファイルを ADALIC という名前のロードモジュールに変換します。このモジュールは、指定されたユーザーロードライブラリに格納され、Adabas ニュークリアスによってロードされます。

または、製品ライセンスファイルをシーケンシャルファイルまたは PDS メンバから直接読み込むこともできます。これを行うには、次の手順に従います。

1. ライセンスファイルは Adabas ソースライブラリ上に格納されていることを確認してください (RECFM=F または FB および LRECL=80)。また、フォーマットは ASCII が保持されるように注意してください。
2. 以前に作成した ADALIC ロードモジュールから、ニュークリアスジョブによって使用される Adabas ロードライブラリにアクセスできないことを確認します。Adabas は最初に ADALIC のロードを試み、失敗すると DDLIC から読み込みます。

3. すべての Adabas ニュークリアス起動ジョブに次の DD ステートメントを追加します。

```
//DDLIC DD DISP=SHR,DSN=ddsn
```

d^{dsn} は ASCII フォーマットのライセンスファイルのデータセット名です。*d^{dsn}* はパーティション分割されたデータセットのメンバを参照することがある点に注意してください。

手順 5：Adabas ニュークリアスの開始および Adabas コミュニケーションのテスト（SMA ジョブ番号 I040）

ジョブ ADANUC を修正して実行し、Adabas ニュークリアスを開始します。次の項目を修正する必要があります。

- データベースとライブラリのデータセット名



Note: 適切なユーザーロードライブラリ、特に ADALIC ライセンスロードモジュールを含むライブラリを、必ず組み込んでください。

- Adabas SVC 番号、データベース ID、データベースデバイスタイプ
- ADANUC パラメータ

手順 6：Adabas アドレススペースコミュニケーションのテスト（該当する場合）

ADAREP を修正し、CPEXLIST パラメータを使用して MULTI モードで実行し、Adabas アドレススペースコミュニケーションをテストします。次の項目を修正する必要があります。

- データベースとライブラリのデータセット名
- Adabas SVC 番号、データベース ID、データベースデバイスタイプ
- ADAREP パラメータ

手順 7：Adabas Online System アドオン製品のロード（該当する場合）（SMA ジョブ番号 I061）

ジョブ AOSINPL を修正して実行し、バージョン Natural 3.1 以上のバッチを使用した Natural システムファイルに Adabas Online System (AOS) をロードします。次の項目を修正する必要があります。

- データベースとライブラリのデータセット名
- Adabas SVC 番号、データベース ID、デバイスタイプ
- Natural INPL パラメータとシステムファイル番号

または、Adabas 付属の AOS デモバージョンをインストールします。「[AOS デモバージョンのインストール](#)」を参照してください。

手順 8：Adabas ニュークリアスの終了

次の OS MODIFY コマンドで ADAEND オペレータコマンドを使用して Adabas ニュークリアスとコミュニケーションし、セッションを終了します。

```
F jobname ,ADAEND
```

または次のコマンドで終了します。

```
P jobname
```

jobname は、開始したニュークリアスのジョブまたはタスク名です。

手順 9：データベースのバックアップ

データベースをバックアップするには、ADASAV ユーティリティを修正し、実行します。次の項目を修正する必要があります。

- データベースとライブラリのデータセット名
- Adabas SVC 番号、データベース ID、デバイスタイプ
- ADASAV パラメータ

手順 10：ADARUN デフォルトの挿入

Adabas JCL ライブラリのメンバ DEFAULTS を変更して、ADARUN デフォルトを設定できます。次の項目を修正する必要があります。

- データベースとライブラリのデータセット名
- 次の項目の ADARUN ユーザーデフォルト
 - デバイスタイプ（デフォルト：3390）
 - SVC 番号（デフォルト：249）
 - データベース ID（デフォルト：1）

DEFAULTS ジョブを修正して実行し、OS ZAP ユーティリティを使用して ADARUN デフォルトを設定します。

手順 11：Adabas に必要な TP リンクルーチンのインストール

TP リンクルーチンプロシージャの詳細については、「[TP モニタを使用する場合の Adabas のインストール](#)」を参照してください。

SVC 保全性

かつては、同一のサブシステム ID が複数の SVC に存在すると、複数の SVC によって単一の ID テーブルが使用される結果になりました。これは問題を引き起こし、場合によっては重大な問題（ニュークリアスの異常終了またはデータベースの破壊）が発生することもありました。

この危険性を避けるため、バージョン 8 の SVC では、ID テーブルにアクセスする SVC が、ADASIP/ADASIR によってテーブルの初期化に使用された SVC と同じであるかチェックします。SVC が同じでないとき、アベンド 650 が発生します。

アベンド 650 はニュークリアスの ADARUN パラメータに無効な SVC 番号が指定された場合に発生し、Adabas 初期化中、ユーザープログラムから最初の Adabas コールを発行したとき、または Entire Net-Work のような他の Software AG 製品によって ID テーブルが要求されたときに発生することがあります。

ADASIP はこの問題が発生しないように機能が拡張されています。現在他のサブシステムに関連付けられている SVC 番号を使用して ADASVC のインスタンスをインストールしようとすると、ADASIP はアベンド 435 で終了します。

クロスメモリサービスの要件

z/OS におけるクロスメモリサービスの実装のため、MULTI モードで Adabas ニュークリアスを実行するときは次の点に注意してください。

- 最大 1 ステップのジョブでクロスメモリ環境を確立できます。これは、ジョブにはターゲットとして最大 1 ステップを組み込めるといことです（例：Adabas ニュークリアス）。
- クロスメモリアクセスはスワップアウトしたアドレススペースにアクセスできません。したがって、ニュークリアスセッションの間、Adabas ニュークリアスのアドレススペースは NON SWAPPABLE に設定されます。これにより、実ストレージの必要量が増加します。この動作については、IBM マニュアル『*Extended Addressability Guide*』の「Synchronous Cross-Memory Communication」を参照してください。
- アクティブクロスメモリ環境のニュークリアスが終了すると、正常終了か異常終了かに関わらず、イニシエータを含むアドレススペース全体も終了します。

このアドレススペースを表す ASID は次の IPL まで再び割り当てられることはありません。したがって、SYS1.PARMLIB のアクティブ IEASYS_{xx} メンバの MAXUSERS パラメータには十分大きい値を指定してください。または、システムがサポートする場合、同じメンバの RSVNONR パラ

メータを適宜調整できます。また、Adabas ニュークリアスを理由もなく停止したり起動したりしないでください。

この点については、上記のマニュアルの「Recovery Considerations」および「Resource Management」を参照してください。IBM APAR の OZ61154、OZ61741、OZ67637 も参照してください。

クロスメモリサービスをシステム内のすべてのアドレススペースから利用できるようにするため、Adabas ニュークリアスはシステムリンケージインデックス (LX) を z/OS から取得します。LX は全アドレススペースのリンケージスペースで予約済みのスロットで、システム全体の全アドレススペースへのリンケージを許可します。

コンフィグレーションで z/OS 1.6 以降を使用していて、ハードウェアがアドレススペースと LX の再利用機能 (ALRF) を使用している場合、バージョン 8 の ADASVC は再利用可能なシステム LX を使用します。ALRF を使用していない場合、前リリースと同様、再利用不可のシステム LX が使用されます。再利用可能な LX では、再利用不可の LX による制限が解決されます。次に、これらの制限について説明します。

システムで使用するために z/OS が確保できる再利用不可の LX の数はあまり多くありません (最大 2048、通常は 165)。

z/OS がクロスメモリサービスを使用する方法が原因で、Adabas が取得した再利用不可のシステム LX は、次の IPL まで z/OS の制御に返すことができません。しかし、LX を所有するシステムは、異なるアドレススペース用であっても LX を再利用することができます。Adabas では、CSA 内のテーブルで使用中の LX を識別することにより、将来のニュークリアスで利用可能なときに、LX を再使用できるようにします。

再利用不可のシステム LX 数は SYS1.PARMLIB 内のメンバ IEASYSxx に NSYSLX パラメータを使用して指定できます。この値を変更する場合、IPL を実行して変更を有効にする必要があります。

次の点を考慮して NSYSLX 値を決定してください。

- おそらく一部の LX はすでに他のシステム機能によって使用されています。したがって、他のユーザーがシステムで使用する LX が不足する可能性は少ないと思われます。
- 同時にアクティブになる Adabas ニュークリアス (または他の任意のターゲット) ごとにシステム LX が 1 つ必要です。10 進数の 64 の値を指定すると、Adabas ニュークリアスやその他のターゲットを同時に 64 個まで実行できます。このとき、LX を使用する他のコンポーネントを制限することはほとんどありません。
- Entire Net-Work バージョン 5 は、リモートデータベース数に関係なく、1 つの LX と 1 つの ID テーブルエントリしか使用しません。これは、以前の Entire Net-Work バージョンの擬似 MPM コンセプトとは異なります。
- ADASIP を REPLACE オプションで実行すると、現在の ID テーブルに保存されたすべての LX は次の IPL まで再使用できません。

同様に、Adabas セッションが `FORCE` オペレータコマンドで正常に終了した場合または `ESTATE` 処理中に異常終了した場合も（例：S222 オペレータキャンセルまたはスナップダンプ中の S722 スプール制限超過アベンドなど）、LX を次の IPL まで再使用できません。

これらのターゲットに送られたどのコマンドも S0D6 でアベンドします。ニュークリアスを再スタートしようとする、`DUP ID (LOCAL)` 付きの `ADAM98` メッセージを受け取り、アベンドします。この2つの問題を解決するには、`ADARUN FORCE=YES` および `IGNDIB=YES` パラメータでニュークリアスを再スタートします。

最初のターゲットがシステム LX を取得しようとしたときに、利用できる LX がなかった場合、アベンドコード S053 と理由コード 0112 で終了します。ターゲットは次の IPL まで開始できません。

Global Resource Serialization の要件

Adabas では Adabas ニュークリアスとユーティリティ間の同期処理の実行に Global Resource Serialization (GRS) を使用します。Adabas の GRS 要求が有効になるよう、システムで GRS が正しく設定されていることが重要です。

次の点を考慮して GRS を設定してください。

- Adabas は GRS の `ENQ`、`DEQ` マクロを、範囲をシステム全体 (`SCOPE=SYSTEMS`)、メジャーネームを `'ADABAS'(QNAME)` で使用します。
- データベースがオペレーティングシステムの複数のイメージ（複数の LPAR もしくはマシン）間で共有されているディスク上にあり、Adabas ニュークリアスとユーティリティがこれらの複数のイメージから実行される可能性のある場合、すべてのシステムイメージでシステム全体を対象とする `ENQ` 要求が有効になるように GRS がインストールされていることを確認してください。

EXCPVR の使用

Adabas では、`EXCP` の代わりに `EXCPVR` を使用することで、チャンネルプログラム変換のオーバーヘッドが削減されるため、パフォーマンスとシステムスループットを向上できます。

Adabas で `EXCPVR` を起動するには、次の条件を満たしている必要があります。

- Adabas モジュールが APF 認可ライブラリ内にあること。
- `ADARUN` が `SETCODE AC(1)` でリンクされていること。

現在では、APF 認可された Adabas ニュークリアスとユーティリティの実行が標準的になっています。詳細については、「[APF 認可の要件](#)」を参照してください。

以前のバージョンの Adabas では、データベース I/O の EXCP または EXCPVVR の使用と、ロードライブラリの APF 認可は相互に関連します。APF 認可なしで実行している場合は常に EXCP が使用され、APF 認可ありで実行している場合には EXCPVVR が常に使用されます。APF 認可を実行している場合に EXCP を使用するには、特別な A\$-zap または AY-zap の適用が必要です。

Adabas 8 では、z/OS 環境で使用できる新しい ADARUN パラメータ、EXCPVVR が導入されています。このパラメータを使用すると、APF 認可の実行時に EXCP または EXCPVVR のどちらを使用するかを指定できます。EXCPVVR パラメータを "YES" に設定すると、EXCPVVR が使用され、"NO" に設定すると、EXCP が使用されます。APF 認可なしのロードライブラリから実行しているときに、EXCPVVR パラメータを "YES" に設定すると、EXCPVVR パラメータは無視されます。このパラメータの詳細については、「EXCPVVR : EXCP または EXCPVVR の使用の制御」を参照してください。

EXCPVVR パラメータに加え、別の ADARUN パラメータ PGFIX も EXCPVVR の使用に影響します。z/OS システムで EXCPVVR が使用されている場合、PGFIX パラメータは I/O コントロールブロックを含むページが I/O 処理の完了後に解放されるか、ジョブ終了後に解放されるかを制御します。このパラメータは、z/OS で EXCPVVR を使用している場合のみ有効です。PGFIX パラメータを NO に設定すると、I/O コントロールブロックを含むページは I/O 処理の間だけ固定され、YES に設定すると、I/O コントロールブロックを含むページは、ジョブが継続する間固定されます。このパラメータの使用の詳細については、「PGFIX : EXCPVVR ページ固定制御」を参照してください。

共有可能な ADARUN の作成

Adabas ロードライブラリ内に提供される ADARUN モジュールは再利用できません。共有可能な ADARUN が必要な場合、REUS=YES リンクエディット属性を付けて再リンクする必要があります。

再利用可能オプションを指定して ADARUN をリンクすると、同じアドレススペースで実行されている複数のプログラムが同一の ADARUN を共有し、最終的には ADALNK の同一のコピーを共有することができます。異なるプログラムが 1 つの Adabas ユーザー ID を共有する必要があるとき、また、ADALNK ユーザー出口の単一コピーが必要な場合にも、これは重要です。

共有可能な ADARUN を作成するには、JOBS ライブラリのサンプルジョブ JCLLINRR を使用して、REUS 属性で再リンクします。

再利用可能な ADARUN と再利用不可の ADARUN の両方が必要な場合は、どちらも ADARUN という名前でロードできるため、それぞれを異なるロードライブラリに置く必要があります。

16MB を超えるストレージ

Adabas では、16MB のアドレス制限より上に必要なエリア（バッファスペースを含む）を取得し、バッファプールサイズを拡張できます。

逆にスペースを 16MB 制限より下に割り当てるには、次の例の MODE ステートメントにある AMODE の値を AMODE(24) に設定します。

```
//LINKRUN EXEC PGM=IEWL,PARM='REUS'  
//SYSPRINT DD SYSOUT=*  
//SYSUT1 DD UNIT=SYSDA,SPACE=(CYL,(1,1))  
//ADALIB DD DSN=user.loadlib,DISP=SHR  
//SYSLMOD DD DSN=user.loadlib,DISP=SHR <=APF-authorized library  
//SYSLIN DD *  
MODE AMODE(24),RMODE(24)  
SETCODE AC(1)  
INCLUDE ADALIB(ADARUN)  
NAME ADARUN(R)
```

さらに、Adabas は JOB ステートメントまたは EXEC ステートメントで、あるいはインストールのデフォルトとして、十分な REGION を指定して実行する必要があります。例としては、次のようなものがあります。

```
//BIG JOB ...,REGION=30M,...
```

2GB を超えるストレージ (64 ビット)

- 実ストレージ
- 仮想ストレージ

実ストレージ

Adabas では 2GB 境界を超えるリアルページを利用することができます。この機能により Adabas は 64 ビットのリアルアドレスを I/O エリアとして使用することができます。

Adabas が 64 ビットの実ストレージをサポートするには Adabas 実行時に APF 認可されている (EXCPVR を使用) か、されていない (EXCP を使用) かによって決まります。実行モードは、ADAI65 メッセージに示されます。

```
ADAI65 EXCPVR IS {BEING | NOT BEING} USED FOR THIS RUN IN ESA64 MODE
```

64 ビットの実ストレージをサポートするには、次のいずれかが必要です。

- ARCHLEVEL=2 (z/architecture モード) の z/OS R10
- z/OS V1 R2 以降

これらは、実ストレージ割り当て用の 2 ギガバイト以上の LPAR を持つ IBM 2064 ファミリのプロセッサ上で動作している必要があります。

仮想ストレージ

IBM は 64 ビット仮想ストレージを z/OS V1 R2 以降でのみサポートしています。

Software AG では、Adabas Caching Facility (ACF) 製品を使用する IBM の 64 ビット仮想ストレージのサポートを提供しています。詳細については Software AG にお問い合わせください。

Adabas Caching Facility のデモが ADAvrs.ALLINPL ファイル (vrs はテープで配布される最新の Adabas バージョン) に含まれています。

ZAP の適用

個々のオペレーティングシステムに ZAP を適用するには、z/OS の AMASPZAP ユーティリティを使用します。この方式はデータを検証 (VER) し、置き換え (REP) ます。次のサンプル JCL は、AMASPZAP を実行します。

```
//ADAZAP JOB
//STEP1 EXEC PGM=AMASPZAP
//SYSPRINT DD SYSOUT=X
//SYSLIB DD DSN=ADABAS.Vvrs.LOAD,DISP=SHR
//SYSIN DD *
(zap control statements)
/*
//
```

上記の場合の ZAP コントロールステートメントの例を次に示します。

```
NAME membername csectname
VER displacement data
REP displacement data
IDRDATA (up to eight bytes of user data)
* (comment)
```



Note: VER ステートメントおよび REP ステートメントで、コマンド、displacement、data の区切りには必ずスペースを使用してください。コンマをデータ区切りとしても問題ありませんが、コンマとスペースの同時使用またはスペースのみは受け入れられず、エラーが発生します。

Adabas 7 ADALINK について



Note: ユニバーサルエンコーディングサービス (UES) を使用したデータ変換が有効なデータベースの接続についての詳細は、「[UES 対応データベースの接続](#)」セクションを参照してください。

ユーザー出口 B (プレコマンド) およびユーザー出口 A (ポストコマンド)

1 つまたは 2 つのユーザー出口を ADALINK ルーチンとリンクできます (SMA ジョブ番号 I088)。

- UEXITB は、コマンドがルーター 04 コールでターゲットに渡される前に制御を受け取ります。



Note: ユーティリティおよび Adabas Online System から発行される特殊コマンドは、物理的なコールとしてマークされます。これらのコールはユーザー出口にバイパスされなければなりません。これらのコールでは、コマンドの Adabas コントロールブロック (ACB) の先頭バイト (TYPE フィールド) に X'04' があります。UEXITB はこのバイトをチェックし、それが X'04' に設定されている場合にはリターンする必要があります。リターン時には必ず R15 を 0 にリセットしてください。

- UEXITA は、ターゲット、ルーター、ADALINK 自身でコマンドの処理を完了した後に制御を受け取ります。

出口へのエントリで、レジスタの内容は次のようになっています。

レジスタ	内容
1	UB のアドレス。 フラグビット UBFINUB がリセットされると、Adabas+X'86' のハーフワードの内容は UBLUINFO に移されます。この内容が 0 より大きい場合、UBINFO (UB+X'40') で開始する 2 バイトは 0 にセットされます。 UBFINUB がセットされると、UB または ACB (ACBRSP を除く) への変更は行われません。
2	16 ワードのセーブエリアのアドレス (ADALNC の場合だけ)
13	18 ワードのセーブエリアのアドレス (非 CICS ADALINK 出口用)
14	リターンアドレス
15	エントリポイントアドレス。UEXITB または UEXITA

ユーザー出口に変更されるレジスタ (レジスタ 15 以外) はどれも保存され、リストアされる必要があります。このためのセーブエリアのアドレスはレジスタ 13 にあります (ADALNC の場合はレジスタ 2)。

UEXITB からの戻りでレジスタ 15 に 0 以外の値が含まれている場合、コマンドはターゲットに送られず、コール元に返されます。ユーザー出口は、コール元プログラムにコマンドが抑制されたことを示すため、ACBRSP にゼロ以外の値を設定する必要があります。レスポンスコード 216 はこのために予約されています。

UEXITB 出口は UB フィールド UBLUINFO に小さい値 (0 を含む) を設定することができます。ユーザー出口が UBLUINFO に大きい値を設定するとアベンドが発生します。UBLUINFO 長は、他の出口 (例: Adabas Review や Adabas Fastpath) が使用されているときは変更できません。

UEXITA 出口が受け取ったユーザー情報は変更されていることがあります。この変更には、ADALINK ユーザー出口のどれかによる長さの縮小 (0 の場合もあり) などがあります。

ADALINK ルーチンが返す ACBRSP のゼロ以外のレスポンスコードを次に示します。

レスポンスコード	説明
213	ID テーブルがありません。
216	UEXITB はコマンドを抑制しました。
218	UB を使用することができません。

少なくとも次の3つの EQU 文（ソースの先頭に記述）は、ADALINK ルーチンをアセンブルする前に変更できます。ただし、いくつかの ADALINK ルーチンでは対応する情報を ZAP できます。

EQU 文	説明
LOGID	デフォルトの論理 ID。1~65535。デフォルトは 1 です。
LNUINFO	ADALINK ユーザー出口に渡すユーザー情報の長さ。0~32767。デフォルトは 0 です。
SVCNR	Adabas SVC 番号。値の範囲やデフォルトはオペレーティングシステムに依存します。この値は、次の ADALINK ルーチンのアセンブリで SYSPARM 値として指定することができます。 //EXEC PGM=ass,PARM='.....,SYSPARM(svcnr)'

全 Adabas リンクルーチンの先頭 152 (X'98') バイトは次の構造を保持する必要があります。

オフセット	ラベル	内容	説明
00	ADABAS		エントリコード
12		CL6'ADALNx'	プログラム名
18		XL4'yymmdd'	アセンブリ日付
1C		A(ZAPTAB)	ZAP テーブルのアドレス
20	PATCH	XL96'00'	パッチエリア
80	LNKLOGID	AL2(LOGID)	デフォルト論理 ID (デフォルト: 1)
82		82 XL2'00'	予約済み
84	LNKSVC	SVC SVCNR	Adabas SVC の実行可能 SVC 命令 (デフォルト: オペレーティングシステム依存)
86	LUINFO	Y(LNUINFO)	ユーザー情報の長さ (デフォルト: 0)
88	VUEXITA	V(UEXITA)	コール後のユーザー出口のアドレス (weak)
8C	VUEXITB	V(UEXITB)	コール前のユーザー出口のアドレス (weak)
90	ADABAS51	CL8'ADABAS51'	IDT ID

ADAUSER について

ADAUSER はユーザーを Adabas にリンクするプログラムです。これはオペレーティングシステムに固有であり、リリースレベルやモードには依存しません。バッチ環境やいくつかの TP 環境 (TSO など) で使用できます。

ADAUSER は次のような方法で処理を行います。

- ADAUSER にはエントリポイント ADABAS が含まれているので、Adabas をコールするすべてのユーザープログラムとリンクしてください。CSECT またはエントリポイント名 ADABAS

を含んでいる他のプログラムをこれらのロードモジュール／フェーズにリンクすることはできません。

- Adabas への最初のコールで、ADAUUSER は最新バージョンの ADARUN をロードします。これにより、コール処理をリリースレベルに依存することなく使用できます。後続の Adabas コールは ADARUN をバイパスします。
- ADARUN はそのコントロールステートメントを処理します。ADARUN PROGRAM パラメータの値が USER の場合（デフォルト）、ADARUN は ADAIOR をロードし、ADARUN MODE パラメータの指定が MULTI か SINGLE かによって適切な TP ADALINK または ADANUC をそれぞれロードします。これにより、コール処理がモードに依存しなくなります。

Adabas 8 ADALINK について



Note: ユニバーサルエンコーディングサービス (UES) を使用したデータ変換が有効なデータベースの接続についての詳細は、「[UES 対応データベースの接続](#)」セクションを参照してください。

リンクルーチンユーザー出口1 (プレコマンド) およびユーザー出口2 (ポストコマンド)

1つまたは2つのユーザー出口を ADALINK ルーチンとリンクできます。

- リンクルーチンユーザー出口1の LUEXIT1 は、コマンドがルーター 04 コールでターゲットに渡される前に制御を受け取ります。



Note: ユーティリティおよび Adabas Online System から発行される特殊コマンドは、物理的なコールとしてマークされます。これらのコールはユーザー出口にバイパスされなければなりません。これらのコールでは、コマンドの Adabas コントロールブロック (ACBX) の先頭バイト (TYPE フィールド) に X'04' があります。LUEXIT1 はこのバイトをチェックし、それが X'04' に設定されている場合にはリターンする必要があります。リターン時には必ず R15 を 0 にリセットしてください。

- リンクルーチンユーザー出口2の LUEXIT2 は、ターゲット、ルーター、または ADALINK 自身でコマンドの処理を完了した後に制御を受け取ります。

出口へのエントリで、レジスタの内容は次のようになっています。

レジスタ	内容
1	UB のアドレス。 フラグビット UBFINUB がリセットされると、Adabas+X'86' のハーフワードの内容は UBLUINFO に移されます。この内容が 0 より大きい場合、UBINFO (UB+X'40') で開始する 2 バイトは 0 にセットされます。 UBFINUB がセットされると、UB または ACB (ACBRSP を除く) への変更は行われません。
2	16 ワードのセーブエリアのアドレス (ADALNC の場合だけ)
13	18 ワードのセーブエリアのアドレス (非 CICS ADALINK 出口用)
14	リターンアドレス
15	エントリポイントアドレス。LUEXIT1 または LUEXIT2。

ユーザー出口によって修正されるレジスタ (レジスタ 15 以外) はどれも保存され、リストアされる必要があります。このためのセーブエリアのアドレスはレジスタ 13 にあります。

LUEXIT1 からの戻りでレジスタ 15 に 0 以外の値が含まれている場合、コマンドはターゲットに送られず、コール元に返されます。ユーザー出口は、コール元プログラムにコマンドが抑制されたことを示すため、ACBXRSP にゼロ以外の値を設定する必要があります。レスポンスコード 216 はこのために予約されています。

LUEXIT1 出口は UB フィールド UBLUINFO に 0 より小さい値 (0 を含む) を設定することができます。ユーザー出口が UBLUINFO に 0 より大きい値を設定するとアベンドが発生します。その他の出口が使用されている場合、UBLUINFO の長さを変更することはできません。

LUEXIT2 出口が受け取ったユーザー情報は変更されていることがあります。この変更には、ADALINK ユーザー出口のどれかによる長さの縮小 (0 の場合もあり) などがあります。

ADALINK ルーチンが返す ACBXRSP のゼロ以外のレスポンスコードを次に示します。

レスポンスコード	説明
213	ID テーブルがありません。
216	LUEXIT1 はコマンドを抑制しました。
218	UB を使用することができません。

ADAUSER について

ADAUSER はユーザーを Adabas にリンクするプログラムです。これはオペレーティングシステムに固有であり、リリースレベルやモードには依存しません。バッチ環境および一部の TP 環境で使用できます。

ADAUSER にはエントリポイント ADABAS が含まれているので、Adabas をコールするすべてのユーザープログラムとリンクしてください。CSECT またはエントリポイント名 ADABAS を含んでいる他のプログラムをこれらのロードモジュールにリンクすることはできません。

Adabas への最初のコールで、ADAUSER は最新バージョンの ADARUN をロードします。これにより、コール処理をリリースレベルに依存することなく使用できます。後続の Adabas コールは ADARUN をバイパスします。

ADARUN はそのコントロールステートメントを処理します。ADARUN の設定が PROGRAM=USER (デフォルト) の場合、ADARUN は非リエントラント ADALINK モジュールをロードします。リエントラントバッチリンクルーチンをロードするには、ADARUN パラメータ PROGRAM=RENTUSER を使用します。これにより、コール処理がモードに依存しなくなります。

4 TP モニタを使用する場合の Adabas のインストール

■ IBM プラットフォーム用の Adabas リンクルーチンの準備	42
■ Adabas 8 配下での IMS/TM を使用する場合の Adabas のインストール	46
■ CICS を使用する場合の Adabas インストールの全般的な考慮事項	49
■ Adabas 8 配下での CICS を使用する場合の Adabas のインストール	51
■ Adabas 8 用 CICS ハイパフォーマンススタブルーチンのインストール	53
■ Adabas 8 配下での Com-plete を使用する場合の Adabas のインストール	78
■ バッチ/TSO を使用する場合の Adabas インストールの全般的な考慮事項	81
■ Adabas 8 配下での Batch/TSO を使用する場合の Adabas のインストール	82
■ Adabas データベース ID による Adabas SVC ルーティングの確立	85
■ バージョン 8 のソースメンバのデフォルト (LGBLSET マクロ) の変更点	94

このchapterでは、Adabas をバッチモードで、TP モニタとともにインストールするために必要な情報について説明します。本書には、次のトピックが含まれています。

IBM プラットフォーム用の Adabas リンクルーチンの準備

このsectionでは、IBM プラットフォーム用の TP モニタの Adabas リンクルーチンの準備について説明します。Adabas 8 リンクルーチンのソースモジュールは、Adabas 8 の基本ソースライブラリでは提供されません。Adabas 8 リンクルーチンは、ZAP 経由またはリンクグローバルテーブルを使用してのみカスタマイズできます。

- Adabas リンクルーチンのアドレッシングモードのアセンブリ指示
- UES 有効リンクルーチン

Adabas リンクルーチンのアドレッシングモードのアセンブリ指示

すべての Adabas 8 リンクルーチンには、AMODE および RMODE アセンブリ指示が含まれています。これらのアセンブリ指示により、リンク JCL、JCS、または EXEC で、AMODE または RMODE リンケージエディタコントロールステートメントの競合が発生した場合、リンケージエディタは警告メッセージを生成します。

これらのアセンブリ指示は、各リンクルーチンに対して選択された AMODE および RMODE を記述するためにも使用されます。リンクルーチン内外のこれらの指示が、実行中のリンクルーチンの実際のアドレッシングモードを変更しないことに注意してください。

Adabas 8 リンクルーチンの再リンク

特定の AMODE と RMODE の組み合わせで、Adabas 8 リンクルーチンを再リンクすると、リンケージエディタによって警告メッセージが生成されることがあります。これは、リンクルーチンを構成するロードモジュールの ESD レコードの AMODE または RMODE の競合に関連している場合、かつ意図したアプリケーションプログラムのコールで結果のモジュールが実行される際に、モジュールに適切な AMODE および RMODE 属性が設定されている場合に限り、無視しても問題ありません。

注意が必要なのは、リンクルーチンの再リンク時に選択した属性でリンクルーチンを起動するとき、AMODE(24) アプリケーションが正常に動作するかどうかです。これは、RMODE(ANY) 属性が関連付けられているリンクルーチンが、動的にロードされるにもかかわらず、AMODE(24) のプログラムによって起動される場合に特に重要です。この場合、アドレスの例外によるアベンドを避けるため、リンクルーチンを AMODE(31)、RMODE(24) で再リンクする必要があります。これは AMODE(24) アプリケーションはリンクルーチンが 16MB 境界より上にある場合に、リンクルーチンを正常に起動できないためです。

Adabas 8 リンクルーチンはすべて、初期化後に AMODE(31) で実行されますが、コール元の AMODE でコール元に返されます。

 **Note:** CICS では、V8 リンクは AMODE(31) で実行されますが、実行中の CICS トランザクションの AMODE と RMODE は、Dataloc RDO パラメータによって決定されます。

バッチ/TSO 非リエントラントリンクルーチン ADALNK は AMODE(31)、RMODE(24) でアSEMBルとリンクが行われています。これは AMODE(24) または RMODE(24) アプリケーションプログラムをサポートするための推奨コンフィグレーションです。必要に応じて、AMODE(31)、RMODE(ANY) で再リンクすることもできますが、すべてのコール元プログラムが AMODE(31) であることを確認できた場合のみにしてください。

ADALNKR バッチ TSO リエントラントリンクルーチンは、AMODE(31)、RMODE(ANY) でリンクエディットされています。このリンクルーチンが AMODE(24) のアプリケーションによってロードされる場合、AMODE(31)、RMODE(24) で再リンクする必要があります。

ZOS Com-plete モジュール ADALCO は、AMODE(31)、RMODE(ANY) でアSEMBルとリンクが行われています。Com-plete TP モニタは、Com-plete Adabas インターフェイスルーチン TLOPADAB を通じて、ADALCO を起動する AMODE(24) または RMODE(24) プログラム間を切り替えるときに、適切な AMODE が選択されるようにします。

すべての V8 CICS リンクルーチンモジュール (ADACICS、ADACICT、ADACIC0、ADACIRQ) は、AMODE(31)、RMODE(ANY) でリンクエディットされています。CICS はプログラムのロードと起動を、プログラムおよびトランザクション定義に関連付けられた DATALOC 値に基づいて管理します。

Adabas IMS インターフェイスリンクルーチン ADALNI は、AMODE(31)、RMODE(ANY) でリンクエディットされています。これは最近の IMS アプリケーション用に推奨されるコンフィグレーションです。インストール環境に AMODE(24) IMS アプリケーションが存在する場合は、ADALNI を AMODE(31)、RMODE(24) で再リンクすることもできます。

ADAUSER AMODE/RMODE について

Software AG では、すべてのバッチアプリケーションで ADAUSER モジュール経由で Adabas コールを起動することをお勧めしています。このモジュールは、通常アプリケーションプログラムとリンクエディットされ、その後 ADARUN や ADAIOR/ADAIOS とともに適切なリンクルーチンをロードします。ソースメンバには AMODE 31、RMODE ANY としてコーディングされた AMODE 命令と RMODE 命令があります。これはさまざまなアプリケーションプログラムに対して ADAUSER をアSEMBルおよびリンクするための最も柔軟なコンフィグレーションです。しかし、ADAUSER が動的にロードされる場合は、再アSEMBルの前に RMODE アSEMBラ指示を RMODE 24 に変更する必要があります。または、ADAUSER モジュールを AMODE(31)、RMODE(24) で再リンクして、AMODE 24 アプリケーションが 16 MB 境界の下で適切に起動できるようにする必要があります。

UES 有効リンクルーチン

Adabas8 バッチおよび TSO リンクルーチンのソースコードは Adabas8 には含まれていません。これらのモジュールは、LNKUES で、変換テーブル ASC2EBC および EBC2ASC とともに提供されます。このバージョンの Adabas 8 では、これらの UES コンポーネントとともに実行してください。バージョン 8 ターゲットデータベースが UES 有効データベースではない場合、リンクルーチンがそれを検出し、コールが UES 変換を必要とするクライアントからのものであった場合、Adabas レスポンス 228 を返します。

Adabas 8 Com-plete リンクルーチンは、LCOGBL モジュールの設定から、UES サポートが必要かどうかを判断します。LCOGBL モジュールの設定は、Com-plete を使用する Adabas のインストールで修正およびアセンブルします。詳細については、「[Com-plete を使用する場合の Adabas のインストール](#)」を参照してください。

この section では、次のトピックについて説明します。

- デフォルトまたはカスタマイズした変換テーブル
- LNKUES と LNKUES7 のコール
- z/OS 用の Adabas 8 ジョブのユニバーサルエンコーディングサポート
- Adabas 8 ルーチンの UES サポートの無効化

デフォルトまたはカスタマイズした変換テーブル

デフォルトでは、Adabas 8 リンクルーチンのロードモジュールは、LNKUES およびデフォルト変換テーブルとリンクされています。

LNKUES は Adabas バッファのデータを変換し、コール元のアーキテクチャに応じて、必要な場合はバイトスワップを行います。

標準的な変換テーブルは次の 2 つです。

- ASC2EBC：ASCII から EBCDIC への変換
- EBC2ASC：EBCDIC から ASCII への変換

Adabas 変換テーブルペアについては、「[変換テーブル](#)」を参照してください。

ロードモジュールとリンクされたデフォルトの変換テーブルを使用する代わりに、独自のカスタマイズした変換テーブルを準備し、テーブルを再アセンブルして、配布された LNKUES または LNKUES7 モジュールとリンクすることもできます。



Notes:

1. ただし、変換テーブルの変更が必要になるのは、コントロールブロックのアディション1（ユーザーID）またはアディション3フィールドで、“A～Z、a～z、0～9”以外の言語固有の文字を使用する必要のある特殊な場合のみです。

2. LNKUES モジュールは機能的にはリエントラントですが、Adabas ロードライブラリでは、その方法でリンクされていません。
3. LNKUES ロードモジュールと変換テーブルをリンクするときに、リンケージエディタは、リンクするモジュールのリエントラントまたは再利用可能ステータスに関する警告メッセージを生成することがあります。これらの警告メッセージは無視できます。
4. UES サポート用の Adabas 8 リンクルーチンを再リンクする場合、LNKUES を組み込む必要があります。これにより、新しい Adabas 8 アプリケーションが確実に Adabas 8 ダイレクトコールとコントロールブロックをサポートするようになります。

LNKUES と LNKUES7 のコール

LNKUES は、Adabas リンクルーチン要求 (X'1C') コールおよび応答 (X'20') コールで、コミュニケーション ID の先頭バイトに X'01' が含まれ、2 バイト目に EBCDIC (X'04') ビットが設定されていない場合にのみ、呼び出されます。Adabas 8 要求では、LNKUES は UEXIT1 の前に制御を受け取ります。Adabas 8 応答では、LNKUES は UEXIT2 の後に制御を受け取ります。

z/OS 用の Adabas 8 ジョブのユニバーサルエンコーディングサポート

次のリストは、z/OS 環境の Adabas リンクルーチンでユニバーサルエンコーディングサポートを管理するためのサンプルジョブです。

サンプルジョブ	説明
LNKGCICS	CICS グローバルテーブルをアSEMBルし、LNKUES およびデフォルト変換テーブル ACS2EBC、EBC2ASC とリンクします。
LNKLCO8	Com-plete リンクグローバルテーブルを、LNKUES およびデフォルト変換テーブル ACS2EBC、EBC2ASC とリンクします。
LNKLN18	IMS リンクルーチンを、LNIGBL リンクグローバルテーブル、LNKUES、およびデフォルト変換テーブル ACS2EBC、EBC2ASC とリンクします。
LNKLNK8	バッチリンクルーチンを、LNKGBLS リンクグローバルテーブル、LNKUES、およびデフォルト変換テーブル ACS2EBC、EBC2ASC とリンクします。
LNKLNKR8	リエントラントバッチリンクルーチンを、LNKRGBL リンクグローバルテーブル、LNKUES、およびデフォルト変換テーブル ACS2EBC、EBC2ASC とリンクします。

これらのジョブを使用する前に、編集して JOB カードを準備し、ロードライブラリ名を更新する必要があります。また、必要に応じて、環境に合わせたその他の変更を行うこともできます。詳細については、ジョブ内に記述されているコメントを参照してください。

Adabas 8 ルーチンの UES サポートの無効化

このsectionでは、何らかの理由で必要になったときに、Adabas 8 の IMS/TM、Com-plete、およびバッチ/TSO リンクルーチンの UES サポートを無効化する方法について説明します。

▶ **手順 4.1. リンクルーチンの UES サポートを無効化するには、次の手順に従います。**

- 1 関連するリンクルーチンのリンクグローバルテーブルを編集します。UES パラメータを NO に設定します。
- 2 必要に応じて、ソースモジュールの EQU 文とその他の指示に必要な変更を加えた後、リンクグローバルテーブルをアセンブルします。
- 3 Adabas リンクルーチンを新しくアセンブルされたリンクグローバルテーブルとリンクします。このとき、UES コンポーネント (LNKUES、ASC2EBC、EBC2ASC) を組み込まないでください。

特定のリンクルーチンの詳細については、「[Adabas 8 配下での IMS/TM を使用する場合の Adabas のインストール](#)」、「[Adabas 8 配下での Com-plete を使用する場合の Adabas のインストール](#)」、および「[Adabas 8 配下での Batch/TSO を使用する場合の Adabas のインストール](#)」を参照してください。

Adabas 8 配下での IMS/TM を使用する場合の Adabas のインストール

このセクションでは、Adabas 8 で IMS/TM TP モニタを使用する場合の Adabas リンクルーチンのインストールについて説明します。

IMS が Adabas データベースとコミュニケーションする場合は、Adabas リンクルーチンが必要になります。Adabas バージョン 8 実行可能デフォルトリンクルーチンは、AIIvrs.LOAD ライブラリのメンバ ADALNI で提供されます (vrs はテープで提供される最新の Adabas バージョン番号)。このリンクルーチンを変更する場合は、メンバ ADALNI8 を使用してください。ADALNI8 は、準備したリンクグローバルモジュールとリンクする必要があります。また、アプリケーションが IMS メッセージ処理プログラム (MPP) リージョンをコールしたときに、IMS MPP によってロードされる、最終的な ADALNI ロードモジュールを作成するために必要なリンクルーチンもすべてリンクする必要があります。メンバ ADALNI と ADALNI8 には一部デフォルトが設定されています。

このsectionでは、次のトピックについて説明します。

- [Adabas 8 の IMS/TM リンクルーチン](#)
- [Adabas ユーザー ID の取得](#)
- [SAF ID の取得](#)

■ Adabas 8 配下でのインストール手順

Adabas 8 の IMS/TM リンクルーチン

IMS/TM 用の Adabas 8 リンクルーチンを次に示します。

- ADALNI は、メッセージ処理プログラム (MPP) の実行可能なデフォルトモジュールです。リンクルーチンのデフォルトを変更する必要がない場合は、このモジュールを使用します。
- ADALNI8 は、メッセージ処理プログラム (MPP) の基本モジュールとして使用します。インストール環境で ADALNI をカスタマイズする必要がある場合は、ADALNI8 を使用して、更新された ADALNI を生成します。
- ADALNK は、バッチメッセージ処理 (BMP) プログラム、バッチ指向 BMP プログラム、およびバッチ処理プログラム (DLIBATCH) 用の Adabas リンクルーチンです。

ADALNI および ADALNK は、デフォルトで CSECT 名および ENTRY 指示 ADABAS を使用します。

Adabas バージョン 8 の ADALNI と ADALNK は UES が有効な状態で提供されています。詳細は、「[UES 対応データベースの接続](#)」セクションを参照してください。

このsectionでは、ADALNI と ADALNI8 の使い方についてのみ説明します。ADALNK の使い方については、「[バッチ/TSO を使用する場合の Adabas インストールの全般的な考慮事項](#)」を参照してください。

Adabas ユーザー ID の取得

Adabas ユーザー ID は、ADALNI ロードモジュールの実行時に IOPCB の LTERM フィールド (先頭 8 バイト) から取得されます。ユーザー ID は Adabas ユーザーブロックフィールド UBUID に格納され、Adabas コミュニケーション ID の末尾 8 バイトに使用されます。

SAF ID の取得

SAF ID は、IBM の RACF または CA の ACF2 などの外部セキュリティパッケージが存在する場合に、Adabas SAF Security (ADASAF) で使用するためにサポートされています。SAFID は、ADALNI ロードモジュールの実行時に、IOPCB のユーザー ID フィールド (33~40 バイト目) から取得されます。有効な SAF ユーザー ID を取得するには、IMS インストール環境で SAF サインオンがアクティブになっている必要があります。また、ユーザーが IMS/SIGN コマンドを実行して、IMS 端末にログオンしている必要があります。

Adabas 8 配下でのインストール手順

▶手順 4.2. デフォルト設定を変更し、IMS 用の Adabas 8 リンクルーチンを準備するには、次の手順に従います。

- 1 Adabas 8 AIIvrs.SRCE ライブラリに含まれているサンプルメンバ LNIGBL を変更可能な任意のユーザーソースライブラリにコピーします。これらのモジュールには、リンクコンポーネントのデフォルト設定を作成するのに使用する LGBLSET パラメータが含まれています。LGBLSET パラメータの詳細については、「バージョン 8 でのソースメンバデフォルトの修正 (LGBLSET マクロ)」に記載されています。
- 2 ユーザーソースライブラリの LNIGBL メンバを修正します。



Note: OPSYS パラメータは ZOS に設定する必要があります。

- 3 サンプルジョブ ASMGBLS を、ジョブの先頭で示されているとおりに修正および実行します。ASMGBLS は Adabas 8 ADAvrs.JOBS ライブラリ内にあります。完全に修正する場合、ジョブの SET ステートメントが前の手順で準備した LNIGBL メンバを参照し、NAME リンクエディットコントロールステートメントが LNIGBL メンバの GBLNAME パラメータによって指定される名前を参照している必要があります。

修正したら、ASMGBLS ジョブを実行してリンクグローバルモジュールをアセンブルおよびリンクエディットします。

新しいリンクグローバルモジュールが、ASMGBLS ジョブで指定されたユーザーロードライブラリに生成されます。モジュール名は LNIGBL の GBLNAME パラメータで指定された名前になります。

- 4 サンプルジョブ LNKLNI8 をユーザーソースライブラリにコピーし、前の手順で作成した新しいリンクグローバルモジュールおよびその他の必要な出口が ADALNI8 基本モジュールとリンクするように修正します。サンプルジョブの修正手順は、ジョブの先頭に記載されています。このジョブの出力先を適切なユーザーロードライブラリに設定するようにしてください。LNKLNI8 は Adabas 8 AIIvrs.SRCE ライブラリ内にあります。

このジョブで生成されるモジュールは ADALNI です。

- 5 IMSMPP リージョンで使用できるロードライブラリに ADALNI モジュールを配置します。

これで Adabas 8 リンクルーチンの準備は完了です。

CICS を使用する場合の Adabas インストールの全般的な考慮事項

マクロレベルリンクルーチン ADALNC は z/OS 配下で実行する CICS のすべてのレベルでサポートされなくなりました。これらの環境では、現行バージョンの Adabas を実行し、提供されるコマンドレベルリンクコンポーネントを使用する必要があります。

Adabas コマンドレベルリンクルーチンは CICS トランザクションサーバー (CTS) 環境をサポートしています。



Notes:

1. USERID フィールドに対する OPID オプションは CICS/TS 1.1 以上ではサポートされていません。したがって、コマンドレベルリンクルーチンも提供されていません。
2. Adabas 8 SVC では、Adabas 7.4 以降の CICS コンポーネントが必要になります。

次のセクションでは、Adabas/CICS に固有のインストールと操作について、CICS 側の観点から説明します。

- [Adabas Bridge for VSAM の考慮事項](#)
- [CICS MRO 環境要件](#)
- [CICS ストレージ保護の使用](#)
- [リソース定義のサンプル](#)

Adabas Bridge for VSAM の考慮事項

CICS 配下で Adabas Bridge for VSAM 4.2 もしくは 5.1 を実行している場合、CICS 3.3 以上および Adabas バージョン 7.1 以上のコマンドレベルリンクルーチンを実行しなければなりません。

CICS MRO 環境要件

Adabas CICS コマンドレベルリンクルーチンを CICS マルチプルリージョンオプション (MRO) で実行する場合、MRO パラメータを "YES" に設定し、NETOPT パラメータのデフォルト値を使用する必要があります。Adabas 8 インストールでは、これらのパラメータは、LGBLSET マクロで指定します。詳細については、「[バージョン8でのソースメンバデフォルトの修正 \(LGBLSET マクロ\)](#)」を参照してください。

Adabas をコールするアプリケーションが複数アプリケーションリージョン間に渡る場合、LGBLSET NTGPID パラメータを使用して、Adabas SVC によって使用される Adabas コミュニケーション ID に 4 バイトのリテラルを指定できます。

あるいは、次の処理を行うリンクルーチンのユーザー出口を作成することができます。

- UBFLAG1 (UB DSECTのバイト X'29') に値 X'08' (UBF1IMSR) を指定します。
- 4 バイトの英数字値を UB フィールド UBIMSID に挿入します。

この出口はリンクユーザー出口 1 (LUEXIT1) です。この出口によって、複数リージョンでトランザクションが発生する場合でも、AdabasSVCは、Adabasコマンドキューエレメント (CQE) に適切な Adabas コミュニケーション ID を提供することができます。

CICS ストレージ保護の使用

ストレージ保護機構 (STGPROT) は、CICS/ESA 3.3 で導入されました。ストレージ保護は、ストレージプロテクションキーを使用することによって、CICS ストレージまたはユーザーストレージにアクセスする許可をリソースに与えます。

- ユーザーキーでは CICS ストレージを上書きできません。このようにして CICS をある程度保護できます。
- CICS キーでは CICS ストレージまたはユーザーキーストレージのどちらも読み書きでき、CICS リソースへの最高のアクセス権を与えます。

トランザクションアイソレーションは CICS/ESA 3.3 配下で導入されたストレージ保護メカニズムの拡張です。CICS リソースをサブスペースに隔離することにより、さらに強力に CICS リソースを保護します。これは、ユーザーキーリソースを相互に保護し、CICS のカーネルから CICS キーリソースを保護します。トランザクションアイソレーションは CICS TRANISO システム初期化 (SIT) パラメータを使用してグローバルに有効にできます。また、新しいリソース定義 ISOLATE キーワードで各 CICS トランザクションを有効にできます。トランザクションアイソレーションによって、CICS システムの存続中および CICS システムで実行しているすべてのトランザクションに有効でなければならない CICS リソースが制限されます。

Adabas 8 インストールでは、CICS リンクルーチンはタスク関連のユーザー出口、モジュール ADACICT を常に使用するため、ストレージアイソレーションはデフォルトでサポートされています。

リソース定義のサンプル

z/OS および VSE 用の CICS/TS 1.1 以上の場合、CICS プログラムやトランザクションの定義およびインストールには、RDO (リソース定義オンライン) を使用することをお勧めします。CICS ドキュメントでは、現在、リソースの定義に PPT と PCT エントリの再アセンブリを推奨していません。

メンバ DEFADA8 のサンプル DEFINE ステートメントを修正して、IBM DFHCSDUP への入力として使用し、Adabas CICS コマンドレベルコンポーネントを定義します。DFHCSDUP ユーティリティに関する情報については、該当する IBM CICS ドキュメントを参照してください。DEFADA8 メンバは Adabas 8 CICS コマンドレベルソースライブラリ (ACIvrn.SRCE) にあります。

Adabas 8 配下での CICS を使用する場合の Adabas のインストール

- 提供されるモジュール
- Adabas 8 配下でのインストール手順
- CICS 用の DDLINK 入力の準備

提供されるモジュール

次の表に、Adabas 8 配下で CICS を使用する場合の Adabas のインストールをサポートするための、Adabas インストレーションで提供されるモジュールを示します。

モジュール	説明
ADACICS	CICS コマンドレベルモジュール
ADACICT	CICS タスク関連ユーザー出口 (TRUE) モジュール

Adabas 8 配下でのインストール手順

▶ **手順 4.3. Adabas 8 CICS リンクルーチンコンポーネントをインストールするには、次の手順に従います。**

- 1 Adabas 配布ライブラリから Adabas 8 CICS ロードモジュールを CICS DFHRPL 連結内のロードライブラリにコピーします (Adabas 8 ADAvrn.JOBS ライブラリのサンプルメンバ CPYCICSM を参照)。
- 2 Adabas 8 ACIvrn.SRCE ライブラリ内のサンプルメンバ CICSGBL を修正します。このメンバには、デフォルトのインストールパラメータ (LGBLSET) 設定のサンプルが含まれています。このメンバの何を修正するかについては、「バージョン 8 でのソースメンバデフォルトの修正 (LGBLSET マクロ)」を参照してください。



Note: OPSYS パラメータは ZOS に設定する必要があります。

- 3 修正した CICSGBL メンバを固有の名前で適切なユーザーソースライブラリに保存します。
- 4 サンプルジョブ ASMGBLS を、ジョブの先頭で示されているとおりに修正および実行します。ASMGBLS は Adabas 8 ADAvrs.JOBS ライブラリ内にあります。完全に修正する場合、ジョブの SET ステートメントが前の手順で準備した CICSGBL メンバを参照し、NAME リンクエディットコントロールステートメントが CICSGBL メンバの GBLNAME パラメータによって指定される名前を参照している必要があります。
- 5 ACIvrn.SRCE ライブラリの LNKGCICS の内容を確認してから実行し、前の手順で新しくアSEMBルしたグローバルテーブルを任意のユーザー出口または Software AG 製品出口とリンクします。特定の Software AG 製品出口の詳細については、製品のインストールマニュアルを参照してください。LNKGCICS メンバは特定の命令を提供します。グローバルテー

ブルをリンクするロードライブラリは、DFHRPL ライブラリ連結で CICS から利用可能にする必要がある点に注意してください。任意のユーザー出口または Software AG リンクルーチン出口は、このモジュールとリンクエディットする必要があります。

- DEFADA8 メンバを修正して、前の手順（手順4）で作成したリンクルーチングローバルデフォルトテーブルの正しい名前を指定します。デフォルトのモジュール名は CICSGBL です。必要に応じて、このメンバのその他の CICS インストール値を修正します。



Note: リンクグローバルテーブルの名前以外の Adabas 8 CICS プログラム名は、事前に定義されており、変更できません（ADACICS、ADACICT、ADACIRQ、ADACIC0 など）。

- IBM DFHCSDUP ユーティリティを実行して、修正した DEFADA8 を入力として使用する CICS の CICS CSD ファイルを更新します。
- CICS PLTPI テーブルを修正して、Adabas CICS タスク関連ユーザー出口（TRUE）を有効にして開始するエントリを追加します。レガシー Adabas TRUE および新しいバージョン 8 の TRUE を有効にして、PLT の第 2 フェーズで開始するには、Adabas 8 ACIvrn.SRCE ライブラリのメンバ ADAPLTXX をサンプルとして使用してください。
- 修正した PLTPI テーブルをアSEMBルし、目的の CICS リージョンで利用可能なライブラリにリンクします。
- CICS 起動 JCL を修正して、手順 4 で準備したリンクグローバルテーブルの名前を指定する DDLINK DD ステートメントを追加します。この作業の詳細については、「[CICS 用の DDLINK 入力の準備](#)」を参照してください。
- CICS を開始し、コンソールに表示される Adabas TRUE モジュールのインストールに関連するメッセージに注意します。

CICS 用の DDLINK 入力の準備

Adabas 8 CICS リンクルーチンの操作は、各 CICS アドレススペースごとにカスタマイズできます。リンクグローバルテーブルをアSEMBルおよびリンクし、そのテーブルを実行時に CICS から利用できるようにします。テーブルには規則に従った任意のロードモジュール名を付けることができます。CICS が識別可能で、CICS リージョンで使用されている既存のロードモジュール名と競合しないことが条件です。

グローバルテーブルは CICS にプログラムとして定義する必要があります。Adabas 8 ACIvrn.SRCE ライブラリのサンプル DEFADA8 メンバの内容を確認して、CICSGBL というサンプルの定義を参考にしてください。DEFADA8 メンバは、必要に応じて修正し、DFHCSDUP ユーティリティへの入力として使用します。このユーティリティで、CICS CSD の Adabas 8 コンポーネントを定義します。この手順は、Adabas 8 配下のインストール手順でも説明されています。DFHCSDUP ユーティリティの詳細については、適切な IBM CICS ドキュメントを参照してください。

CICS リージョンの各リンクルーチングローバルテーブルには固有の名前があります。Adabas 8 CICS リンクルーチンは、外部の一時的なデータキューから、この名前を指定されます。キュー名は ADAI です。この定義も DEFADA8 メンバに記述されています。

Adabas 8 タスク関連ユーザー出口 (TRUE) を ADACIC0 プログラムから有効にすると、CICS の起動時または ADA0 トランザクション発生時に、ADACIRQ モジュールが起動され、ADAI 一時データキューが読み込まれます。読み込まれたデータは、ファイルまたはパーティション分割されたデータセットメンバ内に提供されるか、DDLINK DD ステートメントによってシステムに入力されます。CICS JCL を修正してこの DD ステートメントを指定する必要があります。修正しない場合は、デフォルトのリンクグローバルテーブル名 "CICSGBL" が使用されます。リンクグローバルテーブルが見つからなかった場合、Adabas 8 TRUE は有効にならず、開始されません。

読み込まれる入力データのフォーマットは次のとおりです。

内容	説明
ADALINK	このキーワードが 1~6 カラム目に存在する必要があります。
スペース	7 カラム目にスペース (空白) が必要です。
LGTNAME= または LGT=	7 カラム目のスペースの後、8 カラム目からキーワード LGTNAME または LGT とそれに続く等号 (=) が必要です。
module-name	LGTNAME または LGT キーワードに続く等号の後ろに、準備したグローバルテーブルのモジュール名が必要です。

例えば、CICS JCL に次のような文を追加します。

```
//DDLINK DD *
ADALNK LGTNAME=CICSGBL
/*
```

この例では、"CICSGBL" という名前のリンクデフォルトグローバルテーブルを準備し、アセンブルおよびリンクエディットして、この CICS に定義する必要があります。

Adabas 8 用 CICS ハイパフォーマンススタブルーチンのインストール

このセクションでは、Adabas 8 で使用する CICS ハイパフォーマンススタブルーチンのインストールについて説明します。ここで説明するモジュールとインストール手順は、既存の Adabas 8 アプリケーションを引き続き使用できるように構成されています。

Adabas ハイパフォーマンススタブルーチンは、Natural バージョン以外の言語 (アセンブラ、COBOL、PL/I など) で書かれたアプリケーションに対する Adabas CICS コマンドレベルリンクコンポーネントで利用可能なダイレクトコールインターフェイス (DCI) 機能を拡張します。



Note: スタブルーチンは Adabas CICS コマンドレベルリンクコンポーネントで使用する必要があります。スタブルーチンは Adabas CICS/VSE マクロレベルリンクコンポーネントで正常に機能しません。Adabas バージョン 8 ライブラリで提供される LNCSTUB モジュールは、Adabas バージョン 7.4 CICS リンクルーチンを使用しても正常に動作します。

DCI によって CICS/TS アプリケーションから、Adabas コマンドレベルリンクルーチン経由での Adabas コールが可能になります。そのため、EXEC CICS LINK と EXEC CICS RETURN コマンドセットを使用してプログラムコントロールを転送するときのオーバーヘッドが回避されます。ハイパフォーマンススタブまたは Natural 3.1 以上からのイニシャルコール (IC) コマンドで正しい環境が設定されると、DCI によって BALR インターフェースの使用が許可されます。

ハイパフォーマンススタブルーチンはアセンブラ言語で記述されています。アプリケーションプログラムとリンクすると、アプリケーションと Adabas CICS コマンドレベルリンクコンポーネントの間のインターフェースとして使用できます。また、Adabas コマンドを実行するときに、アプリケーションプログラムから CALL ステートメントを発行して、スタブルーチンにアクセスできます。

CICS/TS 1.1 以上のアプリケーションは、ハイパフォーマンススタブから次のような効果を得られます。

- CICS/TS 1.1 以上で Adabas コマンドを発行するときに、CICS LINK および RETURN プログラムコントロールメカニズムに関連して CICS サービスの使用が削減されるため、パフォーマンスとスループットが向上します。
- CICS/TS 1.1 以上での Adabas 要求に対するコールメカニズムは、CICS 環境で 1 つのプログラムから別のプログラムに情報とコントロールを渡すときに通常使用される手法よりも単純になっています。

この section では、次のトピックについて説明します。

- [制限と必要条件](#)
- [スタブコンポーネント](#)
- [インストールの概要](#)
- [LNCSTUB を使用した場合のパフォーマンス](#)
- [ソースメンバデフォルトの修正 \(ADAGSET マクロ\)](#)

制限と必要条件

次の制限と必要条件がハイパフォーマンススタブルーチンに適用されます。

1. CICS/TS 1.1 以上

Adabas ハイパフォーマンススタブルーチンは CICS/TS 1.1 以上でサポートされます。

アプリケーションがハイパフォーマンススタブルーチンを正常に実行するには、少なくとも 24 バイトの CICS トランザクションワークエリア (TWA)、または少なくとも 32 バイトの CICS COMMAREA が必要です。Adabas 8 LNCSTUB モジュールおよび Adabas 8 インストー

ル確認プログラムは、IVP プログラム、LNCSTUB、および CICS の間のデータ受け渡しに、CICSTWA ではなく、CICS COMMAREA を使用するようになりました。CICS COMMAREA の使用により、CICS TWA の場合に比べて、次のような利点があります。

- COMMAREA のサイズは、アプリケーションによってコールごとに設定することができます。一方、TWA のサイズは CICS トランザクションが定義されるときに設定されます。
- CICS COMMAREA を使用するアプリケーションは、CICS PLTPI のステージ II または III で実行することができます。CICS TWA は、PLTPI 処理中に使用できません。
- CICS COMMAREA の動的サイズ変更は、Adabas 8 ACBX ダイレクトコール、ACBX コントロールブロック、および Adabas バッファ記述 (ABD) の制限のないフォーマットに適しています。Adabas バージョン 8 のダイレクトコールインターフェイスおよびそのデータ構造については、『Adabas コマンドリファレンス』を参照してください。

2. CICS コマンドレベルリンク

アプリケーションプログラムは CICS コマンドレベルインターフェイスと命令を使用して書かれる必要があります、CICS マクロレベルコマンドは発行できません。

3. サポートされるプログラミング言語

アプリケーションプログラムは ALC (アセンブラ言語)、VS/COBOL、COBOL II、COBOL/LE、PL/I、または C で作成できます。インストール確認プログラム (IVP) は、Adabas ソースライブラリに ALC および COBOL で提供されます。

特定のプログラミング言語に必要な追加条件については、この章の各言語に関連するセクションを参照してください。

スタブコンポーネント

タイプ	メンバ	説明
ソース	ADAGSET ALCSIVP COBSIVP LNCSTUB	LNCSTUB と ALCSIVP のアセンブルに必要なマクロ ALC インストール確認用のソース COBOL インストール確認用のソース ハイパフォーマンススタブ用ソース
ジョブ制御	JCLALCI JCLCOBI JCLLNCS	ACL インストール確認用のサンプル JCL COBOL インストール確認用のサンプル JCL LNCSTUB (ハイパフォーマンススタブ) 用のサンプル JCL

インストールの概要

Adabas CICS ハイパフォーマンススタブルーチンをインストールするには、次の手順に従います。

1. LNCSTUB モジュールを編集、プリプロセス、アセンブル、リンクします。
2. RDO または DFHCSDUP ユーティリティを使用して、アプリケーションプログラム、オプションの IVP、CICS リンクコンポーネントを CICS に定義します。
3. (オプション) 適切なインストール確認プログラム (IVP) を修正、プリプロセス、コンパイルまたはアセンブル、リンクし、実行します。
4. アプリケーションプログラムを修正、プリプロセス、コンパイルまたはアセンブル、リンクし、実行します。

詳細については、次の手順を参照してください。

- [手順 1：LNCSTUB モジュールのインストール](#)
- [手順 2：\(オプション\) IVP のインストールと実行](#)
- [手順 3：アプリケーションプログラムのリンクと実行](#)

手順 1：LNCSTUB モジュールのインストール

Adabas CICS ハイパフォーマンススタブルーチンはアセンブラ言語のモジュールで、配布テープのメンバ LNCSTUB にソース形式で提供されます。

手順 1 では、次の作業を行います。

- [ADAGSET マクロの編集](#)
- (オプション) LNCSTUB エントリポイントの別名を設定します。
- [メンバ JCLLNCS の修正](#)
- [LNCSTUB モジュールのプリプロセス、アセンブル、リンク](#)
- [LNCSTUB をアプリケーションプログラムから利用可能にします。](#)

ADAGSET マクロの編集

 **Note:** ADAGSET マクロの編集については、「[ソースメンバデフォルトの修正 \(ADAGSET マクロ\)](#)」を参照してください。

ライブラリ内の ADAGSET マクロを編集します。これは、LNCSTUB をアセンブルするときに、SYSLIB 連結で使用できます。

LNCSTUB と ALCSIVP IVP モジュールは両方とも次の ADAGSET キーワードから値を取得するようになります。

- LOGID。データベース ID を識別します。

- PARMTYP。LNCSTUB および ALCSIVP プログラムからデータを渡すときに、TWA または COMMAREA を使用するかを指定します。
- ENTPT。LNCSTUB および ALCSIVP プログラムによって起動される CICS リンクルーチンまたは CICS スタブの名前を指定します。Adabas CICS コマンドレベルリンクコンポーネントプログラムが ADACICS 以外の名前でリンクされている場合、**ADAGSET マクロ**の ENTPT キーワードの値を変更します。このフィールドの値は LNCSTUB が発行する EXEC CICS LINK コマンドを提供するときに使用されます。
- TRUENM。使用する Adabas TRUE の名前を指定します。

(オプション) LNCSTUB エントリポイントの別名を設定します。

Adabas 8 LNCSTUB モジュールには、アセンブラの GBLC 変数 (&STBNAME) があります。この変数にはエントリポイントの別名を設定し、コール元プログラムで使用できます。必要に応じて、LNCSTUB ソースメンバの先頭近くにある SETC ステートメントを修正し、別名を設定します。アプリケーションプログラムは、"LNCSTUB"、または SETC ステートメントで設定したエントリポイントの別名を使用して、コールを発行することができます。

メンバ JCLLNCS の修正

メンバ JCLLNCS を使用して LNCSTUB モジュールをプリプロセス、アSEMBル、リンクします。この JCL を環境に合わせて修正するには、メンバの JOB カードと次の表に示す記号の値を変更します。

値	説明
&SUFFIX	CICS トランスレータに使用する接尾値。デフォルト値は 1\$。
&ASMBLR	LNCSTUB ソースコードをアSEMBルするために使用されるアSEMBラプログラムの名前 (ASMA90)。
&M	処理するメンバ名。LNCSTUB または ALCSIVP を指定します。
&STUBLIB	LNCSTUB ロードモジュールを挿入するためのロードライブラリの名前。このライブラリはアプリケーションプログラムをリンクするとき、使用可能である必要があります。
&INDEX	アSEMBラ用の SYSLIB DD ステートメントで使用する CICS マクロライブラリに対するハイレベル修飾子。
&INDEX2	トランスレータ STEPLIB DD ステートメントおよびリンクステップの SYSLIB に使用する CICS ロードライブラリに対するハイレベル修飾子。
&ADACOML	ADACB、ADAGDEF、ADAGSET、および LNCDS コピーコードとマクロが存在する Adabas コマンドレベルソースライブラリの名前。
&ADASRCE	追加コピーコードまたはマクロ展開に使用する Adabas ソースライブラリの名前。
&STBSRCE	配布した Adabas CICS ハイパフォーマンススタブ LNCSTUB が存在するソースライブラリの名前。
&MAC1	プライマリシステムマクロライブラリ (通常は SYS1.MACLIB)。
&OUTC	SYSPRINT、SYSOUT メッセージの出力クラス。

値	説明
®	ステップリージョンサイズ。
&NCAL	リンケージエディタ NCAL パラメータの値。推奨値は NCAL です。
&LSIZE	リンケージエディタに使用されるプライマリおよびセカンダリテーブルサイズ。
&WORK	テンポラリデータセットとユーティリティデータセットに使用する DASD デバイスタイプ。

LNCSTUB モジュールのプリプロセス、アセンブル、リンク

31 ビット命令が使用されているため、CICS プリプロセスの後、LNCSTUB モジュールをアセンブルするとき、ハイレベルアセンブラ (ASMA90) を使用する必要があります。

 **Note:** LNCSTUB モジュールは、リエントラントまたは再利用可能としてリンクできます。リエントラントでリンクした場合、自動的に再利用可能になります。再利用可能でリンクした場合、自動的にリエントラントにはなりません。

CICS マクロライブラリの他に、Adabas CICS コマンドレベルソースライブラリと標準 Adabas ソースライブラリをアセンブリステップの SYSLIB DD ステートメントに指定する必要があります。

- LNCSTUB ロードモジュールをリンクするとき、CICS ロードライブラリを SYSLIB DD ステートメントに連結しないでください。
- LNCSTUB オブジェクトデッキの後の SYSLIN データストリームに次のコントロールステートメントを使用してください。

```
NAME LNCSTUB(R)
```

- LNCSTUB ロードモジュールに CICS スタブモジュール DFHEAI0 と DFHEAI1 を組み込まないでください。次のような実行結果となります。
 - リンケージエディタは、DFHEAI1 が未解決の外部参照であることを示す IEW462 または同様のメッセージを発行します。
 - LNCSTUB モジュールはリンケージエディタによって NOT EXECUTABLE としてマークされることがあります。
 - リンクステップのコンディションコードは 8 になります。

アプリケーションプログラムを LNCSTUB とリンクすると、外部参照はすべて解決されます。CICS スタブピースが欠落した場合に、LNCSTUB のリンクエディットからコンディションコード '04' が発行されるように、リンクエディットパラメータ LET および NCAL を使用することをお勧めします。

LNCSTUB をアプリケーションプログラムから利用可能にします。

LNCSTUB モジュールにはエントリ名 ADABAS があります。このエントリ名は、アプリケーションプログラムがパラメタリスト付きで LNCSTUB に制御を渡すときに CALL ステートメントのオブジェクトとして使用できます。LNCSTUB に対する言語固有のコール規則については、このセクションで後述します。

LNCSTUB モジュールには、&STBNAME の値を設定するための、エントリ名 LNCSTUB または SETC ステートメントで記述したエントリ名の別名があります。どちらの値も、アプリケーションプログラムからパラメタリスト付きで LNCSTUB に制御を渡すときに、CALL ステートメントのオブジェクトとして使用できます。LNCSTUB に対する言語固有のコール規則については、この section で後述します。

LNCSTUB ロードモジュールは DCI 機能を使用するためにアプリケーションプログラムのリンクステップで有効にする必要があります。



Note: 同じステップで、CICS ロードライブラリを有効にしてください。そうでないと、CICS スタブモジュールに対する外部参照が未解決になります。

LNCSTUB ロードモジュールはアプリケーションプログラムをリンクするとき組み込まれるので、アプリケーション言語のアセンブラやコンパイラから利用できるライブラリに設置してください。

手順 2：（オプション）IVP のインストールと実行

2つのインストール確認プログラム（IVP）がソース形式で提供されています。1つはアセンブラ言語用、もう1つは COBOL/VS 用です。これらのプログラムはユーザーアプリケーションに Adabas ハイパフォーマンススタブルーチンを実装するためのサンプルです。また、LNCSTUB モジュールの適切なインストールを確認する方法も提供します。

この section では、これらの IVP について 1 つずつ説明します。

- [アセンブラ IVP \(ACLSIVP\) のインストールと実行](#)
- [COBOL IVP \(COBSIVP\) のインストールと実行](#)



Note: 2つのインストール確認プログラム ALCSIVP と COBSIVP は、Software AG が提供するデモンストレーションファイル EMPLOYEES のフィールド AA と AE のみを使用します。Software AG が提供するデモンストレーションファイルについての詳細は、z/OS インストレーション指示の「[デモンストレーションファイルのロード](#)」を参照してください。

アセンブラ IVP (ACLSIVP) のインストールと実行

LNCSTUB モジュールを使用する Adabas DCI のデモと確認のために、ソースメンバ ALCSIVP が提供されています。このプログラムは、従来の CICS LINK/RETURN メカニズムを使用して一連の Adabas コマンドを発行し、出力データの一部を表示します。次に Adabas DCI と LNCSTUB サブプログラムを使用して同じコール順序を再実行します。

▶ **手順 4.4. アセンブラ IVP (ACLSIVP) をインストールして実行するには、次の手順に従います。**

1 ソースメンバ ALCSIVP の修正

- ファイル番号フィールド DBFNR を修正して、使用する Software AG 提供のデモンストレーションデータベースの EMPLOYEES ファイルにアクセスするために必要な値に一致するようにします。Software AG が提供するデモンストレーションファイルについての詳細は、z/OS インストール指示の「[デモンストレーションファイルのロード](#)」を参照してください。

ALCSIVP プログラムは、**ADAGSET マクロ**に指定された LOGID キーワードからデータベース ID を取得します。

- フィールド FBUFF、SBUFF、VBUFF の値が EMPLOYEES ファイルの FDT とデータ内容に一致しているかをチェックします。
- EXEC CICS LINK ステートメントで使用している名前が Adabas CICS コマンドレベルリンクコンポーネントプログラムの名前に一致しているかをチェックします。フィールド LNCNAME が使用されるようになりました。このフィールドの値は、ADAGSET マクロの ENTPT キーワードによって決定されます。

LNCSTUB モジュールのエントリポイントの別名は、ALCSIVP でテストできます。SETC フィールド名 &STBNAME を使用して、記述されたエントリポイント名と一致するように SETC ステートメントのフィールド &STUBNM を変更します。



Note: ALCSIVP プログラムは、ADAGSET キーワード PARMTYP の値を使用して、CICS LINK コマンドで Adabas CICS リンクルーチンを起動するときに、処理の最初の部分で自身と Adabas CICS リンクルーチン間のデータの受け渡しに CICS TWA と CICS COMMAREA のどちらを使用するかを判定します。ALCSIVP がアセンブルされるときに使用された ADAGSET マクロで、PARMTYP=TWA と記述されていた場合、CICS TWA が使用されます。それ以外の場合は、CICS COMMAREA が EXEC CICS LINK コマンドで使用されます。

2 サンプルジョブストリーム JCLALCI の修正

- インストール確認プログラム ALCSIVP をプリプロセス、アセンブル、リンクするためにメンバ JCLALCI を使用します。ロードモジュールを CICS DFHRPL ライブラリ連結に挿入します。

- 環境に合わせてこの JCL を修正するには、メンバの JOB カードと手順 1 の表で示した記号の値を変更します（「[手順 1、メンバ JCLLNCS の修正](#)」参照）。

JCLALCI メンバは追加の記号パラメータ &CICSLIB を使用します。これは CICS RPL ライブラリの名前です。

- 修正したサンプル JCLALCI メンバを使用して ALCSIVP をプリプロセス、アセンブル、リンクします。
- CICS システムに次の RDO エントリを追加するか、RDO 機能を使用して ALCSIVP プログラムを実行するための STB1 トランザクションを追加します。

```
DEFINE PROGRAM(ALCSIVP) GROUP(ADABAS)
DESCRIPTION(ADABAS s ASSEMBLER IVP FOR HIGH-PERFORMANCE STUB)
LANGUAGE(ASSEMBLER) RELOAD(NO) RESIDENT(NO) USAGE(NORMAL)
USELPACOPY(NO) STATUS(ENABLED) CEDF(YES) DATALOCATION(ANY)
EXECKEY(USER) EXECUTIONSET(FULLAPI)

DEFINE TRANSACTION(STB1) GROUP(ADABAS)
DESCRIPTION(TRANSACTION TO EXECUTE THE ASSEMBLER IVP FOR HIGH-PERFORMANCE STUB)
PROGRAM(ALCSIVP) TWASIZE(32) PROFILE(DFHICST) STATUS(ENABLED)
TASKDATALOC(ANY) TASKDATAKEY(USER) STORAGECLEAR(NO)
RUNAWAY(SYSTEM) SHUTDOWN(DISABLED) ISOLATE(YES) DYNAMIC(NO)
PRIORITY(1) TRANCLASS(DFHTCL00) DTIMOUT(NO) INDOUBT(BACKOUT)
RESTART(NO) SPURGE(NO) TPURGE(NO) DUMP(YES) TRACE(YES)
RESSEC(NO) CMDSEC(NO)
```

- STB1 トランザクションを起動して ALCSIVP を実行します。ALCSIVP を実行すると LNCSTUB モジュールが確認されます。

COBOL IVP (COBSIVP) のインストールと実行

メンバ COBSIVP は Adabas DCI と COBOL プログラムの使用例です。CICS LINK/RETURN 機能から実行した一連の Adabas コールによって出力行が作成され、COBIVP がこれを画面に表示します。続いて、DCI を使用してこれらの Adabas コマンドを再実行します。

▶手順 4.5. COBOL IVP (COBSIVP) をインストールして実行するには、次の手順に従います。

- ソースメンバ COBSIVP の修正
 - フィールド WORK-DBID および WORK-FNR を修正して、使用する Software AG 提供のデモンストレーションデータベースの EMPLOYEES ファイルにアクセスするためのデータベース ID とファイル番号を VALUE 節に記述します。Software AG が提供するデモンストレーションファイルについての詳細は、z/OS インストール指示の「[デモンストレーションファイルのロード](#)」を参照してください。

- フィールド LINK-NAME の値が Adabas CICS コマンドレベルリンクコンポーネントプログラムで使用する名前に一致していることを確認します。
- 次のフィールドの値（PROCEDURE DIVISION のリテラル）が、使用している EMPLOYEES ファイルの FDT とデータ内容の要求に一致していることを確認します。

```
ADABAS-FORMAT-BUFFER,
ADABAS-SEARCH-BUFFER, and
ADABAS-VALUE-BUFFER
```

2 サンプルジョブストリーム JCLCOBI の修正

- インストール確認プログラム COBSIVP をプリプロセス、アSEMBル、リンクするためにメンバ JCLCOBI を使用します。環境に合わせて JCLCOBI を修正するには、メンバの JOB カードと次の表のシンボリックプロシージャ変数に指定する値を変更します。

値	説明
&ADALIB	ADASTWA ロードモジュールをリンケージエディタに指定するために使用する Adabas ロードライブラリの名前。
&MEM	処理するメンバ名。この場合は COBSIVP。
&CICSLIB	CICS 配下で実行するために COBSIVP ロードモジュールが挿入される CICS RPL ライブラリの名前。
&COBLIB	COBOL コンパイラ STEPLIB。
&INDEX	コンパイラ用の SYSLIB DD ステートメントで使用する CICS マクロライブラリに対するハイレベル修飾子。
&INDEX2	トランスレータ STEPLIB DD ステートメントおよびリンクステップの SYSLIB に使用する CICS ロードライブラリに対するハイレベル修飾子。
&LINKLIB	COBOL の LINKLIB。
&STBSRCE	配布した Adabas CICS ハイパフォーマンススタブ LNCSTUB が存在するソースライブラリの名前。
&STUBLIB	LNCSTUB ロードモジュールを挿入するためのロードライブラリの名前。このライブラリはアプリケーションライブラリをリンクするとき、使用可能である必要があります。
&SYSMSG	トランスレータメッセージの出力クラス。
&SYSOUT	SYSOUT メッセージと SYSPRINT メッセージの出力クラス。
&WORK	テンポラリデータセットとユーティリティデータセットに使用する DASD デバイスタイプ。

3 COBSIVP のプリプロセス、コンパイル、リンク

- 修正した JCLCOBI ジョブを使用して COBSIVP プログラムをプリプロセス、コンパイル、リンクします。COBOL プログラムをリンクするときに使用可能なライブラリに ADASTWA をアセンブルします。ADASTWA ロードモジュールを COBSIVP のリンクに組み込みます。

修正した JCLCOBI ジョブを使用して COBSIVP プログラムをプリプロセス、コンパイル、リンクします。COBSIVP は Adabas CICS リンクルーチンにデータを渡すときに CICS COMMAREA を使用するようになりました。そのため、バージョン 8 では、ADASTWA プログラムを COBSIVP とリンクする必要はありません。

LNCSTUB サブルーチンは、渡された Adabas パラメータを TWA に挿入するので、ADASTWA を使用しません。したがって、LNCSTUB モジュール経由で Adabas DCI を利用する COBOL アプリケーションをリンクするとき、ADASTWA ルーチンは必要ありません。

- COBSIVP プログラムを LNCSTUB ロードモジュールとリンクし、LNCSTUB ロードモジュールをリンケージエディタから利用できるようにして、COBSIVP ロードモジュールに組み込みます。



Note: CICS スタブモジュールはリンクステップでも解決されます。

- 4 CICS システムに次の RDO エントリを追加するか、RDO 機能を使用して COBSIVP プログラムを実行する STB2 トランザクションを追加します。

```
DEFINE PROGRAM(COBSIVP) GROUP(ADABAS)
DESCRIPTION(ADABAS s COBOL IVP FOR HIGH-PERFORMANCE STUB)
LANGUAGE(COBOL) RELOAD(NO) RESIDENT(NO) USAGE(NORMAL)
USELPACOPY(NO) STATUS(ENABLED) CEDF(YES) DATALOCATION(ANY)
EXECKEY(USER) EXECUTIONSET(FULLAPI)

DEFINE TRANSACTION(STB2) GROUP(ADABAS)
DESCRIPTION(TRANSACTION TO EXECUTE THE COBOL IVP FOR HIGH-PERFORMANCE STUB)
PROGRAM(COBSIVP) TWASIZE(32) PROFILE(DFHICST) STATUS(ENABLED)
TASKDATALOC(ANY) TASKDATAKEY(USER) STORAGECLEAR(NO)
RUNAWAY(SYSTEM) SHUTDOWN(DISABLED) ISOLATE(YES) DYNAMIC(NO)
PRIORITY(1) TRANCLASS(DFHTCL00) DTIMOUT(NO) INDOUBT(BACKOUT)
RESTART(NO) SPURGE(NO) TPURGE(NO) DUMP(YES) TRACE(YES)
RESSEC(NO) CMDSEC(NO)
```

- 5 STB2 トランザクションを起動して COBSIVP を実行します。COBSIVP を実行すると LNCSTUB モジュールが確認されます。

手順3：アプリケーションプログラムのリンクと実行

IVP プログラムが正常に実行されると、Adabas DCI で実際のアプリケーションプログラムを使用できる状態になります。手順3では、LNCSTUB サブプログラムを利用するためのアプリケーションプログラムインターフェイス (API) を作成します。

手順3では、次の作業を行います。

- 次のセクションで説明するガイドラインに従って、Adabas CICS ハイパフォーマンススタブルーチンを利用するアプリケーションプログラムを修正します。
- アプリケーションプログラムをプリプロセス、コンパイルまたはアセンブル、リンクして、LNCSTUB モジュールを組み込みます。
- Adabas CICS ハイパフォーマンススタブを使用して、アプリケーションプログラムを実行します。

アプリケーションプログラムを修正するためのガイドライン

LNCSTUB ロードモジュールをアプリケーションプログラムにリンクする必要があります。アプリケーションプログラムは標準のバッチ的なコールメカニズムを使用してDCIインターフェースを起動します。LNCSTUB モジュールはデータを Adabas CICS コマンドレベルリンクコンポーネントに渡すために必要な追加の CICS 要求を行います。

■ LNCSTUB がサポートするプログラミング言語

LNCSTUB は、アセンブラ言語、VS/COBOL、COBOL II、COBOL/LE、PL/I、および C で書かれたアプリケーションプログラムで動作します。

■ CICS トランザクションワークエリアの使用

Adabas DCI または Adabas CICS コマンドレベルリンクコンポーネントを使用するトランザクションは最低 28 バイトのトランザクションワークエリア (TWA) を提供できます。十分な TWA が提供できないと、アベンド U636 (タスクの異常終了) が発生します。

■ CICS COMMAREA の使用

Adabas バージョン 8 CICS リンクルーチンおよび Adabas 8 LNCSTUB モジュールでは、EXEC CICS LINK コマンドによるデータの受け渡しに、CICS COMMAREA を使用することを強くお勧めします。CICS COMMAREA は最低 32 バイト長で、かつ COMMAREA の先頭 8 バイトに文字列 "ADABAS52" または "ADABAS8X" が含まれている必要があります。文字列 "ADABAS8X" は、新しい Adabas バージョン 8 ACBX ダイレクトコールインターフェイスとそのパラメータリストを排他的に使用するアプリケーションに使用します。

■ リエントラント条件

アプリケーションプログラムはリエントラントでもリエントラントでなくてもかまいません。LNCSTUB モジュールは、リエントラント対応で記述されていますが、アプリケーションプログラムもリエントラントとしてマークしない限り、リンケージエディタパラメータを使用して、LNCSTUB ロードモジュールをリエントラントとしてマークすることはお勧めしません。

■ LNCSTUB が発行する CICS 要求

LNCSTUB モジュールは呼び出されると次のコマンドレベル CICS 要求を発行します。

```
EXEC CICS ADDRESS EIB
EXEC CICS LINK
```

Adabas カスタマイズレベルリンクにデータを渡すときに TWA を使用する場合は、次の CICS 要求を発行します。

```
EXEC CICS ADDRESS TWA
EXEC CICS ASSIGN TWALENG
```

■ DCI エントリポイントアドレス

Adabas CICS コマンドレベルリンクコンポーネントプログラムから DCI エントリポイントを取得するために、最低 1 回 EXEC CICS LINK コマンドが LNCSTUB によって発行されます。このアドレスはこのトランザクションの間、後続の全 Adabas コールの BALR アクセスで使用されます。したがって、コール元のアプリケーションプログラムはフルワード (4 バイト) フィールドを提供し、LNCSTUB に取得される DCI エントリポイントアドレスを保持する必要があります。この 4 バイトフィールドはコールメカニズムによって LNCSTUB モジュールに渡される最初のパラメータです。残りのパラメータは Adabas 要求を実行するために必要な Adabas パラメータリストで構成されます (バージョン 7 またはバージョン 8 のいずれかのパラメータリストを使用できます)。

■ DCI パラメータリスト

LNCSTUB プログラムからの Adabas DCI パラメータリストは、Adabas CICS コマンドレベルリンクコンポーネント内の DCI エントリポイントに対するポインタと、それに続く 6 つのポインタ (Adabas コントロールブロック、フォーマットバッファ、レコードバッファ、サーチバッファ、バリュースバッファ、ISN バッファ) で構成されています。

標準 Adabas コントロールブロックとバッファのコーディングについては、『*Adabas コマンドリファレンスマニュアル*』を参照してください。

Adabas 7 のパラメータリストのオフセットは次の表のとおりです。

オフセット	ポインタ
0	Adabas コマンドレベルリンクコンポーネントの DCI エントリポイント
4	Adabas コントロールブロック
8	Adabas フォーマットバッファ
12	Adabas レコードバッファ
16	Adabas サーチバッファ
20	Adabas バリュースバッファ

オフセット	ポインタ
24	Adabas ISN バッファ

先頭 (DCI エントリポイント) 以外のパラメータはすべて Adabas コールの必要条件に従ってアプリケーションプログラムに構築され、保守されます。

DCI エントリポイントパラメータはタスクの最初にバイナリのゼロに設定されます。その後アプリケーションプログラムでは修正しないでください。疑似リエントラント機能を維持するためにパラメータリストを構成するフィールドを CICS ストレージ (COBOL の場合 WORKING-STORAGE またはアセンブラの場合 DFHEISTG ユーザーストレージエリア) に配置することを強くお勧めします。

次の例は、アセンブラ言語プログラムの場合のサンプルパラメータリストです。

```
DFHEISTG DSECT
.
PARMLIST DS 0F
DS A(DCIPTR)
DS A(ADACB)
DS A(ADAFB)
DS A(ADARB)
DS A(ADASB)
DS A(ADAVB)
DS A(ADAIB)
.
DCIPTR DS F
ADACB DS CL80
ADAFB DS CL50
ADARB DS CL250
ADASB DS CL50
ADAVB DS CL50
ADAIB DS CL200
.
DFHEIENT CODEREG=(R12),EIBREG=(R10),DATAREG=(R13)
.
LA R1,PARMLIST
L R15,=V(LNCSTUB)
BALR R14,R15
.
END
```



Note: アセンブラ例の DFHEIENT マクロでは DATAREG パラメータにレジスタ 13 を使用しています。これは LNCSTUB プログラムの場合には必要条件です。LNCSTUB プログラムを呼び出すとき、レジスタ 13 は標準 CICS セーブエリア (DFHEISA) を、レジスタ 1 はパラメータリストをポイントしている必要があります。この標準を確実にする最良の方法は、例のように DFHEIENT マクロを使用したアセンブラアプリケーションをコーディングすることです。

次の例は、COBOL 言語プログラムの場合のサンプルパラメータリストです。

```
WORKING-STORAGE SECTION.  
.  
01 STUB-DCI-PTR PIC S9(8) COMP VALUE ZERO.  
01 ADACB PIC X(80).  
01 ADAFB PIC X(50).  
01 ADARB PIC X(250).  
01 ADASB PIC X(50).  
01 ADAVB PIC X(50).  
01 ADAIB PIC X(200).  
.  
PROCEDURE DIVISION.  
.  
CALL 'LNCSTUB' USING STUB-DCI-PTR,  
ADACB,  
ADAFB,  
ADARB,  
ADASB,  
ADAVB,  
ADAIB.  
.  
EXEC CICS RETURN END-EXEC.  
.  
GOBACK.
```

■ アプリケーションプログラムのコーディングに関する制限

他のすべての点で、アプリケーションプログラムは標準 CICS コマンドレベルルーチンと同様にコーディングしてください。LNCSTUB が呼び出されると DCI パラメータリストが正しいければ、アプリケーションが発行できる CICS コマンドに関する制限はありません。

■ 標準バッチコールメカニズムの使用

アセンブラおよび COBOL 言語プログラム例のパラメータリストで示したように LNCSTUB エントリポイントへのコールはバッチアプリケーションと同じように処理されます。同様に、サポートされる他の言語に対するコールは、その標準バッチコールメカニズムでコーディングしてください。

LNCSTUB モジュールを組み込むためのアプリケーションプログラムのリンク

LNCSTUB モジュールをアプリケーションプログラムと正しくリンクするには、アプリケーションをリンクして LNCSTUB モジュールと CICS スタブモジュールを組み込みます。これを行うための方式はアプリケーションに使用するプログラミング言語によって異なります。

■ アセンブラ言語プログラムには DFHEAI と DFHEAI0 CICS モジュールを組み込む必要があります。

■ COBOL アプリケーションには DFHECI と DFHEAI0 を組み込む必要があります。

DFHEAI0モジュールへの二重の参照を回避するために、リンケージエディタ REPLACEDFHEAI0 コントロールステートメントを SYSLIN データの先頭に指定します。

▶手順 4.6. アセンブラ言語プログラムをリンクする場合

- アセンブラプログラムの場合、SYSLIN 入力は次のようになります。

```
INCLUDE DFHEAI
```

アセンブラオブジェクト入力は次のようになります。

```
REPLACE DFHEAI0  
INCLUDE SYSLIB(LNCSTUB)  
INCLUDE SYSLIB(DFHEAI0)  
NAME ALCSIVP(R)
```

リンケージエディタからクロスリファレンスを検査するとき、シンボル entry-name はリンクマップの LNCSTUB モジュールと同じ開始位置にする必要があります。

▶手順 4.7. COBOL 言語プログラムをリンクする場合

- COBOL プログラムの場合、SYSLIN 入力は次のようになります。

```
REPLACE DFHEAI0  
INCLUDE SYSLIB(DFHECI)
```

COBOL オブジェクト入力は次のようになります。

```
INCLUDE SYSLIB(LNCSTUB)  
INCLUDE SYSLIB(DFHEAI0)  
NAME COBSIVP(R)
```

リンケージエディタからクロスリファレンスを検査するとき、シンボル entry-name はリンクマップの LNCSTUB モジュールと同じ開始位置にする必要があります。

▶手順 4.8. PL/I および C 言語プログラムをリンクする場合

- CICS 配下の PL/I と C アプリケーションのリンクについての情報は IBM マニュアル『CICS System Definition Guide』を参照してください。

LNCSTUB を使用した場合のパフォーマンス

Adabas DCI を使用するアプリケーションから最高のパフォーマンスを得るには、DCI インターフェースの動作を論理レベルで確認します。

Adabas リンクルーチンへのアクセスに標準 LINK/RETURN メカニズムを使用する CICS アプリケーションは、すべての Adabas からのリンクルーチンに対する要求で CICS プログラムコントロールサービスをコールします。LNCSTUB モジュールによって BALR インターフェースの使用が許可されます。BALR インターフェースは、アプリケーションプログラムから Adabas CICS コマンドレベルリンクコンポーネントに制御を渡すときに発生する CICS オーバーヘッドを減らします。

LNCSTUB モジュールは、Adabas CICS コマンドレベルリンクルーチンへのイニシャルコール (IC) を行うための標準 EXECCICS LINK/RETURN を使用してこれを実行します。リンクルーチンはこのコールを認識し、DCI サブルーチンのエン트리ポイントアドレスを LNCSTUB に返します。次に、LNCSTUB はこのアドレスを、呼び出したタスクの存続中に確実に存在する場所に保存する必要があります。これはコール元プログラムが DCI エントリーポイントアドレスを保持する 4 バイトフィールドを提供しなければならないためです。DCI アドレスを取得した後で、このアドレスを後続の Adabas コールで渡される最初のパラメータとして受け取ると、LNCSTUB は BALR インターフェースを利用して、Adabas CICS コマンドレベルリンクコンポーネントプログラムに制御を渡します。

このロジックによって、IC 間の Adabas 要求が多くなった場合に、CICS 配下の Adabas とアプリケーション間のデータの受け渡しに関してアプリケーションが効率的になります。実際、タスクが呼び出されるたびに 1 つの Adabas コールを発行する擬似会話型アプリケーションでは、DCI を使用しないでください。なぜなら、コール元プログラムが Adabas コマンドを発行するたびに IC 要求が発生するからです。

Adabas CICS コマンドレベルリンクコンポーネントプログラムを常駐するように CICS に定義することによって、さらにパフォーマンス向上を実現できます。これにより、DCI エントリーポイントを CICS タスク全体に保存でき、さまざまなプログラムから有効な DCI エントリーポイント付きの LNCSTUB モジュールをコールできるようになります。これにより、各プログラム開始時の IC は回避されます。ただし、この手順を使用した場合には、エン트리ポイントアドレスを無効にする CICS 環境への変更 (NEWCOPY など) は、予期しない結果や損害をもたらすことがあります。

CICS サービスを使用する Adabas CICS コマンドレベルリンクコンポーネントプログラムには、1 つ以上の IC を作成することが必須です。このコールは Adabas ユーザーブロック (UB) およびレジスタセーブエリアの配列に対する共通ストレージの取得をトリガするために使用されます。IC 要求を行わない場合、Adabas コールは、ワーキングストレージが不足しているため、またリンクルーチンで使用する重要なコントロールブロックや Adabas SVC エントリが作成されていないために実行されません。

ソースメンバデフォルトの修正 (ADAGSET マクロ)

 **Caution:** Adabas 8 では、Adabas 8 ACIvrn.SRCE ライブラリの ADAGSET マクロは、Adabas 8 CICS ハイパフォーマンススタブルーチンのデフォルト値を生成するためだけに使用してください。

Adabas CICS ハイパフォーマンススタブルーチンのアセンブリを容易にするために、環境に特有のデフォルト値を持つ ADAGSET マクロをプログラミングし、アセンブリ時にそれを SYSLIB 連結に有効なソースライブラリに入れることを推奨します。

ADAGSET パラメータオプションとそのデフォルト値 (下線付き) の説明は次のとおりです。

- AVB : Adabas VSAM Bridge のサポート
- ENABNM : Adabas TRUE を有効にするプログラムのエントリポイント名
- ENTPT : Adabas CICS コマンドレベルリンクルーチンの名前
- LOGID : デフォルトの論理データベース ID
- LRINFO : Adabas Review データエリアの長さ
- LUINFO : Adabas UEXITA および UEXITB に渡されるユーザーデータの長さ
- LUSAVE : Adabas UEXITA および UEXITB のユーザーセーブエリアのサイズ
- LXITAA : UEXITA に提供するワークエリアの長さ
- LXITBA : UEXITB のワークエリアの長さ
- MRO : マルチプルリージョンオプション
- NETOPT : ユーザー ID の作成に使用されたメソッド
- NTGPID : Natural グループ ID
- NUBS : CICS リンクルーチンに作成されるユーザーブロック数
- PARMTYP : Adabas パラメータリストのエリア
- PURGE : トランザクションの削除
- RMI : リソースマネージャインターフェイス
- SAF : Adabas SAF Security
- SAP : SAP アプリケーションのサポート
- SVCNO : Adabas SVC 番号
- TRUE : Adabas タスク関連ユーザー出口
- TRUENM : Adabas タスク関連ユーザー出口の名前
- UBPLC : ユーザーブロックプールの割り当て

■ XWAIT : CICS の XWAIT 設定

AVB : Adabas VSAM Bridge のサポート

パラメータ	説明	構文
AVB	<p>このコマンドレベルリンクルーチンで Software AG の Adabas Bridge for VSAM をサポートするかどうかを示します。</p> <ul style="list-style-type: none"> ■ AVB=YES の場合、Adabas VSAM Bridge はサポートされます。 ■ AVB=NO の場合、Adabas VSAM Bridge はサポートされません。 	AVB={ <u>NO</u> YES }

ENABNM : Adabas TRUE を有効にするプログラムのエントリポイント名

パラメータ	説明	構文
ENABNM	<p>CICS PLTPI 処理中に Adabas TRUE を有効にするために実行されるプログラムに対するエントリポイント名。値は、使用環境の DFHPLT テーブルに指定されたモジュール名と一致している有効なプログラム名でなければなりません。デフォルト値は ADAENAB です。</p> <p>TRUE=NO を指定する場合、このパラメータは無視されます。</p>	ENABNM={ ' <u>ADAENAB</u> ' 'name' }

ENTPT : Adabas CICS コマンドレベルリンクルーチンの名前

パラメータ	説明	構文
ENTPT	<p>Adabas CICS ハイパフォーマンススタブリックルーチンの名前。この名前を EXEC CICS LINK コマンドに使用して CICS アプリケーションプログラムから Adabas サービスを呼び出します。</p> <p>インストール手順の注 1 と注 2 も参照してください。</p>	ENTPT={ ' <u>ADACICS</u> ' 'name' }

LOGID : デフォルトの論理データベース ID

パラメータ	説明	構文
LOGID	デフォルトの論理データベース ID の値。有効な ID 番号は 1~65535 です。	LOGID= <i>nnn</i>

LRINFO : Adabas Review データエリアの長さ

パラメータ	説明	構文
LRINFO	REVEXITB プログラムに使用される Adabas Review データエリアの長さ (バイト)。指定しない場合は 0 (デフォルト) になります (Adabas Review は使用されません)。最小 (および推奨) 値は 256 です。この値は、REVEXITB プログラムが呼び出されたときに Adabas Review で必要になるサイズです。詳細については、Adabas Review のドキュメントを参照してください。	LRINFO={ 0 256 }

LUINFO : Adabas UEXITA および UEXITB に渡されるユーザーデータの長さ

パラメータ	説明	構文
LUINFO	CICS リンクルーチンから Adabas UEXITA と UEXITB に渡されるユーザーデータの長さ。 LUINFO を指定しない場合、0 (デフォルト) になり、ユーザーセーブエリアは渡されません。	LUINFO={ 0 length }

LUSAVE : Adabas UEXITA および UEXITB のユーザーセーブエリアのサイズ

パラメータ	説明	構文
LUSAVE	Adabas ユーザー出口 UEXITA と UEXITB に使用されるユーザーセーブエリアのサイズ。LUSAVE を指定する場合は、72 以上の値を指定する必要があります。 LUSAVE を指定しない場合、0 (デフォルト) になり、ユーザーデータは渡されません。	LUSAVE={ 0 size }

LXITAA : UEXITA に提供するワークエリアの長さ

パラメータ	説明	構文
LXITAA	UEXITA ユーザー出口プログラムに提供するワークエリアの長さ。 指定できる値は 0 (デフォルト) ~32767 です。0 は、UEXITA プログラムが Adabas コマンドレベルリンクルーチンにリンクされていないため、UEXITA にデータが渡されないことを示します。 注意: このパラメータはまだ完全にインプリメントされていません。LNKOLM とリンクされた CICS ユーザー出口 A プログラムでの将来の使用のために提供されています。	LXITAA={ 0 nn }

LXITBA : UEXITB のワークエリアの長さ

パラメータ	説明	構文
LXITBA	<p>UEXITB ユーザー出口プログラムに提供するワークエリアの長さ。</p> <p>指定できる値は0 (デフォルト) ~32767 です。0 は、UEXITB プログラムが Adabas コマンドレベルリンクルーチンにリンクされていないため、UEXITB にデータが渡されないことを示します。</p> <p>注意: このパラメータはまだ完全にインプリメントされていません。LNKOLM とリンクされた CICS ユーザー出口 A プログラムでの将来の使用のために提供されています。</p>	LXITBA={ 0 nn }

MRO : マルチプルリージョンオプション

パラメータ	説明	構文
MRO	<p>MRO パラメータは、CICS マルチプルリージョンオプションを使用するかどうかを指定します。</p> <p>CICS マルチプルリージョンオプション (MRO) とともに CICS コマンドレベルリンクを実行する場合、MRO=YES をセットします。そうでないと、デフォルト値 MRO=NO を使用します。</p> <p>MRO=YES の場合、NETOPT に NETOPT=NO (デフォルト) をセットし、マルチプルアプリケーションリージョンからユニークでない LU 名を回避しなければなりません。</p> <p>NETOPT=YES および MRO=YES のとき、アセンブラ MNOTE およびリターンコード 16 がアセンブリステップから出力されます。</p>	MRO={ NO YES }

NETOPT : ユーザー ID の作成に使用されたメソッド

パラメータ	説明	構文
NETOPT	<p>NETOPT=YES が指定されている場合は、8 バイトのユーザー ID が VTAM LU 名から構成されます。NETOPT=NO が指定されている場合、ユーザー ID は定数 CICS に端末タスク用の 4 バイトの CICS 端末 ID (TCTTETI) で作られます。非端末タスクの場合、ユーザー ID は定数 CIC と CICS タスク番号で構成されます。</p> <p>CICS マルチプルリージョンオプション (MRO) 付きで実行する場合、このオプションに対してデフォルト値を使用する必要があります。NETOPT=YES および MRO=YES のとき、アセンブラ MNOTE およびリターンコード 16 がアセンブリステップから出力されます。</p>	NETOPT={ NO YES }

NTGPID : Natural グループ ID

パラメータ	説明	構文
NTGPID	<p>このパラメータは、必要に応じて、Natural バージョン 2.2.8 以降を使用して、固有の Adabas ユーザー ID を CICSplex 環境で生成するために 4 バイトの Natural グループ ID の指定に使用します。値は、指定値でアセンブルされた Adabas コマンドレベルリンクルーチンをコールするすべてのユーザーに関連します。</p> <p>デフォルト値はありません。値が指定されない場合、従来どおり Adabas 内部ユーザー ID が作成されます。</p> <p>4 バイトの英数字の値を指定することもできますが、CICSplex または z/OS イメージで実行している Adabas コマンドレベルリンクルーチンごとに固有である必要があります。複数の NTGPID が必要なとき（テストと本番の Natural 2.2.8 など）、TRUE に関連する複数の Adabas コマンドレベルリンクルーチンを生成する必要があります。</p> <p>CICS マルチプルリージョンオプション（MRO）付きで実行する場合、NTGPID を使用して、マルチプルアプリケーションリージョンが Adabas をコールするとき、Adabas SVC に使用される Adabas コミュニケーション ID 用の 4 バイトリテラルを指定します。</p>	NTGPID=4-byte-value

NUBS : CICS リンクルーチンに作成されるユーザーブロック数

パラメータ	説明	構文
NUBS	<p>CICS リンクルーチンで作成されるユーザーブロック（UB）数。ブロック数は同時に発生することが予想される最大 Adabas 要求数を処理可能な大きさにする必要があります。</p> <p>注意: Adabas 6.2 以上のコマンドレベルリンクルーチンは、16 MB 境界より上にユーザーブロック（UB プール）用のストレージを取得します。</p>	NUBS={ <u>50</u> blocks }

PARMTYP : Adabas パラメータリストのエリア

パラメータ	説明	構文
PARMTYP	<p>Adabas パラメータリストを取得するためのエリア。TWA は、トランザクションワークエリア（TWA）の先頭の 6 フルワードのパラメータリストを取得します。PARMTYP=COM の場合、Adabas パラメータが EXEC CICS LINK コマンドを伴ったコール元プログラムからコールされた CICS COMMAREA に指定されます。ACB コールの COMMAREA リストは最低 32 バイト必要で、ラベル "ADABAS52" で始まります。PARMTYP=ALL（デフォルト）は COMMAREA と TWA の両方を使用して Adabas パラメータを渡します。この場合、COMMAREA が最初にチェックされます。</p>	PARMTYP={ <u>ALL</u> COM TWA }

パラメータ	説明	構文
	TRUE=YES オプションが指定されている場合は、PARMTYP=ALL または PARMTYP=COM を使用する必要があります。	

PURGE：トランザクションの削除

パラメータ	説明	構文
PURGE	<p>CICS 3.2 以上でアSEMBルするとき、PURGE パラメータが使用されます。PURGE=YES を指定すると、CICS WAIT EXTERNAL はそのパラメータの 1 つとして PURGEABLE を含むので、DTIMOUT 値を超えたときに PURGE が指定されると、CICS はトランザクションを削除できるようになります。</p> <p>PURGE=NO (デフォルト) が指定されている場合は、NONPURGEABLE オプションが生成されます。</p>	PURGE={ NO YES }

RMI：リソースマネージャインターフェイス

パラメータ	説明	構文
RMI	<p>RMI パラメータは、CICS リソースマネージャインターフェイスを使用するかどうかを示すために使用します。</p> <p>RMI=YES が指定されている場合は、CICS リソースマネージャインターフェイス (RMI) を使用して、Adabas タスク関連ユーザー出口 (TRUE) がリソースマネージャ (RM) として実行されます。</p> <p>RMI=YES は、Adabas Transaction Manager がインストールされ、有効で、CICS 環境で実行しているユーザーに利用可能なときにだけ有効です。Adabas TRUE のインストールと関連する付加的な手順については Adabas Transaction Manager のドキュメントを参照してください。</p>	RMI={ NO YES }

SAF：Adabas SAF Security

パラメータ	説明	構文
SAF	<p>Adabas SAF Security (ADASAF) を使用するかどうかを示します。ADASAF を使用している場合は、SAF=YES を設定する必要があります。</p> <ul style="list-style-type: none"> ■ YES の場合、Adabas SAF Security を使用します。 ■ NO の場合、Adabas SAF Security は使用しません。 <p>CICS/TS 1.1 以上で実行するとき、ADASAF は Adabas タスク関連ユーザー出口 (TRUE) を必要とします。SAF=YES で TRUE=YES のとき、タスク関連ユーザー出口はユーザーの外部セキュリティ ID (サインオン) を Adabas に渡します。</p>	SAF={ NO YES }

パラメータ	説明	構文
	<p>この場合 TRUE=YES を指定しないと、ADAGSET マクロは LNKOLSC、LNKTRUE、または LNKENAB アセンブリ処理を MNOTE で終了し、コード 16 を返します。</p> <p>CICS/ESA 3.3 以下で ADASAF を実行している場合、TRUE=YES は必要ありません。この場合、SAF=YES と TRUE=NO の組み合わせは有効です。</p>	

SAP : SAP アプリケーションのサポート

パラメータ	説明	構文
SAP	<p>SAP パラメータは、SAP アプリケーションシステムに Adabas サポートが必要かどうかを指定するために使用されます。</p> <p>SAP=YES を指定すると、LNKOLSC プログラムは SAP 初期化コールを検出し、初期化コールに提供された定数にフィールド ACBADD2 を合わせて SAP アプリケーションのためのユーザー ID をセットします。</p> <p>詳細については、SAP アプリケーションシステムを使用するお客様のために提供されている補足情報を参照してください。</p>	SAP={ NO YES }

SVCNO : Adabas SVC 番号

パラメータ	説明	構文
SVCNO	SVCNO パラメータは、Adabas SVC 番号の値を指定するために使用します。	SVCNO={ 0 nnn }

TRUE : Adabas タスク関連ユーザー出口

パラメータ	説明	構文
TRUE	<p>TRUE パラメータは、Adabas タスク関連ユーザー出口を使用するかどうかを示すために使用します。</p> <p>TRUE=YES を指定する場合、LNKOLSC は Adabas タスク関連ユーザー出口 LNKTRUE を使用します。</p> <p>TRUE=YES を指定する場合、PARMTYP={ ALL COM } と TRUENM='name' のパラメータ設定も指定する必要があります。</p>	TRUE={ NO YES }

TRUENM : Adabas タスク関連ユーザー出口の名前

パラメータ	説明	構文
TRUENM	<p>TRUENM パラメータは、Adabas タスク関連ユーザー出口の名前を指定するために使用します。</p> <p>TRUE=YES を指定する場合、このパラメータが必要です。</p> <p>インストール手順の注 1 と注 2 も参照してください。</p>	<pre>TRUENM= { 'name' 'ADATRUE' }</pre>

UBPLOC : ユーザーブロックプールの割り当て

パラメータ	説明	構文
UBPLOC	<p>UBPLOC パラメータは、ユーザーブロック (UB) プールを CICS の 16 MB 境界より上 (デフォルト) に取得するか、または下に取得するかを指定します。</p> <p>EXEC CICS WAIT WAITCICS または EXEC CICS WAIT EXTERNAL によって使用される ECB を UP プールに組み込みます。</p> <p>UBPLOC=BELOW を設定すると、ECB を 16 MB 境界より上に許可しない CICS バージョン、つまり、CICS/ESA 3.2 以下をサポートします。</p> <p>詳細については、IBM マニュアル『<i>CICS Application Programming Reference</i>』を参照してください。</p>	<pre>UBPLOC= { ABOVE BELOW }</pre>

XWAIT : CICS の XWAIT 設定

パラメータ	説明	構文
XWAIT	<p>XWAIT パラメータは、LNKOLSC モジュールのアセンブラ処理によってコマンドレベルリンクコンポーネントに、標準 EXEC CICS WAITCICS (XWAIT=NO) または WAIT EVENTS EXTERNAL (XWAIT=YES) のどちらを生成するかを指定するために使用します。XWAIT=YES がデフォルトです。</p> <p>CICS/TS 1.1 以上では CICS WAIT EVENTS EXTERNAL (XWAIT=YES) が望ましいインターフェイスです。</p> <p>CICS WAITCICS ステートメント (XWAIT=NO) は、CICS/MVS 2.1.2 および CICS/VSE 2.1~2.3 での使用のために提供されています。これは、CICS/TS 1.1 以上でも使用できますが、CICS トランザクションのパフォーマンスを低下させるか、またはビジーな CICS 環境では予期しない結果を招くことがあります。</p> <p>注意: CICS/ESA 3.3 配下で XWAIT=NO を指定する場合、IBM APAR PN39579 を CICS/ESA 3.3 システムに適用しなければなりません。CICS/TS 1.1 以上では、この APAR は必要ありません。</p>	<pre>XWAIT={ NO YES }</pre>

**Notes:**

1. IBM 仕様と一致させるため、XWAIT パラメータのデフォルトは XWAIT=YES になります。
2. XWAIT=NO を指定すると、LNKOLSC モジュールは、EXEC CICS WAIT EVENT コマンドの代わりに EXEC CICS WAITCICS コマンドを発行します。この指定を行うと、CICS/TS バージョン 1.1 以上を搭載した高トランザクション処理の CICS システムにおける WAIT リストと ECB リストの使用方法が、推奨される IBM 仕様と同じになります。
3. EXEC CICS コマンドはすべて CICS プリプロセッサによって処理されます。ADAGSET パラメータにより、後続のアセンブリステップはいくつかのステートメントをスキップします。

XWAIT ポスティングメカニズム

CICS WAITCICS (XWAIT=NO) は、指定された ECB のソフトポストをサポートすることができます。これには、ハンドポストダブルワークは全ワークサイクル上で CICS に処理されないため、優先度の低い CICS/TS 環境のディスパッチ可能なワークユニットになるという短所があります。

これに反して、EXEC CICS WAIT EXTERNAL (XWAIT=YES) により、CICS がその特別ポスト出口コードを使用できるようにし、全 CICS ワークサイクル上でチェックと処理（ポストされた場合）を常に行います。

各種の CICS WAIT コマンドとそのハードおよびソフトポスティングメカニズムの関係の違いの詳細については、IBM の『CICS/ESA Application Programming Reference Guide』および添付テキストの IBM APAR PN39579 または IBM InfoLink サービスの Item RTA000043874 を参照してください。

XWAIT および Adabas SVC / ルーター

Adabas SVC は XWAIT=YES 設定と完全な互換性があります。SVC は、Adabas 6 コマンドレベルリンクルーチンを使用する CICS/TS 配下の Adabas コール元に対して必要なハードポストを実行します。ハードポストを必要としないバッチコール元に対しては、同じ SVC がソフトポストを実行します。

Adabas 8 配下での Com-plete を使用する場合の Adabas のインストール

Adabas をインストールするとき、Com-Plete (Software AG 提供の TP モニタ) には一定のパラメータが必要です。詳しくは『Com-plete System Programming』を参照してください。

Software AG の TP モニタ Com-plete が Adabas データベースとコミュニケーションする場合は、Adabas リンクルーチンが必要です。Software AG の Entire Net-Work 製品を使用するか、Com-plete 配下で動作する Entire System Server のような製品を使用してください。現在のところ Com-plete は Adabas 7 と Adabas 8 のリンクルーチンの混在をサポートしていないため、Com-plete は、Adabas 7 リンクルーチンか Adabas 8 リンクルーチンのどちらか一方で実行する必要があります。

Adabas バージョン 8 リンクルーチンは、Adabas 8 z/OS ロードライブラリのメンバ ADALCO で提供されています。このメンバは、準備したリンクグローバルモジュールとリンクする必要があります。また、Com-plete の初期化時に Com-plete によってロードされる最終的な ADALCO ロードモジュールを作成するために必要なすべてのリンクルーチン出口とリンクする必要があります。リンクする最終的な ADALCO ロードモジュールと出口はリエントラントである必要があります。

次の表に、Adabas 8 配下で Com-plete を使用する場合の Adabas のインストールをサポートするための、Adabas インストールで提供されるモジュールを示します。

モジュール	説明
ADALCO8	基本モジュール
ADALCO	実行可能なデフォルトのモジュール

▶ **手順 4.9. Adabas リンクルーチンを準備するには、次の手順に従います。**

- 1 Adabas 8 ADAvrs.SRCE に含まれているサンプルメンバ LCOGBL ライブラリを修正可能な任意のユーザーソースライブラリにコピーします (vrs はテープで提供される Adabas の最新バージョン番号)。LCOGBL は、LGBLSET パラメータを含むモジュールです。このパラメータはコマンドレベルリンクコンポーネントのデフォルト設定を作成するために使用します。LGBLSET パラメータの詳細については、「[バージョン 8 でのソースメンバデフォルトの修正 \(LGBLSET マクロ\)](#)」に記載されています。
- 2 ユーザーソースライブラリの LCOGBL メンバを修正します。

LCOGBL では、少なくとも次の LGBLSET パラメータの値を指定する必要があります。

パラメータ	指定する内容
LOGID	デフォルトデータベースまたはターゲット ID。"1"~"65535" の範囲の数値で指定します。デフォルト値は "1" です。 注意: Com-plete で LOGID および SVCNO のデフォルト値を指定する場合、ドキュメンテーションのみが対象になります。ADASVC Com-plete ランタイムコントロールステートメントでは、実行中の Com-plete で使用される、サポートされるデータベース ID と AdabasSVC を組み合わせることができます。詳細については、Com-plete のドキュメントを参照してください。
SVCNO	デフォルトの AdabasSVC 番号。z/OS の場合、この番号の範囲は "200"~"255" です。 注意: Com-plete で LOGID および SVCNO のデフォルト値を指定する場合、ドキュメンテーションのみが対象になります。ADASVC Com-plete ランタイムコントロールステートメントでは、実行中の Com-plete で使用される、サポートされるデータベース ID と AdabasSVC を組み合わせることができます。詳細については、Com-plete のドキュメントを参照してください。

パラメータ	指定する内容
OPSYS	Com-plete が実行されるオペレーティングシステムの 3 文字の省略形。有効な値は "ZOS" や "VSE" などです。 注意: OPSYS パラメータは ZOS に設定する必要があります。
TPMON	COM。このキーワードは、3 文字の TP モニタの省略形を指定します。Com-plete の場合、この値は "COM" にする必要があります。
RENT	YES。このキーワードは、モジュールが連続的にリエントラントかどうかを示します。Com-plete の場合、この値は "YES" にする必要があります。
GEN	CSECT。このキーワードは、CSECT と DSECT のどちらを生成するかを指定します。リンクルーチンググローバルロードモジュールとしてリンク可能なオブジェクトモジュールを生成するには、CSECT を指定する必要があります。
UES	Adabas ユニバーサルエンコーディングサポート (UES) を有効にするかどうかを指定します。デフォルトは YES です。詳細については、「 UES 対応データベースの接続 」を参照してください。
出口パラメータ	その他の出口をアクティブにするかどうか、ユーザー出口を使用する場合は、ユーザー出口モジュール名を指定します。この情報を LGBLCOM の他のパラメータで指定する方法については、「 バージョン8でのソースメンバデフォルトの修正 (LGBLSET マクロ) 」を参照してください。

- 3 サンプルジョブ ASMGBLS を、ジョブの先頭で示されているとおりに修正および実行します。ASMGBLS は Adabas 8 ADA_{vrs}.JOBS ライブラリ内にあります。完全に修正する場合、ジョブの SET ステートメントが前の手順で準備した LCOGBL メンバを参照し、NAME リンクエディットコントロールステートメントが LCOGBL メンバの GBLNAME パラメータによって指定される名前を参照している必要があります。

修正したら、ASMGBLS ジョブを実行してリンクグローバルモジュールをアセンブルおよびリンクエディットします。

新しいリンクグローバルモジュールが、ASMGBLS ジョブで指定されたユーザーロードライブラリに生成されます。モジュール名は LCOGBL の GBLNAME パラメータで指定された名前になります。

- 4 サンプルジョブ LNKLCO8 をユーザーソースライブラリにコピーし、前の手順で作成した新しいリンクグローバルモジュールおよびその他の必要な出口が ADALCO8 基本モジュールとリンクするように修正します。サンプルジョブの修正手順は、ジョブの先頭に記載されています。このジョブの出力先を適切なユーザーロードライブラリに設定するようにしてください。LNKLCO8 は Adabas 8 ADA_{vrs}.SRCE ライブラリ内にあります。

このジョブで生成されるモジュールは ADALCO と呼ばれます。

- 5 Com-plete を開始するジョブステップで使用できるロードライブラリに ADALCO モジュールを配置します。

これで Adabas 8 リンクルーチンの準備は完了です。

バッチ／TSO を使用する場合の Adabas インストールの全般的な考慮事項

TSO システムに Adabas 8 をインストールする場合、Adabas-TSO 間のコミュニケーションは、バッチリンクルーチン ADALNK8（非リエントラント）および ADALNKR8（リエントラント）によって行われます。

Adabas バージョン8.1 の ADALNK は配布時のままで UES に対応しています。詳細は、「[UES 対応データベースの接続](#)」セクションを参照してください。

ただし、ADAUSER とリンクしたユーザープログラムも ADARUN をロードすることに注意してください。また、ADARUN は他のモジュールをロードします。

ADAUSER とリンクしたユーザープログラムを起動するには、次のモジュールが、特定の TSO ユーザーに対して定義されたロードライブラリから実行時にすべて使用できるようにする必要があります。

```
ADAIOR ADAMLF
ADAIOS ADAPRF
ADALNK ADARUN
```

このsectionでは、次のトピックについて説明します。

- [非リエントラント ADALNK バッチルーチンの動作](#)
- [ADALNKR：リエントラントバッチリンクルーチン](#)

非リエントラント ADALNK バッチルーチンの動作

Adabas 8 ロードライブラリの ADALNK モジュールは、次の条件を満たす場合に、Adabas 7 互換の方法で動作します。

- コール元のアプリケーションが ADAUSER とリンクされている。コール元のアプリケーションが ADAUSER とリンクされていない場合、ADALNK は機能しません。
- 最新の Adabas 8 ロードライブラリの ADARUN モジュールが使用されている。
- データベース ID と Adabas SVC 番号が、DD ステートメントで入力として指定されている。それ以外の場合、リンクグローバルテーブルの値によってこれらの値が上書きされます。

これら3つの条件をすべて満たす場合、デフォルトデータベース ID と Adabas SVC 番号は、DD ステートメント入力指定された値によって上書きされ、ADARUN によってリンクルーチンに渡されます。

この方法で動作させる場合、データベース管理者（DBA）とアプリケーションプログラマによる改変が最小限で済みます。Adabas ユーティリティを実行する場合にも、この動作モードをお勧めします。

ADALNKR：リエントラントバッチリンクルーチン

Software AG のいくつかの製品では、リエントラントバッチリンクルーチンを使用する必要があります。これらの製品をサポートするために、Adabas ロードライブラリで ADALNKR ロードモジュールが提供されています。Adabas 8 ADALNKR ソースモジュールは提供されません。

これらのリエントラントバッチリンクルーチンのデフォルト値を変更できます。詳細については、次のセクションを参照してください。

- **Adabas 8 ADALNK または ADALNKR モジュールのデフォルト値の変更**
- **Adabas 8 ADALNK または ADALNKR モジュールのデフォルト値の ZAP**

バッチアプリケーションプログラムには ADALNK や ADALNKR ではなく ADAUSER モジュールをリンクすることをお勧めします。ADAUSER ロードモジュールはリエントラントではありませんが、アプリケーションプログラムが「バッチ/TSO リエントラント ADALNKR8 モジュールのワークエリア」で解説されているコール要件を満たし、DDCARD 入力でキーワードパラメータ PROG=USER の代わりに PROG=RENTUSER ADARUN パラメータが指定されている場合に限り、リエントラントリンクルーチンモジュールとリンクすることができます。

最新の Adabas 8 ADALNKR モジュールを使用して、バッチまたは TSO でリエントラント操作を実行するには、事前に ADALNKR モジュールを準備する必要があります。データベース ID、Adabas SVC 番号、およびその他の要件のデフォルトを指定する、修正したリンクグローバルテーブルとリンクする必要があります。リエントラント出口もすべてリンクが必要です。

Adabas 8 配下での Batch/TSO を使用する場合の Adabas のインストール

Adabas 8 を TSO システムにインストールする場合、Adabas/TSO コミュニケーションには標準 Adabas 8 バッチリンクルーチン（ADALNK）が使用されます（SMA ジョブ番号 I056）。

この section では、次のトピックについて説明します。

- **提供されるモジュール**
- **Adabas 8 ADALNK または ADALNKR モジュールのデフォルト値の変更**

■ Adabas 8 ADALNK または ADALNKR モジュールのデフォルト値の ZAP

提供されるモジュール

次の表は、Adabas 8 配下でバッチ/TSO 使用の Adabas インストールをサポートするための、Adabas インストールで提供されるモジュールのリストです。

モジュール	説明
ADALNK8	基本モジュール
ADALNKR8	基本リエントラントモジュール
ADALINK	実行可能なデフォルトのモジュール
ADALNKR	実行可能なデフォルトのリエントラントモジュール

Adabas 8 ADALNK または ADALNKR モジュールのデフォルト値の変更

Adabas 8 ADALNK および ADALNKR モジュールで使用されるさまざまなリンクルーチンパラメータのデフォルト値を変更できます。

▶手順 4.10. デフォルト値を変更するには、次の手順に従います。

- 1 Adabas 8 ADA_{vrs}.SRCE ライブラリ (*vrs* はテープで配布される最新の Adabas バージョン番号) において提供されるサンプルメンバ LNKGBLS (非リエントラントリンクの場合) または LNKRGBL (リエントラントリンクの場合) メンバを、修正可能な任意のユーザーソースライブラリにコピーします。これらのモジュールには、リンクコンポーネントのデフォルト設定を作成するのに使用する LGBLSET パラメータが含まれています。LGBLSET パラメータの詳細については、「バージョン 8 でのソースメンバデフォルトの修正 (LGBLSET マクロ)」に記載されています。
- 2 ユーザーソースライブラリの LNKGBLS または LNKRGBL メンバを修正します。インストール要件に合わせて、LOGID、SVC、その他のキーワードの値を指定します。



Note: OPSYS パラメータは ZOS に設定する必要があります。

- 3 サンプルジョブ ASMGBLS を、ジョブの先頭で示されているとおりに修正および実行します。ASMGBLS は Adabas 8 ADA_{vrs}.JOBS ライブラリ内にあります。完全に修正する場合、ジョブの SET ステートメントが、前の手順で準備した LNKGBLS または LNKRGBL メンバを参照し、NAME リンクエディットコントロールステートメントが LNKGBLS または LNKRGBL メンバの GBLNAME パラメータによって指定される名前を参照している必要があります。

修正したら、ASMGBLS ジョブを実行してリンクグローバルモジュールをアセンブルおよびリンクエディットします。

新しいリンクグローバルモジュールが、ASMGBLS ジョブで指定されたユーザーロードライブラリに生成されます。モジュール名は、LNKGBLS または LNKRGBL メンバの GBLNAME パラメータで指定された名前になります。

- 4 サンプルジョブ LNKLNK8 または LNKLNKR8 (リエントラント) をユーザーソースライブラリにコピーし、前の手順で作成した新しいグローバルモジュールおよび必要なすべての出口が適切な ADALNK8 または ADALNKR8 (リエントラント) 基本モジュールとリンクするように修正します。サンプルジョブの修正手順は、ジョブの先頭に記載されています。ジョブの出力先を適切なユーザーロードライブラリに設定するようにしてください。LNKLNK8 と LNKLNKR8 は Adabas 8 ADA_{Vrs}.SRCE ライブラリ内にあります。

このジョブから生成されるモジュールは、ADALNK または ADALNKR のいずれか適切な名前になります。

- 5 Adabas 8 バッチ/TSO リンクルーチンを使用するジョブステップの ADARUN DDCARD 入力を修正します。DDCARD 入力に次の更新を含める必要があります。
 - 非リエントラントリンクルーチンの場合、ADARUN PROG=USER パラメータを指定します。ジョブステップでリエントラントリンクルーチンを使用する場合は、ADARUN PROG=RENTUSER を指定します。PROG パラメータの詳細については、「PROGRAM : 実行するプログラム」を参照してください。
- 6 ジョブステップから適切なロードライブラリが利用できることを確認します。これらのロードライブラリは、STEPLIB、TASKLIB、JOB LIB、リエントラントモジュールの場合は、LPA または LINKLIB になります。

Adabas 8 ADALNK または ADALNKR モジュールのデフォルト値の ZAP

Adabas 8 ADALNK および ADALNKR モジュールで使用されるさまざまなリンクルーチンパラメータのデフォルト値を変更できます。

Adabas 8 バッチ/TSO リンクルーチン、ADALNK、および ADALNKR の一部のデフォルト値は、ADALNK または ADALNKR モジュールの ZAP で変更されます。例えば、データベース ID や Adabas SVC 番号のデフォルト値などがあります。その他すべてのデフォルト値は、リンクグローバルテーブルを使用して設定する必要があります。詳細については、「[Adabas 8 ADALNK または ADALNKR モジュールのデフォルト値の変更](#)」を参照してください。

次の IMASPZAP コントロールステートメントを使用して、ADALNK または ADALNKR（適切な方）のデフォルト値を変更します。

```

NAME ADALNK ADALNK8
VER 0080 0001          Default DBID
REP 0080 #####       Site-specific DBID
VER 0084 0AF9         Default Adabas SVC number
REP 0084 0A##        Site-specific Adabas SVC number
*
NAME ADALNKR ADALNKR8
VER 0080 0001          Default DBID
REP 0080 #####       Site-specific DBID
VER 0084 0AF9         Default Adabas SVC number
REP 0084 0A##        Site-specific Adabas SVC number

```

Adabas データベース ID による Adabas SVC ルーティングの確立

z/OS 環境で Adabas リンクルーチンを使用するアプリケーションは、コールで使用されるデータベース ID に基づき、特定の Adabas SVC を通じてデータベースコールをルーティングすることができます。SVC のルーティングは、ユーザーが設定する DBID/SVC ルーティングテーブルによって管理されます。テーブルで最大 1000 個のデータベース ID を指定し、z/OS システムにインストールされている任意の数の有効な SVC 番号と関連付けることができます。DBID/SVC ルーティングテーブルは、MDBSVC マクロを使用して作成します。

データベース ID の重複は DBID/SVC ルーティングテーブルでは許可されていません。これは、複数のデータベース ID がリストされていると、あるデータベース ID にどの SVC を使用すべきかをリンクルーチンが決定する際の信頼性がなくなるからです。テーブルのアセンブル中に重複するデータベース ID が見つかった場合、アセンブラ MNOTE のフラグが設定され、アセンブルの試行に対してリターンコード 16 が返されます。



Notes:

1. クライアントベースの機能を動作させるには、すべてのセッションにおいて複数のルーターを通さずに、単一のルーターを通して経路を決定する必要があるため、Adabas Transaction Manager などのクライアントベースの Adabas アドオンには、この機能との互換性がありません。
2. ADASVCTB とリンクした ADALNK は、アプリケーションプログラムのみにより使用される必要があります。また、Adabas ニュークリアスまたは Entire Net-Work に使用させることはできません。



Caution: この機能の使用には注意が必要です。トランザクションの完全性は保証されません。あるアプリケーションが、複数の Adabas SVC にルーティングされる複数のデータベースにコールを行う場合、1つのデータベースのトランザクションに影響する可能性がある ET、BT、OP、CL、RC、またはその他の Adabas コマンドの発行が可能です。そ

れ以前にアクセスされた、異なる Adabas SVC 上で実行している他のデータベースに対しては発行できません。したがって、複数の Adabas SVC が使用されている複数のデータベース間で、トランザクションの整合性を確実にするために必要なロジックがすべてアプリケーションプログラムに含まれている必要があります。

このsectionでは、次のトピックについて説明します。

- [Adabas DBID/SVC ルーティング機能のインストール](#)
- [全般的な動作](#)
- [MDBSVC マクロの使用](#)

Adabas DBID/SVC ルーティング機能のインストール

Adabas DBID/SVC ルーティング機能の一般的なインストール手順は次のとおりです。

1. MDBSVC マクロステートメントを使用して、ライブラリメンバに DBID/SVC ルーティングテーブルを定義します。DBID/SVC ルーティングテーブルと MDBSVC マクロの詳細については、「[MDBSVC マクロの使用](#)」を参照してください。
2. DBID/SVC ルーティングメンバをアセンブルおよびリンクエディットし、SVC ルーティング機能を使用する操作環境で利用できるロードモジュールを作成します。
3. 操作環境のリンクグローバルテーブルを修正し、LGBLSET キーワード DYNDBSVC=YES および DBSVCTN=name を指定します。name はリンクルーチンによって使用される DBID/SVC ルーティングテーブルロードモジュールの名前です。操作環境が必要な場合は、更新したリンクグローバルテーブルをアセンブルおよびリンクエディットします。リンクグローバルテーブルと LGBLSET マクロの詳細については、「[バージョン8でのソースメンバデフォルトの修正 \(LGBLSET マクロ\)](#)」を参照してください。テーブル更新後のリンクグローバルテーブルのアセンブルとリンクエディットについては、このsectionの各 z/OS TP モニタリング環境に関する指示を参照してください。
4. 準備した DBID/SVC ルーティングテーブルを、アプリケーションプログラムのジョブステップからアクセスできるロードライブラリで利用できるようにします。これにより、実行時にリンクルーチンがロードできます。

このsectionでは、次のトピックについて説明します。

- [バッチ、TSO、および IMS 配下での DBID/SVC ルーティング機能のインストール](#)

■ CICS 配下での DBID/SVC ルーティング機能のインストール

バッチ、TSO、および IMS 配下での DBID/SVC ルーティング機能のインストール

バッチ、TSO、および IMS 配下での Adabas SVC ルーティング機能のインストール手順はどれも同じです。

▶手順 4.11. バッチ、TSO、または IMS 配下で Adabas DBID/SVC ルーティング機能をインストールするには、次の手順に従います。

- 1 ライブラリメンバの一連の MDBCSVC マクロをコーディングして、DBID/SVC ルーティングテーブルを定義または修正します。サンプルの ADASVCTB メンバは、このメンバの準備用のテンプレートとして ADAvrs.MVSSRCE ライブラリで提供されています。MDBSVC マクロの使用についての詳細は、「[MDBSVC マクロの使用](#)」を参照してください。
- 2 DBID/SVC ルーティングテーブルメンバをアセンブルおよびリンクエディットし、アプリケーション実行ジョブステップから利用できるロードモジュールを作成します。同一の Adabas SVC にアクセスする複数のコマンドの処理オーバーヘッドを削減するために、リンクルーチンサブプログラム LNKDSL は DBID/SVC モジュールに Adabas SVC IDT ヘッダーのアドレスを格納する必要があるため、ロードモジュールは再利用不可および非リエントラントとしてリンクされる必要があります。
- 3 実行環境のリンクグローバルテーブルを定義または修正します。Adabas SVC ルーティング機能をサポートするには、次の LGBLSET キーワードが必要です。

LGBLSET キーワード設定	説明
DYNDBSVC=YES	このキーワードおよび設定は、Adabas SVC ルーティングがこのジョブステップでアクティブであることを示します。
DBSVCTN= <i>name</i>	このキーワードは、このジョブステップの DBID/SVC テーブル名を指定します。この名前は、リンクルーチンの実行時に適切なテーブルがロードされるように、作成されたロードモジュールの名前と一致する必要があります。

- 4 TP モニタ別の指示に従って、更新したリンクグローバルテーブルのアセンブルおよびリンクエディットします。バッチ/TSO の場合は、「[Adabas 8 配下での Batch/TSO を使用する場合の Adabas のインストール](#)」を参照してください。IMS の場合は、「[Adabas 8 配下での IMS/TM を使用する場合の Adabas のインストール](#)」を参照してください。

CICS 配下での DBID/SVC ルーティング機能のインストール

▶手順 4.12. CICS 配下で Adabas DBID/SVC ルーティング機能をインストールするには、次の手順に従います。

- 1 ライブラリメンバの一連の MDBCSVC マクロをコーディングして、DBID/SVC ルーティングテーブルを定義または修正します。サンプルの ADASVCTB メンバは、このメンバの準備用のテンプレートとして ADA_{vrs}.MVSSRCE ライブラリで提供されています。MDBSVC マクロの使用についての詳細は、「[MDBSVC マクロの使用](#)」を参照してください。
- 2 DBID/SVCルーティングテーブルメンバをアセンブルおよびリンクエディットし、テーブルをロードモジュールとして作成し、CICSDFHRPL 連結の一部となるライブラリに配置します。同一の Adabas SVC にアクセスする複数のコマンドの処理オーバーヘッドを削減するために、リンクルーチンサブプログラム LNKDSL は DBID/SVC モジュールに Adabas SVC IDT ヘッダーのアドレスを格納する必要があるため、ロードモジュールは再利用不可および非リエントラントとしてリンクされる必要があります。
- 3 RDO または DFHCSDUP ユーティリティを使用して、ロードモジュールをプログラムとして CICS に定義します。サンプル DFHCSDUP 定義ステートメントについては、ACI_{vrs}.MVSSRCE ライブラリのメンバ DEFADA8 を参照してください。プログラム属性は、Reload(No)、Resident(Yes)、Dataloc(Any)、Exekey(CICS) にする必要があります。
- 4 実行環境のリンクグローバルテーブルを定義または修正します。Adabas SVC ルーティング機能をサポートするには、次の LGBLSET キーワードが必要です。

LGBLSET キーワード設定	説明
DYNDBSVC=YES	このキーワードおよび設定は、Adabas SVC ルーティングがこのジョブステップでアクティブであることを示します。
DBSVCTN= <i>name</i>	このキーワードは、このジョブステップの DBID/SVC テーブル名を指定します。この名前は、リンクルーチンの実行時に適切なテーブルがロードされるように、作成されたロードモジュールの名前と一致する必要があります。

- 5 「[Adabas 8 配下での CICS を使用する場合の Adabas のインストール](#)」の説明に従って、更新したグローバルテーブルをアセンブルおよびリンクエディットします。

全般的な動作

このsectionで前述したAdabas SVC ルーティング機能をインストールすると、次のようにロードされます。

- バッチ、TSO、または IMS 環境では、リンクルーチンの初期化時に DBID/SVC ルーティングテーブルがロードされます（リンクグローバルテーブルの LGBLSET DYNDBSVC パラメータが YES に設定されている場合）。ルーティングテーブルのアドレスは、その後のコールで使用できるように、リンクルーチンのワークエリアに保持されます。

- CICS 環境では、通常 PLTPI 処理中に実行される Adabas 8 初期化モジュールが DBID/SVC ルーティングテーブルをロードして検証します（CICS リージョンのリンクグローバルテーブルで LGBLSET DYNDBSVC パラメータが YES に設定されている場合）。ルーティングテーブルのアドレスは、Adabas 8 タスク関連ユーザー出口（TRUE）モジュール ADACICT に関連付けられたグローバルワークエリアに保持されます。また、このアドレスは、Adabas コマンドレベルモジュール ADACICS/ADADCI による TRUE へのアプリケーションコールで利用できます。

アプリケーションコールが実行されると、DBID/SVC ルーティングテーブルは、各操作環境の適切なリンクルーチンにリンクされた LNKDSL サブルーチンによって検索されます。Adabas 以前のコールのユーザー出口がその後の処理用にコマンドのデータベース ID を変更している可能性があるため、LNKDSL は LUEXIT1（リンクルーチンユーザー出口 1）の起動後に呼び出されます。Adabas Review、Adabas Fastpath、Adabas Transaction Manager などのモニタリング製品が、DBID/SVC ルーティングテーブルにある適切な Adabas SVC に基づいて処理を行えるように、LNKDSL へのコールは何らかのモニタリングまたは Adabas Fastpath 出口が呼び出される前に実行されます。

特定のコールに関連付けられているデータベース Id が DBID/SVC ルーティングテーブルで見つからなかった場合、MDBSVC マクロの TYPE=INIT パラメータで指定された、Adabas SVC のデフォルト値が使用されます。特定された SVC が Adabas SVC でない場合、または z/OS システムにインストールされていない場合、Adabas レスポンスコード 213、サブコード 16 または 20 がアプリケーションに返されます。コール元のデータベースが SVC 番号に対してアクティブでない場合、Adabas レスポンスコード 148 がアプリケーションに返されます。

データベース ID の重複は DBID/SVC ルーティングテーブルでは許可されていません。これは、複数のデータベース ID がリストされていると、あるデータベース ID にどの SVC を使用すべきかをリンクルーチンが決定する際の信頼性がなくなるからです。テーブルのアセンブル中に重複するデータベース ID が見つかった場合、アセンブラ MNOTE のフラグが設定され、アセンブルの試行に対してリターンコード 16 が返されます。

MDBSVC マクロの使用

MDBSVC マクロを使用して、Adabas DBID/SVC ルーティングテーブルのさまざまな定義を行います。複数の MDBSVC マクロを TYPE=INIT、TYPE=GEN、TYPE=FINAL キーワードを使用してまとめてコーディングすることで、1つのソースモジュールまたはメンバを構成します。このソースモジュールまたはメンバをアセンブルおよびリンクエディットし、DBID/SVC ルーティングテーブルロードモジュールを作成します。ADA_{vrs}.MVSSRCE のサンプルメンバ ADASVCTB をテンプレートとして使用して、各環境に合わせた DBID/SVC ルーティングテーブルソースモジュールを作成できます。次に CSECT 名 TESTDBT を使用した DBID/SVC ルーティングテーブルソースメンバの例を示します。テーブルをアセンブルすると、ロードモジュール名は TESTDBT になります。

```

TESTDBT CSECT
MDBSVC TYPE=INIT,SVC=249,DBID=001
MDBSVC TYPE=GEN,SVC=237,DBID=(2,10,21,33,175,1149), X
        DBID2=(100,101,102,13500)
MDBSVC TYPE=GEN,SVC=231,DBID=(226,899)
MDBSVC TYPE=GEN,SVC=206,DBID=(15,16,69,99,500,12144)
MDBSVC TYPE=GEN,SVC=248,DBID=(14,54,111,177,1213,5775)
MDBSVC TYPE=GEN,SVC=249,DBID=(17,19,25,35,42,44,61,76)
MDBSVC TYPE=FINAL
END
    
```

MDBSVC マクロステートメントのキーワード値をコーディングする場合、行の継続、リストの識別、およびキーワード値の指定に関するアセンブラの規則に従わないと、アセンブリエラーが発生するため注意してください。キーワードのオブジェクトとしてコーディングされたリストを含むキーワードと値は、コンマで区切る必要があります。MDBSVC マクロでは位置パラメータは使用されません。

MDBSVC マクロには、次の表に示す 4 種類のステートメントを含めることができます。

MDBSVC ステートメントタイプ	説明	使用可能な数
TYPE=INIT	DBID/SVC ルーティングテーブルソースメンバに含めることのできる MDBSVC TYPE=INIT ステートメントは 1 つだけです。また、メンバ内の最初の MDBSVC ステートメントにする必要があります。このステートメントは、DBID/SVC ルーティングテーブルの先頭を識別します。MDBSVC TYPE=INIT ステートメントでは、コールに使用されるデフォルトのデータベース ID と Adabas SVC 番号を指定することもできます。	1
TYPE=GEN	DBID/SVC ルーティングテーブルソースメンバには、任意の数の MDBSVC TYPE=GEN ステートメントを含めることができます。このステートメントによって、特定の有効な Adabas SVC 番号に関連付けられた Adabas データベース ID を指定します。	必要に応じた任意の数
TYPE=FINAL	DBID/SVC ルーティングテーブルソースメンバに含めることのできる MDBSVC TYPE=FINAL ステートメントは 1 つだけです。また、メンバ内の最後の MDBSVC ステートメント（アセンブラ END ステートメントの直前）にする必要があります。このステートメントは、DBID/SVC ルーティングテーブルの末尾を識別します。	1
TYPE=DSECT	このステートメントタイプは、Software AG が内部で使用するため、予約されています。このステートメントタイプを使用しないでください。	0

MDBSVC TYPE=INIT ステートメントの前に、名前付きの CSECT、AMODE および RMODE ステートメントを配置することができます。CSECT、AMODE、または RMODE ステートメントを含める場合、使用する名前が DBID/SVC ルーティングテーブルの名前と一致している必要があります。つまり、MDBSVC TYPE=INIT ステートメントの TABNAME パラメータでコーディングされた名前と、リンクグローバルテーブルを作成するときに、LGBLSET マクロの DBSVCTN キーワードで指定された名前です。

このsectionでは、次のトピックについて説明します。

- [MDBSVC TYPE=INIT の構文](#)
- [MDBSVC TYPE=GEN の構文](#)
- [MDBSVC TYPE=FINAL の構文](#)
- [MDBSVC パラメータ](#)

MDBSVC TYPE=INIT の構文

MDBSVC TYPE=INIT ステートメントの構文は次のとおりです。

```
MDBSVC TYPE=INIT [ ,SVC=svcno ] [ ,DBID=dbid ] [ ,TABNAME={name | ADBSVCT} ]
[ ,OPSYS={ZOS | VSE} ]
```

MDBSVC TYPE=INIT ステートメントでコーディングできるパラメータについては、「[MDBSVC パラメータ](#)」を参照してください。

MDBSVC TYPE=GEN の構文

MDBSVC TYPE=GEN ステートメントの構文は次のとおりです。

```
MDBSVC TYPE=GEN [ ,SVC=svcno ] [ ,DBID=id[ , id]... ] [ ,DBID2=id[ , id]... ]
```

MDBSVC TYPE=GEN ステートメントでコーディングできるパラメータについては、「[MDBSVC パラメータ](#)」を参照してください。

MDBSVC TYPE=FINAL の構文

MDBSVC TYPE=FINAL ステートメントの構文は次のとおりです。

```
MDBSVC TYPE=FINAL
```

MDBSVC TYPE=FINAL ステートメントではパラメータを使用することはできません。

MDBSVC パラメータ

MDBSVC ステートメントで指定できるパラメータには次のようなものがあります。

DBID

DBID パラメータは、MDBSVC TYPE=INIT ステートメントと MDBSVC TYPE=GEN ステートメントの両方で使用できます。

- MDBSVC TYPE=INIT ステートメントで指定する場合、SVC パラメータで指定された SVC に関連付けられたデフォルトデータベース ID を指定します。この場合、TYPE=INIT ステートメントの DBID パラメータにリストできるデータベース ID は 1 つだけです。
- MDBSVC TYPE=GEN ステートメントで指定する場合、SVC パラメータで指定されたパラメータに関連付けられたデータベース ID を指定します。複数のデータベース ID をリストする場合、かっこで囲み、コンマで区切る必要があります。

DBID パラメータでリストするデータベース ID は、インストールした Adabas データベースの ID に対応する数値にする必要があります。z/OS 環境では、データベース ID は 1 から 65535 の範囲である必要があります。同一のデータベース ID を複数の MDBSVC ステートメントに指定することはできません。データベース ID は、DBID/SVC ルーティングテーブルのすべての DBID および DBID2 ステートメントにおいてユニークでなければなりません。重複した値には MNOTE のフラグが設定されます。これにより DBID/SVC ルーティングテーブルのアセンブルが停止してリターンコード 16 が返されます。

MDBSVC ステートメントに対する DBID パラメータの設定例を次に示します。SVC 237 に関連付けられているデータベース ID をリストしている MDBSVC ステートメントが 2 つあることに注意してください。これにより、同一の SVC 番号に対して、より多くのデータベース ID をコーディングすることができます。この方法で記述した内容と同じものを DBID2 パラメータを使用して記述した例と比べてみてください。どちらのコーディングも、結果は同じになります。

```
MDBSVC TYPE=INIT,SVC=249,DBID=1
MDBSVC TYPE=GEN,SVC=237,DBID=( 2,4,10,16,21,33)
MDBSVC TYPE=GEN,SVC=237,DBID=(175,1149,1221)
MDBSVC TYPE=GEN,SVC=242,DBID=( 3,18)
MDBSVC TYPE=FINAL
END
```

DBID2

DBID2 パラメータは、MDBSVC TYPE=GEN ステートメントでのみ使用できます。SVC パラメータで指定した Adabas SVC に関連付けるデータベース ID のリストを追加することができます。DBID2 パラメータはオプションですが、指定する場合は、DBID パラメータの直後に指定する必要があります。

DBID2 パラメータでリストするデータベース ID は、インストールした Adabas データベースの ID に対応する数値にする必要があります。z/OS 環境では、データベース ID は 1 から 65535 の範囲である必要があります。同一のデータベース ID を複数の MDBSVC ステートメントに指定することはできません。データベース ID は、DBID/SVC ルーティングテーブルのすべての DBID および DBID2 ステートメントにおいてユニークでなければなりません。重複した値には MNOTE のフラグが設定されます。これにより DBID/SVC ルーティングテーブルのアセンブルが停止してリターンコード 16 が返されます。

DBID2 パラメータを含む MDBSVC ステートメントの例を次に示します。この方法で記述した内容と同じものを DBID パラメータを使用して記述した例と比べてみてください。どちらのコーディングも、結果は同じになります。

```

MDBSVC TYPE=INIT,SVC=249,DBID=1
MDBSVC TYPE=GEN,SVC=237,DBID=(2,4,10,16,21,33),
DBID2=(175,1149,1221)
MDBSVC TYPE=GEN,SVC=242,DBID=(3,18)
MDBSVC TYPE=FINAL
END

```

OPSYS

OPSYS パラメータは、MDBSVC TYPE=INIT ステートメントでのみ使用できるオプションのパラメータです。このパラメータは、DBID/SVC ルーティングテーブルがアセンブリされるオペレーティングシステムを識別します。OPSYS パラメータに有効な値としては "ZOS" と "VSE" があり、"ZOS" がデフォルトです。

PREFIX

PREFIX パラメータは、MDBSVC TYPE=DSECT ステートメントでのみ使用できます。このパラメータは Software AG が内部で使用するために予約されています。このパラメータは使用しないでください。

SVC

SVC パラメータは、MDBSVC TYPE=INIT ステートメントと MDBSVC TYPE=GEN ステートメントの両方で使用できます。

- MDBSVC TYPE=INIT ステートメントで指定する場合、デフォルトの Adabas SVC 番号を指定します。この番号は、コール元のアプリケーションから指定されたデータベース ID が、DBID/SVC ルーティングテーブルに見つからなかったときに使用されます。
- MDBSVC TYPE=GEN ステートメントで指定する場合、DBID および DBID2 パラメータで指定される Adabas データベースに関連付ける Adabas SVC 番号を指定します。

SVC パラメータで指定する SVC 番号は、数値であり、かつインストールした Adabas SVC の SVC 番号に対応している必要があります。z/OS 環境では、SVC 番号の範囲は 200~255 です。複数の MDBSVC ステートメントで SVC 値を重複して使用できます。これにより、データベース ID の長いリストを記述し、それらを同一の Adabas SVC に関連付けることができます。

次の例で、2つの MDBSVC ステートメントで SVC 249 が指定されている点に注意してください。この SVC は、リンクルーチンのデフォルト SVC であるとともに、データベース 1、2、18 でも使用されます。また、SVC 237 に対しても 2つの MDBSVC ステートメントがあります。2つのステートメントで、SVC 237 に関連付ける 9つのデータベースがリストされています (2、4、10、16、21、33、175、1149、1221)。

```

MDBSVC TYPE=INIT,SVC=249,DBID=1
MDBSVC TYPE=GEN,SVC=237,DBID=(2,4,10,16,21,33)
MDBSVC TYPE=GEN,SVC=237,DBID=(175,1149,1221)
MDBSVC TYPE=GEN,SVC=249,DBID=(3,18)
MDBSVC TYPE=FINAL
END

```

TABNAME

TABNAME パラメータは、MDBSVC TYPE=INIT ステートメントでのみ使用できるオプションのパラメータです。このパラメータには、ソースメンバに独立した（かつ以前にコーディングされた）CSECT ステートメントが含まれていない場合に、DBID/SVC ルーティングテーブルの名前を指定します。この場合、TABNAME パラメータで指定した名前が、名前付き CSECT ステートメントおよび、名前付き AMODE、RMODE 指示の生成に使用されます。

指定する DBID/SVC ルーティングテーブル名は、1～8 文字の英数字にする必要があります。次の例では、TESTDBT という名前の DBID/SVC ルーティングテーブルがコーディングされています。

```
MDBSVC TYPE=INIT,SVC=249,DBID=1,TABNAME=TESTDBT
MDBSVC TYPE=GEN,SVC=237,DBID=(2,4,10,16,21,33)
MDBSVC TYPE=GEN,SVC=237,DBID=(175,1149,1221)
MDBSVC TYPE=GEN,SVC=249,DBID=(3,18)
MDBSVC TYPE=FINAL
END
```

バージョン 8 のソースメンバのデフォルト (LGBLSET マクロ) の変更点

Adabas 8 LGBLSET マクロは、Adabas リンクルーチンのデフォルトのインストール値の設定に使用します。このマクロは、Adabas 8 リンクルーチンとリンクエディットされているか、リンクルーチンを実行しているジョブステップのリンクルーチンに提供されるオブジェクトモジュールを準備するために使用します。Adabas ライブラリには、各環境の各種 TP モニタをサポートするために提供されたサンプルメンバが含まれています。これらのサンプルメンバは適切なライブラリにコピーされ、指定した環境で実行するリンクルーチンに必要なカスタマイズを行うために変更されます。

LGBLSET パラメータオプションとそのデフォルト値（下線付き）については、この section の後半を参照してください。

- ADL : Adabas Bridge for DL/I のサポート
- AVB : Adabas Bridge for VSAM のサポート
- CITSNM : Adabas CICS TS のキュー名
- COR : SYSCOR 出口のサポート
- DBSVCTN : DBID/SVC ルーティングテーブル
- DYNDBSVC : DBID/SVC ルーティングテーブル
- ENTPT : Adabas CICS コマンドレベルリンクルーチンの名前
- GBLNAME : リンクグローバルモジュール名
- GEN : CSECT または DSECT の生成
- IDTNAME : BS2000 IDT 共通メモリ名
- IDTUGRP : BS2000 メモリプールのユーザー連結

- LOGID：デフォルトの論理データベース ID
- LUEXIT1A：LUEXIT1 の長さ
- LUEXIT2A：LUEXIT2 の長さ
- LUINFO：Adabas LUEXIT1 および LUEXIT2 に渡されたユーザーデータの長さ
- LUSAVE：Adabas LUEXIT1 および LUEXIT2 のユーザーセーブエリアのサイズ
- LX1NAME：ユーザー出口 1 のモジュール名
- LX2NAME：ユーザー出口 2 のモジュール名
- MRO：マルチプルリージョンオプション
- NETOPT：ユーザー ID の作成に使用されたメソッド
- NTGPID：Natural グループ ID
- NUBS：CICS リンクルーチンに作成されるユーザーブロック数
- OPSYS：オペレーティングシステム
- PARMTYP：Adabas パラメータリストのエリア
- PRE：DSECT データ接頭辞
- PURGE：トランザクションの削除
- RENT：リエントラントモジュールフラグ
- RETRYX：再試行コマンド出口フラグ
- REVIEW：Adabas Review のサポート
- RMI：リソースマネージャインターフェイス
- RTXNAME：コマンド再試行出口名
- SAF：Adabas Security インターフェイスフラグ
- SAP：SAP アプリケーションのサポート
- SAPSTR：SAP ID 文字列
- SVCNO：Adabas SVC 番号
- TPMON：オペレーティング環境
- TRUENM：CICS TRUE 名
- UBPLC：ユーザーブロックプールの割り当て
- UES：ユニバーサルエンコーディングサポート
- USERX1：ユーザー出口 1 フラグ
- USERX2：ユーザー出口 2 フラグ
- XWAIT：CICS の XWAIT 設定

ADL：Adabas Bridge for DL/I のサポート

パラメータ	説明	構文
ADL	<p>このコマンドレベルリンクルーチンにより Software AG の Adabas Bridge for DL/I の Consistency Interface がサポートされるかどうかを示します。</p> <ul style="list-style-type: none"> ■ ADL=YES の場合、Adabas Bridge for DL/I の Consistency Interface はサポートされます。 ■ ADL=NO の場合、Adabas Bridge for DL/I の Consistency Interface はサポートされません。 	ADL={NO YES}

AVB : Adabas Bridge for VSAM のサポート

パラメータ	説明	構文
AVB	このコマンドレベルリンクルーチンにより Software AG の Adabas Bridge for VSAM がサポートされるかどうかを示します。 <ul style="list-style-type: none"> ■ AVB=YES の場合、Adabas Bridge for VSAM はサポートされます。 ■ AVB=NO の場合、Adabas Bridge for VSAM はサポートされません。 	AVB={NO YES}

CITSNM : Adabas CICS TS のキュー名

パラメータ	説明	構文
CITSNM	Adabas の CICS TS キュー名を表す 16 バイトの文字列を指定します。デフォルトは "ADACICS" です。	CITSNM={ <u>ADACICS</u> <i>qname</i> }

COR : SYSCOR 出口のサポート

パラメータ	説明	構文
COR	Adabas System Coordinator (SYSCOR) 、 Adabas Transaction Manager、および Adabas Fastpath の出口がインストールされアクティブになっているかどうかを示します。 <ul style="list-style-type: none"> ■ COR=YES の場合、出口はインストールされておりアクティブになっています。 ■ COR=NO の場合、出口はインストールされておらず、アクティブになっていません。 	COR={NO YES}

DBSVCTN : DBID/SVC ルーティングテーブル

パラメータ	説明	構文
DBSVCTN	実行中にリンクルーチンに使用される DBID/SVC ルーティングテーブルの名前を指定します (存在する場合)。 ルーティングテーブル名は z/OS 標準ロードモジュールの名前に従う必要があります。このテーブル名は、バッチ、TSO、または IMS 操作の実行中は z/OS LOAD マクロ/SVC に、CICS 操作の実行中は EXEC CICS LOAD PROGRAM コマンドにより使用されます。 リストされたロードモジュールが見つからない場合、またはも無効なヘッダー情報を含んでいる場合は、バッチ、TSO、または IMS 環境でユーザーアベンド U657 が発行されます。	DBSVCTN={ <i>name</i> <u>ADASVCTB</u> }

パラメータ	説明	構文
	<p>ロードモジュールが CICS に定義されていない場合、または CICS DFHRPL 連結に見つからない場合は、Adabas CICS リンクルーチン環境は初期化されません。</p> <p>注意: DYNDBSVC パラメータが NO に設定されている場合、このパラメータ設定は無視されます。</p> <p>z/OS 環境のデータベース ID ごとの SVC ルーチンの詳細については「<i>Establishing Adabas SVC Routing by Adabas Database ID</i>」を参照してください。</p>	

DYNDBSVC : DBID/SVC ルーティングテーブル

パラメータ	説明	構文
DYNDBSVC	<p>リンクルーチンに対して、データベース ID による Adabas SVC ルーティングを有効にするかどうかを示します。DYNDBSVC=YES を指定すると、データベース ID による Adabas SVC ルーティングが有効になりますが、DYNDBSVC により無効になります。デフォルトは NO です。</p> <p>z/OS 環境のデータベース ID ごとの SVC ルーチンの詳細については「<i>Establishing Adabas SVC Routing by Adabas Database ID</i>」を参照してください。</p>	DYNDBSVC={ YES NO }

ENTPT : Adabas CICS コマンドレベルリンクルーチンの名前

パラメータ	説明	構文
ENTPT	<p>Adabas CICS コマンドレベルリンクルーチン名。この名前を EXEC CICS LINK コマンドに使用して CICS アプリケーションプログラムから Adabas サービスを呼び出します。</p> <p>インストール手順の注 1 と注 2 も参照してください。</p>	ENTPT={ <u>ADACICS</u> name }

GBLNAME : リンクグローバルモジュール名

パラメータ	説明	構文
GBLNAME	リンクグローバルモジュール名。	GBLNAME={ <u>LNKGBLS</u> name }

GEN : CSECT または DSECT の生成

パラメータ	説明	構文
GEN	CSECT または DSECT が生成されているかどうかを示します。	GEN={ <u>CSECT</u> DSECT}

IDTNAME : BS2000 IDT 共通メモリ名

パラメータ	説明	構文
IDTNAME	BS2000 IDT の共通メモリプール名。	IDTNAME= <i>name</i>

IDTUGRP : BS2000 メモリプールのユーザー連結

パラメータ	説明	構文
IDTUGRP	メモリプールがユーザー連結かどうかを示します (BS2000)。	IDTUGRP={ <u>NO</u> YES}

LOGID : デフォルトの論理データベース ID

パラメータ	説明	構文
LOGID	デフォルトのターゲットデータベース ID の値。有効な ID 番号は 1~65535 です。デフォルトは "1" です。	LOGID={ <i>nnn</i> <u>1</u> }

LUEXIT1A : LUEXIT1 の長さ

パラメータ	説明	構文
LUEXIT1A	リンクユーザー出口 1 のワークエリアの長さ。有効な値は 0~32,767 の数値です。デフォルトは "0" です。	LUEXIT1A={ <i>nnn</i> <u>0</u> }

LUEXIT2A : LUEXIT2 の長さ

パラメータ	説明	構文
LUEXIT2A	リンクユーザー出口 2 のワークエリアの長さ。有効な値は 0 (ゼロ) ~32,767 の数値です。デフォルトは "0" です。	LUEXIT2A={ <i>nnn</i> <u>0</u> }

LUINFO : Adabas LUEXIT1 および LUEXIT2 に渡されたユーザーデータの長さ

パラメータ	説明	構文
LUINFO	ターゲットユーザー出口4に渡されるユーザーデータの長さ。有効な値は 0~32,767 の数値です。 LUINFO を指定しない場合、0 (デフォルト) になり、ユーザーセーブエリアは渡されません。	LUINFO={0 length}

LUSAVE : Adabas LUEXIT1 および LUEXIT2 のユーザーセーブエリアのサイズ

パラメータ	説明	構文
LUSAVE	Adabas ユーザー出口 LUEXIT1 と LUEXIT2 に使用されるユーザーセーブエリアのサイズ。有効な値の範囲は 0 (ゼロ) ~256 です。デフォルトは "72" です。 LUSAVE を指定しない場合、0 (デフォルト) になり、ユーザーデータは渡されません。	LUSAVE={72 size}

LX1NAME : ユーザー出口 1 のモジュール名

パラメータ	説明	構文
LX1NAME	リンクユーザー出口 1 のモジュール名	LX1NAME={LUEXIT1 name}

LX2NAME : ユーザー出口 2 のモジュール名

パラメータ	説明	構文
LX2NAME	リンクユーザー出口 2 のモジュール名	LX2NAME={LUEXIT2 name}

MRO : マルチプルリージョンオプション

パラメータ	説明	構文
MRO	CICS マルチプルリージョンオプション (MRO) のサポートが必要かどうかを示します。 CICS MRO とともに CICS コマンドレベルリンクを実行する場合、MRO=YES をセットします。それ以外の場合は、デフォルト値 MRO=NO を使用します。 MRO=YES の場合、NETOPT に NETOPT=NO (デフォルト) をセットし、マルチプルアプリケーションリージョンからユニークでない LU 名を回避しなければなりません。	MRO={NO YES}

パラメータ	説明	構文
	NETOPT=YES および MRO=YES のとき、アセンブラ MNOTE およびリターンコード 16 がアセンブリステップから出力されます。	

NETOPT : ユーザー ID の作成に使用されたメソッド

パラメータ	説明	構文
NETOPT	NETOPT=YES が指定されている場合は、8 バイトのユーザー ID が VTAM LU 名から構成されます。NETOPT=NO が指定されている場合、ユーザー ID は定数 CICS に端末タスク用の 4 バイトの CICS 端末 ID (TCTTETI) で作られます。非端末タスクの場合、ユーザー ID は定数 CIC と CICS タスク番号で構成されます。 CICS マルチプルリージョンオプション (MRO) 付きで実行する場合、このオプションに対してデフォルト値を使用する必要があります。 NETOPT=YES および MRO=YES のとき、アセンブラ MNOTE およびリターンコード 16 がアセンブリステップから出力されます。	NETOPT={NO YES}

NTGPID : Natural グループ ID

パラメータ	説明	構文
NTGPID	Natural バージョン 2.2.8 以上の CICSplex 環境でユニークな Adabas ユーザー ID を生成する必要がある場合には、4 バイトの Natural グループ ID を指定します。値は、指定値でアセンブルされた Adabas コマンドレベルリンクルーチンをコールするすべてのユーザーに関連します。 デフォルト値はありません。値が指定されない場合、従来どおり Adabas 内部ユーザー ID が作成されます。 4 バイトの英数字の値を指定することもできますが、CICSplex または z/OS イメージで実行している Adabas コマンドレベルリンクルーチンごとに固有である必要があります。複数の NTGPID が必要なとき (テストと本番の Natural 2.2.8 など)、TRUE に関連する複数の Adabas コマンドレベルリンクルーチンを生成する必要があります。 CICS マルチプルリージョンオプション (MRO) 付きで実行する場合、NTGPID を使用して、マルチプルアプリケーションリージョンが Adabas をコールするとき、Adabas SVC に使用される Adabas コミュニケーション ID 用の 4 バイトリテラルを指定します。	NTGPID=4-byte-value

NUBS : CICS リンクルーチンに作成されるユーザーブロック数

パラメータ	説明	構文
NUBS	<p>CICS リンクルーチンでユーザーブロックプールに作成されるユーザーブロック (UB) 数。ブロック数は同時に発生することが予想される最大 Adabas 要求数を処理可能な大きさにする必要があります。</p> <p>注意: Adabas 6.2 以上のコマンドレベルリンクルーチンは、16 MB 境界より上にユーザーブロック (UB プール) 用のストレージを取得します。</p>	<pre>NUBS={100 blocks}</pre>

OPSYS : オペレーティングシステム

パラメータ	説明	構文
OPSYS	使用しているオペレーティングシステム。	<pre>OPSYS={ZQS VSE CMS BS2}</pre>

PARMTYP : Adabas パラメータリストのエリア

パラメータ	説明	構文
PARMTYP	<p>Adabas パラメータリストを取得するための CICS エリア。"TWA" は、トランザクションワークエリア (TWA) の先頭の 6 フルワードのパラメータリストを取得します。</p> <p>PARMTYP=COM の場合、Adabas パラメータが EXEC CICS LINK コマンドを伴ったコール元プログラムからコールされた CICS COMMAREA に指定されます。ACB コールの COMMAREA リストは最低 32 バイト必要で、ラベル "ADABAS52" で始まります。ACBX コールの COMMAREA リストは最低 24 バイト必要で、ラベル "ADABAS8X" で始まります。また、ACBX コールの COMMAREA リストの最後の ABD を指定するために、VL ビットを設定する必要があり、アドレスの上位ビットを (X'80') にします。</p> <p>PARMTYP=ALL (デフォルト) は COMMAREA と TWA の両方を使用して Adabas パラメータを渡します。この場合、COMMAREA が最初にチェックされます。</p> <p>CICS TWA を Adabas 8 ACBX ダイレクトコールにマッピングすることはお勧めしません。これは、TWA はトランザクションごとにサイズが限定されていることと、CICS 起動時に TWA は使用できないことが理由です。そのため、Adabas 8 CICS リンクルーチンを使用する CICS プログラムでは、データを渡すときだけ COMMAREA を使用することをお勧めします。</p>	<pre>PARMTYP={ALL COM TWA}</pre>

PRE : DSECT データ接頭辞

パラメータ	説明	構文
PRE	DSECT データ接頭辞として、2バイトの文字列が使用されます。デフォルトは "LG" です。	PRE={LG <i>prefix</i> }

PURGE : トランザクションの削除

パラメータ	説明	構文
PURGE	CICS3.2以上でアセンブルするとき、PURGEパラメータが使用されます。PURGE=YES を指定すると、CICS WAIT EXTERNAL はそのパラメータの1つとして PURGEABLE を含むので、DTIMOUT 値を超えたときにPURGEが指定されると、CICSはトランザクションを削除できるようになります。 PURGE=NO (デフォルト) が指定されている場合は、NONPURGEABLE オプションが生成されます。	PURGE={NO YES}

RENT : リエントラントモジュールフラグ

パラメータ	説明	構文
RENT	グローバルモジュールがリエントラントかどうかを示します。	RENT={NO YES}

RETRYX : 再試行コマンド出口フラグ

パラメータ	説明	構文
RETRYX	再試行コマンド出口がアクティブになっているかどうかを示します。	RETRYX={NO YES}

REVIEW : Adabas Review のサポート

パラメータ	説明	構文
REVIEW	Software AG の Review パフォーマンスモニタがインストールされアクティブになっているかどうかを示します。	REVIEW={NO YES}

RMI : リソースマネージャインターフェイス

パラメータ	説明	構文
RMI	<p>CICS リソースマネージャインターフェイスが使用されているかどうかを示すために、RMI パラメータが使用されます。</p> <p>RMI=YES が指定されている場合は、CICS リソースマネージャインターフェイス (RMI) を使用して、Adabas タスク関連ユーザー出口 (TRUE) がリソースマネージャ (RM) として実行されます。</p> <p>RMI=YES は、Adabas Transaction Manager がインストールされ、有効で、CICS 環境で実行しているユーザーに利用可能なときにだけ有効です。Adabas TRUE のインストールと関連する付加的な手順については Adabas Transaction Manager のドキュメントを参照してください。</p>	RMI={NO YES}

RTXNAME : コマンド再試行出口名

パラメータ	説明	構文
RTXNAME	コマンド再試行出口モジュールの名前。	RTXNAME={LUEXRTR name}

SAF : Adabas Security インターフェイスフラグ

パラメータ	説明	構文
SAF	Software AG の Adabas SAF Security サポートが必要かどうかを示します。	SAF={NO YES}

SAP : SAP アプリケーションのサポート

パラメータ	説明	構文
SAP	<p>SAP ユーザー ID の生成がサポートされているかどうかを示します。</p> <p>SAP=YES を指定すると、プログラムは SAP 初期化コールを検出し、初期化コールに提供された定数にフィールド ACBADD2 を合わせて SAP アプリケーションのためのユーザー ID をセットします。</p> <p>詳細については、SAP アプリケーションシステムを使用するお客様のために提供されている補足情報を参照してください。</p>	SAP={NO YES}

SAPSTR : SAP ID 文字列

パラメータ	説明	構文
SAPSTR	使用する 4 バイトの SAP ID 文字列。	SAPSTR={ 'SAP*' <i>string</i> }

SVCNO : Adabas SVC 番号

パラメータ	説明	構文
SVCNO	Adabas SVC 番号の値。 z/OS システムでは、有効な値の範囲は 200~255 です。デフォルトは "249" です。 z/VSE システムでは、有効な値の範囲は 32~128 です。デフォルトは "45" です。	SVCNO= <i>nnn</i>

TPMON : オペレーティング環境

パラメータ	説明	構文
TPMON	TP モニタのオペレーティング環境。有効な値は次のように指定してください。 <ul style="list-style-type: none"> ■ バッチを使用するには、"BAT" を指定します。 ■ CICS を使用するには、"CICS" を指定します。 ■ Com-Plete を使用するには、"COM" を指定します。 ■ IMS を使用するには、"IMS" を指定します。 ■ TSO を使用するには、"TSO" を指定します。 ■ UTM を使用するには、"UTM" を指定します。 <p>注意: OPSYS パラメータで選択したオペレーティングシステムでサポートされている TP モニタオペレーティング環境を指定するようにしてください。</p>	TPMON={ <u>BAT</u> CICS COM IMS }

TRUENM : CICS TRUE 名

パラメータ	説明	構文
TRUENM	Adabas CICS タスク関連ユーザー出口 (TRUE) のモジュール名を指定します。デフォルトは ADACICT です。	TRUENM={ <u>ADACICT</u> name }

UBPLOC : ユーザーブロックプールの割り当て

パラメータ	説明	構文
UBPLOC	<p>ユーザーブロック (UB) プールを CICS の 16 MB 境界より上 (デフォルト) に取得するか、または下に取得するかを指定します。</p> <p>EXEC CICS WAIT WAITCICS または EXEC CICS WAIT EXTERNAL によって使用される ECB を UP プールに組み込みます。</p> <p>UBPLOC=BELOW を設定すると、ECB を 16 MB 境界より上に許可しない CICS バージョン、つまり、CICS/ESA 3.2 以下をサポートします。</p> <p>詳細については、IBM マニュアル『<i>CICS Application Programming Reference</i>』を参照してください。</p>	UBPLOC={ <u>ABOVE</u> BELOW }

UES : ユニバーサルエンコーディングサポート

パラメータ	説明	構文
UES	ユニバーサルエンコーディングサポート (UES) が必要かどうかを示します。	UES={NO <u>YES</u> }

USERX1 : ユーザー出口 1 フラグ

パラメータ	説明	構文
USERX1	ユーザー出口 1 がアクティブになっているかどうかを示します。	USERX1={ <u>NO</u> YES }

USERX2 : ユーザー出口 2 フラグ

パラメータ	説明	構文
USERX2	ユーザー出口 2 がアクティブになっているかどうかを示します。	USERX2={ <u>NO</u> YES }

XWAIT : CICS の XWAIT 設定

パラメータ	説明	構文
XWAIT	<p>標準 EXEC CICS WAITCICS (XWAIT=NO) または WAIT EVENTS EXTERNAL (XWAIT=YES) が Adabas 8 タスク関連ユーザー出口 (TRUE) によって実行されるかどうかを示します。XWAIT=YES がデフォルトです。</p> <p>CICS/TS 1.1 以上では CICS WAIT EVENTS EXTERNAL (XWAIT=YES) が望ましいインターフェイスです。</p> <p>CICS WAITCICS ステートメント (XWAIT=NO) は、CICS/MVS 2.1.2 および CICS/VSE 2.1~2.3 での使用のために提供されています。これは、CICS/TS 1.1 以上でも使用できますが、CICS トランザクションのパフォーマンスを低下させるか、またはビジーな CICS 環境では予期しない結果を招くことがあります。</p> <p>注意: CICS/ESA 3.3 配下で XWAIT=NO を指定する場合、IBM APAR PN39579 を CICS/ESA 3.3 システムに適用しなければなりません。CICS/TS 1.1 以上では、この APAR は必要ありません。</p>	XWAIT={NO YES}



Notes:

1. XWAIT=NO が指定されている場合、ADACICT (Adabas 8 TRUE) モジュールは EXEC CICS WAIT EVENT コマンドの代わりに EXEC CICS WAITCICS コマンドを発行します。XWAIT=YES が指定されている場合、CICS/TS バージョン 1.1 以上を搭載した高トランザクション処理の CICS システムにおける WAIT リストと ECB リストの使用 방법이、推奨される IBM 仕様と同じになります。
2. EXEC CICS コマンドはすべて CICS プリプロセッサによって処理されます。LGBLSET パラメータにより、後続のアセンブリステップはいくつかのステートメントをスキップします。

XWAIT ポスティングメカニズム

CICS WAITCICS (XWAIT=NO) は、指定された ECB のソフトポストをサポートすることができます。これには、ハードポストワークは全ワークサイクル上で CICS に処理されないため、優先度の低い CICS 環境のディスパッチ可能なワークユニットになるという短所があります。

これに反して、EXEC CICS WAIT EXTERNAL (XWAIT=YES) により、CICS がその特別ポスト出口コードを使用できるようにし、全 CICS ワークサイクル上でチェックと処理 (ポストされた場合) を常に行います。

各種の CICS WAIT コマンドとそのハードおよびソフトポスティングメカニズムの関係の違いの詳細については、IBM の『CICS Application Programming Reference Guide』および添付テキストの IBM APAR PN39579 または IBM InfoLink サービスの Item RTA000043874 を参照してください。

XWAIT および Adabas SVC / ルーター

Adabas SVC は `XWAIT=YES` 設定と完全な互換性があります。SVC は、Adabas コマンドレベル リンクルーチンを使用する CICS 配下の Adabas コール元に対して必要なハードポストを実行します。ハードポストを必要としないバッチコール元に対しては、同じ SVC がソフトポストを実行します。

5 UES 対応データベースの接続

- 概要 110
- ダイレクト TCP/IP リンク経由の接続 111
- TCP/IP リンクのアクティブ化 114

このchapterでは、次のトピックについて説明します。

概要

Adabas バージョン7より前のバージョンでは、必要に応じて、Entire Net-Work が全データをメインフレーム Adabas 用に ASCII から EBCDIC に変換していました。バージョン7から、Adabas は、ユニバーサルエンコーディングサポート (UES) と呼ばれる独自のデータ変換機能を備えるようになりました。Entire Net-Work は、データを変換するターゲットデータベースに接続されるとそれを検出し、データを変換しないで Adabas に渡します。

ユニバーサルエンコーディングサポートは、次の場合にアクティブになっている必要があります。

- Adabas ニュークリアス。
- Adabas リンクルーチン。Adabas バージョン7の場合、リンクルーチン ADALNK、ADALNKR、ADALCO の UES はデフォルトで有効になっています。Adabas 8 の場合、すべてのリンクルーチンに対して UES が有効になっています。リンクルーチンで UES の有効/無効を変更する方法については、「[TP モニタを使用する場合の Adabas のインストール](#)」の「[UES 有効リンクルーチン](#)」以降のセクションを参照してください。

UES 対応のデータベースは、Com-plete または Smarts 経由で異なるアーキテクチャのマシンに接続したり、Entire Net-Work 経由で接続したりできます。また、オプションで、z/OS 環境で、Software AG の Jadabas などの Web ベースのアプリケーションまたは PC ベースのアプリケーションから Adabas ニュークリアスにダイレクト TCP/IP リンクで接続することができます。Com-plete または Smarts 経由の接続では、Adabas Com-plete リンクルーチンを、Entire Net-Work 経由の接続では Adabas バッチリンクルーチンを使用します。ダイレクト TCP/IP リンクについては、このsectionで説明します。

- Adabas データベース。詳細については、「[ユニバーサルエンコーディングサポート \(UES\)](#)」および「[ADADEF ユーティリティ：データベースの定義](#)」、「[ADACMP ユーティリティ：データの圧縮/圧縮解除](#)」を参照してください。

 **Note:** UES 有効リンクルーチンおよび UES 有効ニュークリアスの使用は、アプリケーションに対して透過的です。これはエンコード変換サポートを必要としないアプリケーションでも同じです。したがって、UES をすでに有効にしている場合、無効にする必要はありません。

ダイレクト TCP/IP リンク経由の接続

TCP/IP リンクでは、リエントラント ADALNKR モジュールを LNKUES とリンクするだけでなく、変換テーブルを修正および再アセンブルして、さらに生成されたオブジェクトを Adabas steplib で使用できるようにする必要があります。

UES 対応のデータベースは、Adabas リエントラントバッチまたは TSO リンクルーチン ADALNKR を使用して、TCP/IP 経由で直接接続されます。ADALNK のアセンブルとリンクを行うためのサンプルジョブストリームが ALNKLNKR です。

- 手順 1：ADALNKR モジュールの Adabas ロードライブラリへのアセンブル
- 手順 2：2 つの変換テーブルの Adabas ロードライブラリへのアセンブル（SMA ジョブ番号 I056）
- 手順 3：変換テーブルと LNKUES の ADALNKR へのリンク
- 手順 4：ADALNKR を Adabas ニュークリアスから利用できるようにする

手順 1：ADALNKR モジュールの Adabas ロードライブラリへのアセンブル

TCP/IP 経由でデータベースの UES サポートを有効にするには、修正したバッチ ADALNKR を準備する必要があります。

▶ 手順 5.1. 修正したバッチ ADALNKR を準備するには、次の手順に従います。

- 1 ソース ADALNKR を更新します。

```
&RENT SETB 1
SVCNR EQU nnn (hard-coded SVC number)
```

- 2 修正したバッチ ADALNKR をアセンブルおよびリンクします。

MVSJOBS メンバ ALNKLNKR を修正して、次のようにアセンブルおよびリンクします。

- 必要なすべてのジョブカード情報を指定します。
- バージョン、リビジョンレベル、および SM レベルの記号パラメータ値 (vrs) を確認します。この値は、Adabas のソースとロードライブラリのレベルを反映している必要があります。

- SAGASM および LINKALL インラインプロシージャの SYSLIB、SYSIN、SYSLMOD、SYSLIN のデータセット名を確認します。

```
// JOB
//SAGASM PROC MEM=,
// VRS=
//ASM EXEC PGM=ASMA90,
// PARM='ASA,NODECK,OBJECT,USING(MAP),XREF(SHORT),TERM'
//SYSUT1 DD SPACE=(4096,(120,120),,,ROUND),UNIT=VIO,DCB=BUFNO=1
//SYSTEM DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSLIB DD DISP=SHR,DSN=ADABAS.&VRS..SRCE
// DD DISP=SHR,DSN=SYS1.MACLIB
// DD DISP=SHR,DSN=SYS1.MODGEN
//S//SYSLIN DD DSN=&&OBJ,SPACE=(3040,(40,40),,,ROUND),UNIT=VIO,
// DISP=(MOD,PASS),
// DCB=(BLKSIZE=3040,LRECL=80,RECFM=FBS,BUFNO=1)
//LINK EXEC PGM=HEWL,REGION=2M,COND=(5,LT,ASM),
// PARM='XREF,LIST(ALL),LET,MAP'
//SYSPRINT DD SYSOUT=*
//SYSLIN DD DSN=&&OBJ,DISP=(OLD,DELETE)
//SYSLMOD DD DISP=SHR,DSN=ADABAS.&VRS..LOAD(&MEM)
// PEND
// *
//ADALNKR EXEC SAGASM,VRS=Vvrs,MEM=ADALNKR
//LINK.SYSIN DD *
MODE AMODE(31) RMODE(24)
ENTRY ADABAS
NAME ADALNKR(R)YSIN DD DISP=SHR,DSN=ADABAS.&VRS..SRCE(&MEM)
```

手順 2：2 つの変換テーブルの Adabas ロードライブラリへのアセンブル (SMA ジョブ番号 I056)

デフォルトのまたは修正した、ASCII から EBCDIC への変換テーブル、EBCDIC から ASCII への変換テーブルをアセンブルします。

```
/*
//ASC2EBC EXEC SAGASM,VRS=Vvrs,MEM=ASC2EBC
//LINK.SYSIN DD *
MODE AMODE(31) RMODE(ANY)
ENTRY ASC2EBC
NAME ASC2EBC(R)
/*
//EBC2ASC EXEC SAGASM,VRS=Vvrs,MEM=EBC2ASC
//LINK.SYSIN DD *
```

```

MODE AMODE(31) RMODE(ANY)
ENTRY EBC2ASC
NAME EBC2ASC(R)

```

手順 3：変換テーブルと LNKUES の ADALNKR へのリンク

次に ADALNKR を LNKUES および修正して再アセンブルした変換テーブルに（再）リンクする必要があります。

ADALNKR、ASC2EBC、EBC2ASC、LNKUES、およびその他のユーザ出口モジュールを、UES 対応の最終的な ADALNKR モジュールにリンクします。このロードモジュールを USER.LOAD ライブラリに配置します。SYSLMOD ステートメントの &USERLIB シンボルをユーザロードライブラリに一致するよう修正します。

```

/*
//LINKALL PROC VRS=,USERLIB=
//LKED EXEC PGM=HEWL,REGION=2M,COND=(5,LT),
// PARM='XREF,LIST(ALL),LET,MAP,NCAL'
//SYSPRINT DD SYSOUT=*
//ADALIB DD DISP=SHR,DSN=ADABAS.&VRS..LOAD
//SYSLMOD DD DISP=SHR,DSN=&USERLIB
//SYSLIN DD DDNAME=SYSIN
// PEND
//LINKUES EXEC LINKALL,VRS=Vvrs,
// USERLIB='YOUR.USER.LOADLIB'
//LKED.SYSIN DD *
MODE AMODE(31) RMODE(24)
INCLUDE ADALIB(ADALNKR)
INCLUDE ADALIB(LNKUES)
INCLUDE ADALIB(ASC2EBC)
INCLUDE ADALIB(EBC2ASC)
ENTRY ADABAS
NAME ADALNKR(R)
/*

```

手順 4：ADALNKR を Adabas ニュークリアスから利用できるようにする

（再）リンクされた ADALNKR は Adabas ニュークリアスから利用できるようにする必要があります。

TCP/IP リンク経由で Adabas バージョン 7 を直接コールしている場合、Adabas ニュークリアスから正しい ADALNKR を使用できないときは、レスポンスコード 148、空のバッファなど、Adabas に予期しない結果が発生します。

TCP/IP リンクのアクティブ化

▶手順 5.2. Adabas ニュークリアスへのダイレクト TCP/IP リンクをアクティブ化するには、次の手順に従います。

- 1 ADARUN パラメータ `TCPIP=YES` を設定します。
- 2 URL を指定します。

URL の指定

URL は、RFC の URL 仕様に準拠した 20 バイトのアドレスです。

ダイレクト TCP/IP リンクをアクティブにするための URL は、ADARUN パラメータ `TCPURL` で次のように指定します。

```
ADARUN PROG=ADANUC,TCPIP=YES,TCPURL=api-name://stackid:port-number
```

上記の意味は次に示すとおりです。

- *api-name* は使用するアプリケーションプログラミングインターフェイス (API) を表す 1~3 文字の値です。現在、IBM TCP/IP スタック (HPS、OES) の API がサポートされています。
- *stackid* は使用するスタックを識別する 1~8 文字の値です。HPS API の場合、これは TCP/IP 開始タスクの名前になります。OES API の場合、値は必要ありません。ILK API の場合、サブシステム ID になります。
- *port-number* は 1~5 文字の 10 進数です。

例

```
ADARUN PROG=ADANUC,TCPIP=YES,TCPURL=HPS://STACKNAME:1234
ADARUN PROG=ADANUC,TCPIP=YES,TCPURL=OES://:1234
ADARUN PROG=ADANUC,TCPIP=YES,TCPURL=ILK://ILZ5:1234
```

URL の管理

オプションで、オペレータコマンド `TCPIP` を使用して、最初の URL と、追加の URL を指定することができます。

```
TCPIP={ OPEN=url|CLOSE=url | CLOSE }
```

`url` は、オープンまたはクローズする TCP/IP リンクの URL です。フォーマットは `ADARUN TCPURL` パラメータと同じです。

```
api-name://stackid:port-number
```

このコマンドは、Adabas ニュークリアスへの TCP/IP リンクをオープン/クローズしたり、すべてのリンクをクローズすることができます。 `ADARUN TCPPIP=YES` が指定され、その設定のすべての条件が満たされている場合のみ使用できます。このコマンドは、 `ADARUN TCPURL` パラメータで設定された URL を閉じるために使用されます。または追加の TCP/IP リンクを開いたり閉じたりするために使用されます。

例

```
TCPIP=OPEN=ILK://ILZ5:1234  
TCPIP=CLOSE=ILK://ILZ5:1234
```

```
To close all open URLs:  
TCPIP=CLOSE
```


6 デバイスとファイルについて

- サポートされるデバイスタイプ 118
- ECKD デバイス 119
- 新規デバイスの追加 119

このセクションでは、次のデバイスおよびシステムファイルについて説明します。

サポートされるデバイスタイプ

次の表に、Adabas を z/OS で使用する場合にサポートされるデバイスタイプの標準的な特性を示します。Adabas ブロックサイズとトラック当たりの RABN は、各デバイスタイプの Adabas コンポーネント別に記載されています。

デバイス	トラック数/シリンダ	ASSO	DATA	WORK	PLOG/RLOG	CLOG	TEMP/SORT/DSIM	注
0512	16	2044:8	4092:4	8192:2	8192:2	8192:2	8192:2	
3310	11	2044:8	4092:4	4096:4	4096:4	4096:4	8192:2	
3330	19	1510:8	3140:4	4252:3	4252:3	3156:4	3140:4	
3340	12	1255:6	2678:3	3516:2	3516:2	3516:2	3500:2	
3350	30	1564:11	3008:6	4628:4	4628:4	3024:6	3008:6	
3370	12	2044:15	3068:10	5120:6	5120:6	3072:10	7680:4	
3375	12	2016:15	4092:8	4096:8	4096:8	4096:8	8608:4	
3380	15	2004:19	4820:9	5492:8	5492:8	4820:9	7476:6	3
3390	15	2544:18	5064:10	5724:9	5724:9	5064:10	8904:6	3
8345	15	4092:10	22780:2	22920:2	22920:2	22920:2	22920:2	
8350	30	3008:6	6232:3	9442:2	9442:2	9442:2	9442:2	1
8380	15	3476:12	6356:7	9076:5	9076:5	9076:5	9076:5	1
8381	15	3476:12	9076:5	11476:4	11476:4	9076:5	9076:5	1
8385	15	4092:10	23292:2	23468:2	23468:2	23468:2	23468:2	1
8390	15	3440:14	6518:8	10706:5	10706:5	8904:6	8904:6	1
8391	15	4136:12	10796:5	13682:4	13682:4	8904:6	18452:3	1
8392	15	4092:12	12796:4	18452:3	18452:3	18452:3	18452:3	1
8393	15	4092:12	27644:2	27990:2	27990:2	27990:2	27990:2	1
9332	6	2044:10	4092:5	5120:4	5120:4	10240:2	10240:2	2
9335	6	2556:14	3580:10	5120:7	5120:7	7168:5	7168:5	
9345	15	4092:10	7164:6	11148:4	11148:4	22920:2	22920:2	3



Notes:

- 8350、838_n、および 839_n は物理的には 3350、3380 および 3390 に含まれる擬似デバイスタイプです。ただし、前者の標準ブロックサイズの方が長いこともあります。
- ここに記載されたシリンダ当たりのトラック数は形式的な値です。

3. MVS、VSE、または VM/ESA 配下の IBM RAMAC 9394 は、3390 モデル 3、3380 モデル K、または 9345 モデル 2 デバイスをエミュレートします。

ECKD デバイス

Adabas は、3990 コントローラおよび ESCON チャネルを備えた IBM 3390 などの ECKD DASD デバイスをサポートしています。

オープン処理中に、ADAIOR は、ASSO、DATA、WORK、SORT、および TEMP データセットに使用されている DASD デバイスタイプを決定します。そのとき、Adabas は各 Adabas データベースコンポーネントに関する情報メッセージを発行します。メッセージの *type* はコンポーネントです。

```
ADA164 ... FILE DDtype HAS BEEN OPENED IN ckd/eckd MODE - RABN SIZE rabn-size
```



Note: ファイルがオープンされるのは CKD モードのみなので、ファイル内での ECKD エクステンと CKD エクステンとの混在は避けることを強くお勧めします。エクステンを混在させると、ファイル I/O 処理が実行されるときに、パフォーマンスが低下することがあります。

新規デバイスの追加

ユーザー定義ブロックサイズを含む新規デバイスタイプに対するサポートは、このために予約されたデバイスコンスタントエントリテーブル (TDCE) の 1 つを修正することによって、ADAIOR に実装することができます。

TDCE は X'40' バイト長で、最初の未使用の TDCE はその先頭 2 バイト (TDCDT) の X'0000' で示されます。

バージョン 6.2 より前のすべての Adabas バージョンでは、最初の TDCE のアドレスはオフセット ADAIOR+X'34' にあります。

Adabas バージョン 6.2 の場合、TDCE エントリは ADAIOR CSECT TDCON にあります。最初の TDCE エントリはオフセット 0、最初の未使用の TDCE エントリはオフセット X'400' にあります。

Adabas バージョン 7.1 の場合、TDCE エントリは ADAIOS CSECT TDCON にあります。最初の TDCE エントリはオフセット 0、最初の未使用の TDCE エントリはオフセット X'580' にあります。

この情報は、TDCE エントリを追加するときに有効です。

- 最初の未使用 TDCE に ZAP する情報
- デバイスブロックサイズ定義の一般的ルール
- 最大シーケンシャルブロックサイズ
- アソシエータまたはデータストレージのブロックサイズのルール
- WORK データセットのブロックサイズのルール
- TEMP/SORT データセットのブロックサイズのルール
- PLOG または SIBA のブロックサイズのルール
- I_PPT のシーケンシャルプロテクションログのブロックサイズ

最初の未使用 TDCE に ZAP する情報

次の表の情報を最初の未使用 TDCE に ZAP する必要があります。TDCE を変更する場合、「デバイスブロックサイズ定義の一般的ルール」に従う必要があります。

ラベル	オフセット	内容
TDCDT	00	符号なし 10 進数のデバイスタイプ (X'3385')。全 TDCE 間でユニークな数字でなければなりません。
TDCKSN	02	コンスタントセット番号。値 X'2B' または X'2E' からユニークに選択する必要があります。
TDCF	03	CKD デバイスの場合 TDCFCKD (X'40')、ECKD デバイスの場合 TDCFCKD (X'60')、(ユーザ定義でない) ECKD デバイスの場合 TDCFCKD (X'61') のフラグビットをセットする必要があります。
TDCDT1	04	(注 1 参照)
TDCDT2	05	(注 1 参照)
TDCDT3	06	(注 1 参照)
TDCDT4	07	(注 1 参照)
TDCMSBS	08	「最大シーケンシャルブロックサイズ」を参照してください。
TDCTPC	0A	シリンダ当たりのトラック数
TDCCIPT	0C	(注 2 参照)
TDCBPCI	0E	(注 2 参照)
TDCABPT	10	トラック当たりのアソシエータブロック数。
TDCABS	12	アソシエータブロックサイズ。
TDCACPB	14	(注 2 参照)
TDCDBPT	16	トラック当たりの DATA ブロック数。
TDCDBS	18	データストレージブロックサイズ
TDCDCPB	1A	(注 2 参照)
TDCWBPT	1C	トラック当たりの WORK ブロック数。
TDCWBS	1E	WORK ブロックサイズ。
TDCWCPB	20	(注 2 参照)
TDCTSBPT	22	トラック当たりの TEMP または SORT ブロック数

ラベル	オフセット	内容
TDCTSBS	24	TEMP または SORT ブロックサイズ。
TDCTSCPБ	26	(注 2 参照)
TDCPBPT	28	トラック当たりの PLOG ブロック数。
TDCPBS	2A	PLOG ブロックサイズ。
TDCPCPB	2C	(注 2 参照)
TDCCBPT	2E	トラック当たりの CLOG ブロック数。
TDCCBS	30	CLOG ブロックサイズ。
TDCCCPB	32	(注 2 参照)



Notes:

1. デバイスタイプを識別するための、1 つまたは複数のオペレーティングシステム依存のコード。z/OS の場合は、UCBTBYT4 の UCB ユニットタイプになります。
2. z/OS オペレーティングシステムでは使用されません。

デバイスブロックサイズ定義の一般的ルール

Adabas デバイスブロックサイズを定義するとき、次の一般的ルールに従う必要があります。

- すべてのブロックサイズは必ず 4 の倍数にします。
- 1 つのブロックを複数のトラックに分割することはできません（つまり、ブロックサイズはトラックサイズ以下にする必要があります）。

最大シーケンシャルブロックサイズ

新しいデバイスを追加するときに、最大シーケンシャルブロックサイズも指定する必要があります。最大シーケンシャルブロックサイズに設定する値は TDCMSBS です。ADAIOR TDCE テーブルの先頭からオフセット X'08' の位置にあります。

デバイスタイプに応じて、TDCMSBS 値は次の値にする必要があります。

デバイスタイプ	最大ブロック長
0512	32760
3310	32760
3330	13030
3340	8368
3350 (8350)	19069
3370	32760
3375	17600

デバイスタイプ	最大ブロック長
3380 (8380/81)	23476
339 _n	27998
8380/1/5	23476
839 _n	27998
9332	32760
9335	32760

 **Note:** デバイスによっては、より小さいブロックサイズを使用することで最も効率的になることに注意してください。例えば 3380 デバイスに対する最大ブロックサイズを 23476 バイトに設定することで 1トラック当たり 2ブロック確保することができます。

アソシエータまたはデータストレージのブロックサイズのルール

アソシエータおよびデータストレージブロックサイズには次のルールが適用されます。

- アソシエータブロックサイズは最大 FDT サイズの 4 分の 1 より大きくして、さまざまな管理ブロック (RABN1~30) および FCB の定義を受け入れられる十分な大きさにする必要があります。
- アソシエータとデータのブロックサイズは、Adabas バッファプールスペースを節約するために 256 の倍数より 4 バイト少なくしてください (例: 1020)。
- アソシエータおよびデータのブロックサイズはシーケンシャルブロックサイズより最低 32 小さくする必要があります。
- データブロックサイズは (最大圧縮レコード長+10+パディングバイト) より大きくする必要があります。

WORK データセットのブロックサイズのルール

WORK ブロックサイズは (最大圧縮レコード長+110) または (アソシエータブロックサイズ+110) のどちらか大きいほうより大きくする必要があります。

TEMP/SORT データセットのブロックサイズのルール

ADAM ダイレクトアドレッシングを使用する場合、サイズは次のとおりです。

```
size > (maximum compressed record length + ADAM record length + 24);
size > 277 (maximum descriptor length + 24)
```

ただし、TEMP と SORT は一般的に読み込まれた後に書き込まれます。したがって、TEMP/SORT ブロックサイズを大きくした方が効率はよくなります。

TEMP および SORT のブロックサイズは DATA のブロックサイズより大きくする必要があります。

PLOG または SIBA のブロックサイズのルール



Note: プロテクションログファイルに 3480/3490 テープカートリッジ圧縮 (IDRC) を使用することはお勧めしません。ADARES BACKOUT 機能を実行する場合、z/OS では圧縮データの処理に少なくとも 2 倍の時間がかかります。

PLOG または SIBA ブロックサイズには次のルールが適用されます。

- PLOG または SIBA ブロックサイズは (最大圧縮レコード長 + 110) または (アソシエータブロックサイズ + 110) のどちらか大きいほうより大きくする必要があります。
- PLOG/SIBA を最大データストレージブロックサイズより大きく定義することもお勧めします。これにより、オンライン ADASAV 処理中にデータストレージブロックを分割することによる I/O 増加が回避できます。

シーケンシャルファイルのブロックサイズ (BLKSIZE) は次のように決定されます。

```
if PTTF(JCL) then BLKSIZE is taken from file assignment statement or label;
if PTTMBS > 0 then BLKSIZE = PTTMBS;
if PTTMBS = 0 then
if tape then BLKSIZE = 32760;
else BLKSIZE = TDCMSBS;
else if BLKSIZE in file assignment statement or label then use it;
if PTTF(OUT) then
if QBLKSIZE > 0 then BLKSIZE = QBLKSIZE;
if tape then BLKSIZE = 32760;
else BLKSIZE = TDCMSBS;
else error.
```



Note: QBLKSIZE は ADARUN パラメータです。

I_PPT のシーケンシャルプロテクションログのブロックサイズ

ロードモジュール ADAIOR の CSECT I_PPT の対応する PTT エントリで、シーケンシャルプロテクションログのブロックサイズを大きくしなければならないことがあります。

最初の PTT エントリのアドレスは ADAIOR+X'4C8' のフルワードに含まれています。

PTT エントリは CSECT I_PPT のオフセット 0 から始まります。

各 PTT エントリは X'10' バイト長で、次のような構成になっています。

ラベル	オフセット	内容
PTTPN	00	プログラム番号。
PTTFT	01	ファイルタイプ。
PTTN	02	DD名 (2~8文字)。
PTTF	08	フラグ： OUT (X'80') 出力 BSAM (X'40') BSAM BACK (X'20') 後方読み込み JCL (X'10') DATADEF ステートメントまたはラベルから取得された BLKSIZE/LRECL/RECFM UNDEF (X'04') 未定義のレコードフォーマット VAR (X'02') 可変レコードフォーマット
-	09	予約済み
PTTMBSZ	0C	最大ブロックサイズ。

シーケンシャルプロテクションログに対する PTT エントリは、先頭 2 バイトの X'12F1' で識別できます。

7 AOS デモバージョンのインストール

- AOS デモのインストール手順 126
- Natural Security を使用する場合の AOS のインストール 127
- AOS デモバージョンデフォルトの設定 127

このセクションでは、Adabas Online System (AOS) デモバージョンを z/OS または FACOM MSP システムにインストールする方法について説明します。Software AG の System Maintenance Aid (SMA) を使用しているシステムに AOS をインストールする場合は、このマニュアルの操作環境に対応した Adabas のインストールに関するセクションを参照してください。SMA の詳細については、*System Maintenance Aid* のマニュアルを参照してください。

Notes:

1. フルバージョンの AOS のインストールについては、『*Adabas Online System*』を参照してください。
2. デモバージョンまたはフルバージョンの AOS をインストールすると、デモバージョンの Adabas Vista (AVI)、Adabas Fastpath (AFP)、Adabas SAF Security (AAF)、および Adabas Transaction Manager (ATM) も自動的にインストールされます。

AOS デモバージョンには Natural バージョン 4.1 以上が必要です。

AOS デモのインストール手順

▶ **手順 7.1. System Maintenance Aid を使わないで AOS をインストールするには、次の手順に従います。**

- 1 Com-Plete または CICS 環境では、正しいオブジェクトモジュールを Natural TP ニュークリアスとリンクします。

分割された Natural ニュークリアスをインストールするには、AOSASM モジュールをニュークリアスの共有部分にのみリンクし、スレッド部分にはリンクしないようにする必要があります。
- 2 Natural INPL を実行します。

AOS デモバージョンを含むテープには、Natural 4.1 の INPL フォーマットのデータセットが含まれています。AOS デモバージョンのプログラムはライブラリ SYSAOS に格納されています。
- 3 Natural ユーティリティ ERRLODUS を使って ADA エラーメッセージをロードします。

エラーメッセージは、テープに含まれる ERRN フォーマットのデータセットに格納されています。

ERRLODUS ユーティリティについては『*Natural ユーティリティマニュアル*』を参照してください。
- 4 アプリケーションライブラリ SYSAOS にログオンし、コマンド DBMENU を入力して AOS デモバージョンを実行してください。

Natural Security を使用する場合の AOS のインストール

Natural Security がインストールされている場合、最低でもライブラリ SYSAOS を定義してください。

必要に応じて次のライブラリを定義します。

- Adabas Vista の場合：SYSAVI と SYSMV_{vrs}
- Adabas Fastpath の場合：SYSAFP と SYSMW_{vrs}
- Adabas SAF Security の場合：SYSAAF と SYSMX_{vrs}
- Adabas Transaction Manager の場合：SYSATM と SYSMT_{vrs}

Software AG では、SYSAOS およびその他の定義するライブラリに、保護属性を付けて定義することをお勧めします。

SYSAOS のスタートアッププログラムとして DBMENU を指定します。その他のライブラリにはスタートアッププログラム名を指定しないでください。

Natural Security は、Adabas Online System Security を実装する前にインストールする必要があります。詳細は、『*Adabas Security Manual*』を参照してください。AOS セキュリティを使用する場合の Natural Security のインストールについては、『*Natural Security マニュアル*』を参照してください。

Natural Security には、AOS デモバージョンの Natural コマンドモードの LOGON 機能が起動されるとき、すべてのオープンしているデータベースを自動的にクローズする機能があります。

AOS デモバージョンデフォルトの設定

AOS デモバージョンのオペレーションを制御するパラメータは、Natural インストール時に Natural プログラム AOSEX1 のデフォルトを変更することによって設定できます。次の表に、パラメータおよび有効な値の一覧を示します。デフォルトは下線付きで示されています。

パラメータ	有効値/デフォルト	機能
ADMIN-LEVEL (N1)	0~9/ (6)	管理レベル。ADABAS 環境にエラー状態を発生させる可能性のある特定の機能へのアクセスを許可します。8以上に設定すると、CATCHRSP-CODE ダイレクトコマンドの実行が許可されます。9以上に設定すると、ZAP 機能の実行が許可されます。
AOS-END-MSG	Yes (<u>Y</u>) /No (N)	AOSセッション終了メッセージを表示するかどうかを指定します。

パラメータ	有効値/デフォルト	機能
AOS OGO	Yes (<u>Y</u>) /No (N)	AOS ロゴを表示するかどうかを指定します。
BATCH-ERROR (A1)	Yes (<u>Y</u>) /No (N)	バッチジョブのコンディションコード。AOSがバッチジョブから実行されているときにエラー状態が発生した場合、BATCH-ERRORがYに設定されていると、AOSはコンディションコード8で終了します。この機能を完全に実装するには、すべてのAOSプログラムを修正する必要があります。そのため時間がかかります。
BLK-CYL	シリンダ (C) ブロック (B)	ブロック/シリンダによるスペース制御。
CMD-INT	Natural (N) AOS (<u>A</u>)	無効なAOSコマンドのパススルー制御。 "N"は無効なコマンドをNaturalに渡します。 "A"は無効なコマンドのエラーメッセージを表示します。
CPEXLIST	No (<u>N</u>) : 標準リスト Yes (Y) : 拡張	拡張チェックポイントリストを表示するかどうかを指定します。
EXF-UTI	E : EXF 例外 U : UTI 例外	UTI/EXF ファイルロック例外。
MAX-AC-IOS	0~999999 (<u>150</u>)	コンバータブロック限界値を読み込むAC。
MAXANZ	1~99999999 (<u>100</u>)	ユーザーキューエレメントの最大表示数。
NR-EXT	1, 2, 3, <u>4</u> , 5, 6, 7, 8, 9	ファイルのリストに対する"クリティカルな"エクステント限界値。
NR-PERCENT(N3)	1~99/ (<u>89</u>)	レポート機能。NR-PERCENTは、AC/UI/NI/DSテーブルタイプで到達したエクステントの割合に関するクリティカルファイルの表示の限界値。NR-PERCENTを超える値は強調表示されます。
PURGE-UQE	Yes (Y) /No (<u>N</u>)	ユーザーキューエレメントを削除するかどうかを指定します。
SAVEFDT	Yes (Y) /No (<u>N</u>)	削除したファイルのFDTを保存するかどうかを指定します。
STATINTV	1~9999 秒 (<u>60</u>)	統計を集める周期。
TID-DISPLAY	B, A, <u>I</u>	"ユーザーキュー表示"機能におけるTIDの表示制御。 "B" = バイナリ TID 表示 "A" = 英数字 TID 表示 "I" = A~Z/0~9 の場合は英数字、その他はバイナリ
TIMELA	0~99999999 秒 (Q : 無制限)	直前のn秒間にアクティビティのあるユーザーキューエレメントを表示します。
TIN-JOBN	T (開始時刻) J (ジョブ名)	"コマンドキュー表示"機能でジョブ名と開始時刻のどちらを表示するかを指定します。

デフォルトを変更するには、NaturalAOSEX1 プログラムを編集し、次の例に示すように、プログラムリストのデフォルトエリアを直接変更する必要があります。

```
.
.
.
DEFINE DATA PARAMETER USING P-AOSEX1
END-DEFINE
*
* SET THE DEFAULTS
*
ADMIN-LEVEL = '6'      (Allows access to certain functions that can cause error
conditions in the ADABAS environment)
AOS-END-MSG = 'Y'      (Display end-of-session message)
AOS-LOGO = 'Y'         (Adabas Online System logo display-set to 'N' for no logo
display)
BATCH-ERROR = 'N'      (If BATCH-ERROR is set to "Y", AOS will terminate with a
condition code of 8 if an error occurs.)
BLK-CYL = 'B'          (Space allocation default-set to 'C' for cylinders)
CMD-INT = 'A'          (Pass invalid Adabas commands to (N)atural, or intercept (A))
CPEXLIST = 'N'         (Checkpoint list control-set to 'Y' for extended checkpoint list)
EXF-UTI = 'U'          (File locking exception-set to 'E' to except files in EXF status)
MAXANZ = 100           (Maximum user queue elements displayed. range: 1 - 99,999,999
elements)
NR-EXT = 4             (Critical extent threshold-1, 2, 3, 4, 5, 6, 7, 8 or 9)
NR-PERCENT = '89'      (NR-PERCENT is a threshold value for the display of critical
files)
MAX-AC-IOS = 150       (AC read converter block threshold)
PURGE-UQE = 'N'        (Remove element from user queue. Pre-5.1 default is "Y")
SAVEFDT = 'N'          (Keep old FDT for SAVE operation-set to 'Y' to save FDTs)
STATINTV = 60          (Statistic-gathering time. range: 1 - 9999)
TID-DISPLAY = 'I'      (TID display control: B=binary, A=alpha, I=normally alpha,
special characters as binary)
TIMELA = 0             (Include activity in last 'n' seconds. range: "all" (0) -last
99,999,999 seconds)
TIN-JOBN = 'J'         (Command queue display-"J" for jobname, "T" for "time in queue" )
*
END
```


8 Recovery Aid (ADARAI) のインストール

- ADARAI のインストール概要 132
- ADARAI のインストール手順 132

このセクションでは、Adabas Recovery Aid (ADARAI) のインストール方法について説明します。

ADARAI のインストール概要

Adabas Recovery Aid をインストールするには、次の作業が必要です。

- リカバリログの割り当て
- インストール用のスケルトンジョブストリームの修正 (詳細は、『*Adabas オペレーションマニュアル*』を参照)
- 必須のニュークリアス実行/ユーティリティジョブコントロールの更新 (Recovery Aid データ定義ステートメントの追加)
- Adabas/ADARAI ユーティリティコンフィグレーションのインストール
- ログの生成を開始するための ADARAI PREPARE と保存オペレーションの実行

ADARAI のインストール手順

スケルトンジョブストリームの修正を除く、具体的なインストール手順は次のとおりです。

▶ **手順 8.1. Adabas Recovery Aid をインストールするには、次の手順に従います。**

- 1 DDRLOGR1 ファイルを定義してフォーマットします。

ADAFRM RLOGFRM 機能を使用して、RLOG をフォーマットします。
- 2 ニュークリアスジョブストリーム、およびデータベースの更新または保存を行うすべてのユーティリティ (RLOG ファイルに書き込むユーティリティ) に DDRLOGR1 DD または DLBL ステートメントを追加します。

ADARAI がデータベース内でアクティブなときに (PREPARE 機能の実行後に)、これらのユーティリティが実行される場合は、DDRLOGR1 または DDDLBL ステートメントを含める必要があります。

次のユーティリティはデータベースを更新し、RLOG に書き込みます。

```
ADAORD (all STORE and REORDER functions)
ADALOD (all functions)
ADAINV (all functions)
ADARES REGENERATE/BACKOUT database
ADASAV RESTORE (all functions) and RESTPLOG
ADADEF NEWWORK
```

次のユーティリティはデータベースを保存し、RLOG に書き込みます。

```
ADASAV SAVE (all functions)
ADAORD RESTRUCTURE
ADAULD
```

次のユーティリティ機能はリカバリに影響し、RLOG に書き込まれます。

```
ADARES PLCOPY/COPY
ADASAV MERGE
```

さらに、Adabas ニュークリアスは起動時と終了時に RLOG に書き込みます。ニュークリアスは、ADADBS または Adabas Online System 機能が処理される時に、これらのイベントが ADARAI に記録され、リカバリ処理に使用できるように、RLOG にチェックポイント情報を書き込みます。

- 3 データベースに ADARAI をインストールします。

ADARAI の PREPARE 機能を実行します。ADARAI PREPARE は、ADARAI がインストールされたことを示すように ASSO GCB を更新します。また、必要な ADARAI 情報（世代番号、RLOG サイズなど）を含むコントロールレコードを RLOG ファイルに書き込みます。

- 4 最初の ADARAI 世代を作成します。

ADASAV SAVE (データベース) を実行し、RLOG 情報のロギングを開始します。詳細は、『Adabas ユーティリティマニュアル』を参照してください。

ADARAI がデータベース内でアクティブになると、プロテクションロギングを常に使用する必要があります。

9 エラー処理およびメッセージバッファリング機能のインストール

このセクションでは、エラー処理およびメッセージバッファリング機能のインストール方法について説明します。

▶**手順 9.1. エラー処理およびメッセージバッファリング機能をインストールするには、次の手順に従います。**

- 1 ADARUN SMGT=YES を指定します。メッセージバッファリングを使用する場合、0 より大きい値で ADARUN MSGBUF も指定します。

エラー処理ツールを稼働させるために ADARUN SMGT=YES が指定されると、初期化モジュール ADAMXI が ADARUN によりロードされ、その後セッションオープン中にコールされます。

- エラー処理ヘッダー／環境が初期化されます。
- ADARUN MSGBUF に 0 より大きい値が指定されると、メッセージバッファが初期化されます。
- エラー処理モジュールは ADAIOR によってメモリにロードされます。
- Adabas モジュールテーブルが構築されます。
- 提供されたあらゆるエラー処理ユーザー出口が初期化されます。
- デフォルトのリカバリプラグイン (PIN) モジュール ADAMXY がインストールされます。
- プログラムチェックおよび異常終了ハンドラがアクティブになります。
- ヘッダーにエラー処理フラグが立てられ、正常に開始されたことを示します。
- ニュークリアスでエラー処理がアクティブであることを示す ADANI2 メッセージが生成されます。

- 2 どの出口が重要（デフォルト）であるかを判断し（デフォルト）、重要でないものに対して `SMGT XNOTCRITICAL=exit-code` オペレータコマンドを発行します。
- 3 特に、PINRSP または PINAUTOR がアクティブになる場合、必要に応じて ADASMXIT を修正します。出口を再アセンブルし、それが Adabas ロードライブラリにあるか、または起動時に有効なロードライブラリで利用可能であることを確認します。
- 4 どの PIN をアクティブにするかを決定します。

次の表は有効な PIN のリストおよびそれらをアクティブにする方法を示しています。

PIN ルーチン	インストール方法
PINAUTOR	Adabas ロードライブラリの NOAUTOR の名前を PINAUTOR に変更します。
PINOPRSP	Adabas ロードライブラリの NOOPRSP の名前を PINOPRSP に変更します。
PINRSP	ニュークリアスがアクティブなときに、オペレータコマンド <code>SMGT ADDPIN=PINRSP</code> を発行します（下記の注参照）。
PINUES	ニュークリアスがアクティブなときに、オペレータコマンド <code>SMGT ADDPIN=PINUES</code> を発行します（下記の注参照）。



Note: PINRSP および PINUES が処理するレスポンスコードの一部は共通しているため、制御を取得するモジュールの最後に ADDPIN 機能を実行します。例えば、PINRSP および PINUES の両方ともレスポンスコード 55 を処理します。PINUES が制御を取得する場合、PINRSP の後の PINUES で ADDPIN を実行する必要があります。

この時点で、エラー処理は、完全に操作可能であり、SMGT オペレータコマンドを発行できます。

10 Adabas ダンプフォーマットングツール (ADAFDP)

■ ADAFDP 機能	138
■ ADAFDP 出力	138

このセクションでは、Adabas ダンプフォーマットングツール ADAFDP の利用について説明します。

ADAFDP 機能

ADAFDP はアドレススペースダンプフォーマットングモジュールです。Adabas ニュークリアスが異常終了でシャットダウンするとき、このモジュールが制御を受け取り、エラーの原因分析に役立つ情報をフォーマットして表示します。

ニュークリアスがシャットダウンしているときに、ADAMPM はシャットダウンの原因を判断します。原因が異常終了の場合、ADAMPM は Adabas SVC への 20 コールの前に ADAFDP モジュールをアドレススペースにロードします。その後 ADAFDP は制御を受け取り、ニュークリアス情報をフォーマットします。

ADAFDP をロードできない場合、メッセージ ADAF03 がコンソールに表示され、異常終了シャットダウンを継続します。

ADAFDP 出力

ADAFDP がフォーマットした情報の大部分は、明白で、説明不要なものですが、情報のタイプと量はシャットダウン状況に依存するため、このセクションに ADAFDP 出力の概要を示します。

- ADAFDP メッセージ
- プールの省略形
- ユーザースレッド
- コマンド情報
- RABN 情報

ADAFDP メッセージ

メッセージ	説明
ADAH51/ADAH52	このメッセージは、フォーマットの開始時と終了時にコンソールに表示され、同時に DDPRINT に書き込まれます。
ADAMPMABENDCODE and PSW	アベンドコードとプログラム状況ワード (PSW) が Adabas ESTAE によって ADAMPM に保存された場合、ADAFDP はこれらのメッセージを表示します。さらに、ADAFDP はエントリポイントが PSW に最も適合するモジュールを特定し、そのモジュール内のオフセットを計算します。ADAMPM アベンドコードと PSW がゼロの場合、ADAFDP はこの情報をフォーマットしません。
ADABAS MODULE LOCATIONS	ADAFDP は、アドレススペース内の各 Adabas ニュークリアスモジュールの場所をフォーマットして表示します。

メッセージ	説明
ADDRESS LOCATIONS FOR USER EXITS	ADAFDP は Adabas ニュークリアスでロードされたすべてのユーザー出口の場所をフォーマットして表示します。
ADDRESS LOCATIONS FOR HYPEREXITS	ADAFDP は Adabas ニュークリアスでロードされたすべてのハイパー出口の場所をフォーマットして表示します。ハイパー出口 10~31 はそれぞれ A~U として表示されます。
ADANC0 STANDARD REGISTER SAVE AREA	レジスタ 0~7/8~F。これらは ADANC0 に保存されます。これらのレジスタのどれかにストレージ内のニュークリアスプールをポイントするアドレスが含まれているか、ADAFDP は判断します。含まれている場合、ADAFDP はどのプールかを示し、そのアドレスのストレージをスナップします。そのレジスタが 12 でユーザースレッドをポイントしている場合、ADAFDP はスレッド全体をスナップします。
ADANC0 ABEND SAVE REGISTERS	レジスタ 0~7/8~F。これらはユーザーアベンドの結果として、ADANC0 に保存されます。ADAFDP は、これらの保存済みレジスタのいずれかにストレージ内のニュークリアスプールをポイントするアドレスが含まれているかを判断します。含まれている場合、ADAFDP はどのプールかを示し、その場所のストレージをスナップします。その保存済みレジスタが 12 でユーザースレッドをポイントしている場合、ADAFDP はスレッド全体をスナップします。
ADAMPM SAVE REGISTERS	レジスタ 0~7/8~F。これらは Adabas ESTAE によって ADAMPM に保存されています。これらは ADAM99 メッセージで表示されるのレジスタと同じです。ADAFDP は、これらの保存済みレジスタのいずれかにストレージ内のニュークリアスプールをポイントするアドレスが含まれているかを判断します。含まれている場合、ADAFDP はどのプールかを示し、その場所のストレージをスナップします。
BEGIN / ENDING ADDRESSES OF POOLS / TABLES	ADAFDP は、Adabas ニュークリアスのプールとテーブルの開始/終了アドレス位置を判断します。これらのアドレスは、実際のダンプを簡単に検索できるように表示されます。詳細については、「 プールの省略形 」を参照してください。
ADABAS THREADS	ADAFDP はスレッド 0、-1、および -2 を含む物理スレッドをフォーマットします。行数は NT の値に依存します。異常終了のときアクティブだったスレッド（存在する場合）がポインタ --> でマークされます。
USER THREADS	実行するワークが割り当てられた -2 から NT のスレッドについて、ADAFDP はそのスレッドのステータスに関する情報をフォーマットして表示します。詳細については、「 ユーザースレッド 」を参照してください。
FOLLOWING COMMANDS WERE FOUND IN THE CMD QUEUE	ADAFDP はコマンドキューをスキャンし、キューで見つかったコマンドの情報をフォーマットします。詳細については、「 コマンド情報 」を参照してください。
POOL INTEGRITY CHECK	ADAFDP は Adabas ニュークリアスアドレススペース内のさまざまなプールの健全性をチェックします。そのプール内でエラーが検出されると、ADAFDP はエラーになったプールとエラータイプを示します。さらに、ADAFDP はエラーが検出された位置のストレージをスナップします。
FOLLOWING RABNS/FILES ACTIVE IN BUFFER POOL	ADAFDP は、アクティブまたは更新中だった RABN のバッファプールヘッダーをスキャンします。詳細については、「 RABN 情報 」を参照してください。

メッセージ	説明
ADAIOR REGS FOUND AT OFFSET X'080'	このオフセットの ADAIOR に保存されている、レジスタ 0~7/8~F が見つかりました。これらのレジスタ値のいずれかが Adabas プール内を指していることを ADAFDP が検出した場合、その場所のストレージをスナップします。
ADAIOR REGS FOUND AT OFFSET X'0C0'	このオフセットの ADAIOR に保存されている、レジスタ 0~7/8~F が見つかりました。これらのレジスタ値のいずれかが Adabas プール内を指していることを ADAFDP が検出した場合、その場所のストレージをスナップします。
ICCB POINTED FROM X'A0' IN IOR	ICCB アドレスが ADAIOR 内のこのオフセットをポイントしています。
ADAI22 ADAIOR TRACE TABLE	ADAIOR のトレーステーブルのフォーマット。ADAM99 メッセージのフォーマットと同じです。

プールの省略形

プールの省略形	説明
LOG	ログエリア
OPR	Adabas ニュークリアスオペレータコマンド処理エリア
CQ	コマンドキューのアドレス。ADAFDP により後でフォーマットされます。
ICQ	内部コマンドキュー
TT	スレッドテーブル
IA1	内部エリア 1
SFT	セッションファイルテーブル
FU	ファイル使用テーブル
FUP	ファイル更新テーブル
IOT	非同期バッファフラッシュのための I/O テーブル
PL2	非同期バッファフラッシュのための PLOG エリア
PET	ポストされた ET テーブル
TPT	Tpost
TPL	Tplatz
UQP	ユニークディスクリプタテーブル
UHQ	アッパーホールドキュー
HQ	ホールドキュー
UUQ	アッパーユーザーキュー
UQ	ユーザーキュー
FP	フォーマットプール
FHF	ファイル HILF エlement
PA	プロテクションエリア
TBI	ISN テーブル
TBQ	シーケンシャル検索テーブル

プールの省略形	説明
WK3	WORK パート 3 スペース割り当てテーブル
IA2	内部エリア 2
WK2	WORK パート 2 スペース割り当てテーブル
VOL	VOLSER テーブル
WIO	WORK ブロック I/O エリア
FST	フリースペーステーブルのワークエリア
UT	ユーザースレッド
WP	ワークプール
AW2	WORK ブロック非同期 I/O エリア
IOP	非同期バッファフラッシュ関連 I/O プール
IU2	バッファプールインポートانسヘッダーアッパー 2
IU1	バッファプールインポートانسヘッダーアッパー 1
BU2	バッファプールアッパーヘッダー 2
BU1	バッファプールアッパーヘッダー 1
BH	バッファプールヘッダーのアドレス位置。バッファプールヘッダーの情報は ADAFDP により後でフォーマットされます。
BP	バッファプールの物理開始アドレス位置

ユーザースレッド

情報	説明
Thread Number	-2~NT
Status	スレッドの現在のステータスを示します。 <ul style="list-style-type: none"> ■ *Active* : 現在アクティブなスレッド ■ In Use : WORK を割り当てられているスレッド ■ Waiting For I/O : バッファプールにないブロックを待機中 ■ Waiting For RABN : 使用中の RABN を待機中 ■ Waiting For Work-2 Area Block : Waiting For I/O と同様 ■ Waiting Workpool Space : バイト数を 10 進数で表示 ■ Ready To Run : 実行の選択を待機中
CMD	実行されている Adabas コマンド
Response Code	レスポンスコード (存在する場合)
File Number	このコマンドのファイル番号
ISN	このコマンドの内部シーケンス番号
Sub. Rsp	サブルーチンレスポンスコード (存在する場合)

情報	説明
Last RABN for I/O	コマンド処理によって要求された最後の RABN (10 進)
Type	最後の RABN タイプ (A : ASSO、D : DATA)
CQE Addr	このコマンドのコマンドキューエレメントアドレス
User Jobname	このコマンドを実行したユーザーのジョブ名
ITID	このコマンドを実行したユーザーの内部 Adabas ID
User	このコマンドを実行したユーザーのユーザー ID
Unique global ID	このコマンドを所有するユーザーの 28 バイトの ID
Buffer Addresses	コントロールブロック、フォーマットバッファ、サーチバッファ、バリュースバッファ、ISN バッファのバッファアドレス
Buffer Lengths	FL : フォーマットバッファ長 RL : レコードバッファ長 SL : サーチバッファ長 VL : バリュースバッファ長 IL : ISN バッファ長
Snap Thread	ユーザースレッドの最初の 144 バイトがスナップされます。

コマンド情報

情報	説明
CQE Address	この CQE のアドレス位置
F	<p>コマンドキューフラグバイト。</p> <ul style="list-style-type: none"> ■ 1 バイト目：一般的な目的のフラグ <ul style="list-style-type: none"> ■ X'80' : ユーザーバッファがサービスパーティション、リージョン、アドレススペースにある ■ X'40' : ET コマンドが 12 コール待ち ■ X'20' : 16 コール待ち ■ X'10' : 16 コールが必要 ■ X'08' : アタッチドバッファ ■ X'04' : アタッチドバッファが必要 ■ X'02' : X メモリロックあり (MVS のみ) ■ 2 バイト目：選択フラグ <ul style="list-style-type: none"> ■ X'80' : 処理中 ■ X'40' : 選択待ち ■ X'20' : UQE の検索完了 ■ X'10' : UQE が見つかった ■ X'08' : BSS=x'80' ステータスのときは選択不可 ■ X'04' : ET-SYNC 中は選択不可

情報	説明
	<ul style="list-style-type: none"> ■ X'02' : スペース待ち ■ X'01' : HQ 内の ISN 待ち
CMD	コマンドタイプ
ファイル番号	このコマンドのファイル番号
ジョブ名	ユーザーのジョブ名
Addr User	検索して見つかった場合、ユーザーの UQE の UQE アドレス
Addr User ASCB	ユーザーの ASCB のアドレス位置
Addr ECB	ユーザーの ECB のアドレス位置 (ユーザーのアドレススペース内)
Addr User UB	ユーザーの UB のアドレス (ユーザーのアドレススペース内)
Addr User PAL	ユーザーのパラメータアドレスリストのアドレス位置
CQE ACA	CQE の ACA フィールド
CQE RQST	CQE の RQST フィールド
Abuf/Pal	CMD に対するアタッチドバッファ/パラメータアドレスリスト (PAL) のアドレス
Comm Id	このコマンドに対する 28 バイトのユニークなユーザー ID

RABN 情報

情報	説明
RABN Number	RABN 番号 (10進)
タイプ	ブロックタイプ (A : ASSO、D : DATA)
Flag	BP ヘッダーエレメントフラグバイト : <ul style="list-style-type: none"> ■ AKZ X'40' : アクティブインジケータ ■ UKZ X'20' : 更新インジケータ ■ RKZ X'10' : 読み込みインジケータ ■ XKZ X'04' : アクセスはブロック待ち ■ YKZ X'02' : 更新はブロック待ち ■ SKZ X'01' : 書き込みインジケータ
ファイル	このブロックの所有者のファイル番号
Address	ストレージ内のブロックのアドレス位置

11 変換テーブル

- Adabas による EBCDIC から ASCII および ASCII から EBCDIC の変換 146
- Entire Net-Work による EBCDIC から ASCII および ASCII から EBCDIC の変換 147

このセクションでは、Adabas で提供される変換テーブルについて説明します。

Adabas による EBCDIC から ASCII および ASCII から EBCDIC の変換

```
cUES2ASC DS 0F
c* .0.1.2.3.4.5.6.7.8.9.A.B.C.D.E.F
c DC x'000102033F093F7F3F3F3F0B0C0D0E0F' 0.
c DC x'101112133F3F083F18193F3F3F1D3F1F' 1.
c DC x'3F3F1C3F3F0A171B3F3F3F3F050607' 2.
c DC x'3F3F163F3F1E3F043F3F3F3F14153F1A' 3.
c DC x'203F3F3F3F3F3F3F3F3F2E3C282B3F' 4.
c DC x'263F3F3F3F3F3F3F3F3F21242A293B5E' 5.
c DC x'2D2F3F3F3F3F3F3F3F3F7C2C255F3E3F' 6.
c DC x'3F3F3F3F3F3F3F3F3F603A2340273D22' 7.
c DC x'3F6162636465666768693F3F3F3F3F' 8.
c DC x'3FA6B6C6D6E6F7071723F3F3F3F3F' 9.
c DC x'3F7E737475767778797A3F3F3F5B3F3F' A.
c DC x'3F3F3F3F3F3F3F3F3F3F3F3F5D3F3F' B.
c DC x'7B4142434445464748493F3F3F3F3F' C.
c DC x'7D4A4B4C4D4E4F5051523F3F3F3F3F' D.
c DC x'5C3F535455565758595A3F3F3F3F3F' E.
c DC x'303132333435363738393F3F3F3F3F' F.
c* .0.1.2.3.4.5.6.7.8.9.A.B.C.D.E.F
END
cUES2EBC DS 0F
c* .0.1.2.3.4.5.6.7.8.9.A.B.C.D.E.F
c DC x'00010203372D2E2F1605250B0C0D0E0F' 0.
c DC x'101112133C3D322618193F27221D351F' 1.
c DC x'405A7F7B5B6C507D4D5D5C4E6B604B61' 2.
c DC x'F0F1F2F3F4F5F6F7F8F97A5E4C7E6E6F' 3.
c DC x'7CC1C2C3C4C5C6C7C8C9D1D2D3D4D5D6' 4.
c DC x'D7D8D9E2E3E4E5E6E7E8E9ADE0BD5F6D' 5.
c DC x'79818283848586878889919293949596' 6.
c DC x'979899A2A3A4A5A6A7A8A9C06AD0A107' 7.
c DC x'6F6F6F6F6F6F6F6F6F6F6F6F6F6F6F' 8.
c DC x'6F6F6F6F6F6F6F6F6F6F6F6F6F6F6F' 9.
c DC x'6F6F6F6F6F6F6F6F6F6F6F6F6F6F6F' A.
c DC x'6F6F6F6F6F6F6F6F6F6F6F6F6F6F6F' B.
c DC x'6F6F6F6F6F6F6F6F6F6F6F6F6F6F6F' C.
c DC x'6F6F6F6F6F6F6F6F6F6F6F6F6F6F6F' D.
c DC x'6F6F6F6F6F6F6F6F6F6F6F6F6F6F6F' E.
c DC x'6F6F6F6F6F6F6F6F6F6F6F6F6F6F6F' F.
c* .0.1.2.3.4.5.6.7.8.9.A.B.C.D.E.F
END
```

Entire Net-Work による EBCDIC から ASCII および ASCII から EBCDIC の変換

```
NW2ASC DS 0F
* .0.1.2.3.4.5.6.7.8.9.A.B.C.D.E.F
DC X'000102030405060708090A0B0C0D0E0F' 0.
DC X'101112131415161718191A1B1C1D1E1F' 1.
DC X'00000000000000000000000000000000' 2.
DC X'00000000000000000000000000000000' 3.
DC X'20000000000000000000000005B2E3C282B5D' 4.
DC X'260000000000000000000000021242A293B5E' 5.
DC X'2D2F000000000000000000007C2C255F3E3F' 6.
DC X'000000000000000000000000603A2340273D22' 7.
DC X'0061626364656667686900000000000000' 8.
DC X'006A6B6C6D6E6F70717200000000000000' 9.
DC X'007E737475767778797A00005B000000' A.
DC X'000000000000000000000000000005D0000' B.
DC X'7B41424344454647484900000000000000' C.
DC X'7D4A4B4C4D4E4F50515200000000000000' D.
DC X'5C7E535455565758595A00000000000000' E.
DC X'303132333435363738397C00000000FF' F.
* .0.1.2.3.4.5.6.7.8.9.A.B.C.D.E.F
NW2EBC DS 0F
* .0.1.2.3.4.5.6.7.8.9.A.B.C.D.E.F
DC X'000102030405060708090A0B0C0D0E0F' 0.
DC X'101112131415161718191A1B1C1D1E1F' 1.
DC X'405A7F7B5B6C507D4D5D5C4E6B604B61' 2.
DC X'F0F1F2F3F4F5F6F7F8F97A5E4C7E6E6F' 3.
DC X'7CC1C2C3C4C5C6C7C8C9D1D2D3D4D5D6' 4.
DC X'D7D8D9E2E3E4E5E6E7E8E9ADE0BD5F6D' 5.
DC X'79818283848586878889919293949596' 6.
DC X'979899A2A3A4A5A6A7A8A9C06AD0A100' 7.
DC X'00000000000000000000000000000000' 8.
DC X'00000000000000000000000000000000' 9.
DC X'00000000000000000000000000000000' A.
DC X'00000000000000000000000000000000' B.
DC X'00000000000000000000000000000000' C.
DC X'00000000000000000000000000000000' D.
DC X'00000000000000000000000000000000' E.
DC X'000000000000000000000000000000FF' F.
* .0.1.2.3.4.5.6.7.8.9.A.B.C.D.E.F
END
```

索引

A

Adabas Bridge for DL/I のサポート, 95
Adabas Bridge for VSAM のサポート, 96
Adabas CICS タスク関連ユーザー出口 (TRUE)
モジュール名, 105
Adabas Review のサポート, 102
Adabas Security インターフェイスパラメータ, 103
Adabas SVC 番号パラメータ, 104
Adabas Transaction Manager と Adabas Fastpath 出口のサポート, 96
Adabas コールのルーティング, 85
ADL パラメータ, 95
AVB パラメータ, 96

B

BS2000 IDT 共通メモリプール名, 98
BS2000 メモリプールのユーザー連結設定, 98

C

CICS XWAIT 設定, 106
CICS コマンドレベルリンクルーチン名, 97
CICS トランザクションの削除パラメータ, 102
CICS マルチプルリージョンオプション, 99
CICS ユーザー ID の作成方法, 100
CICS リソースマネージャインターフェイスのパラメータ, 103
CICS リンクルーチンに作成されるユーザーブロック, 101
CITSNM パラメータ, 96
COR パラメータ, 96
CSECT または DSECT の生成, 98

D

DBID パラメータ, 91
DBID/SVC ルーティングテーブル, 96, 97
ソースコード, 89
DBID2 パラメータ, 92
DBSVCTN パラメータ, 96
DSECT データ接頭辞のパラメータ, 102
DYNDBSVC パラメータ, 97

E

ENTPT パラメータ, 97

G

GBLNAME パラメータ, 97
GEN パラメータ, 98

I

IDTNAME パラメータ, 98
IDTUGRP パラメータ, 98

L

LGBLSET マクロ
ADL パラメータ, 95
AVB パラメータ, 96
CITSNM パラメータ, 96
COR パラメータ, 96
DBSVCTN パラメータ, 96
DYNDBSVC パラメータ, 97
ENTPT パラメータ, 97
GBLNAME パラメータ, 97
GEN パラメータ, 98
IDTNAME パラメータ, 98
IDTUGRP パラメータ, 98
LOGID パラメータ, 98
LUEXIT1A パラメータ, 98
LUEXIT2A パラメータ, 98
LUIINFO パラメータ, 99
LUSAVE パラメータ, 99
LX1NAME パラメータ, 99
LX2NAME パラメータ, 99
MRO パラメータ, 99
NETOPT パラメータ, 100
NTGPID パラメータ, 100
NUBS パラメータ, 101
OPSYS パラメータ, 101
PARMTYP パラメータ, 101
PRE パラメータ, 102
PURGE パラメータ, 102
RENT パラメータ, 102
RETRYX パラメータ, 102
REVIEW パラメータ, 102
RMI パラメータ, 103
RTXNAME パラメータ, 103
SAF パラメータ, 103
SAP パラメータ, 103
SAPSTR パラメータ, 104
SVCNO パラメータ, 104

TPMON パラメータ, 104
TRUENM パラメータ, 105
UBPLOC パラメータ, 105
UES パラメータ, 105
USERX1 パラメータ, 105
USERX2 パラメータ, 105
XWAIT パラメータ, 106
変更点, 94
LOGID パラメータ, 98
LU 出口 1 ワークエリアの長さ, 98
LU 出口 2 ワークエリアの長さ, 98
LUEXIT1 および LUEXIT2 のユーザーセーブエリア, 99
LUEXIT1A パラメータ, 98
LUEXIT2A パラメータ, 98
LUIINFO パラメータ, 99
LUSAVE パラメータ, 99
LX1NAME パラメータ, 99
LX2NAME パラメータ, 99

M

MDBSVC マクロ
TYPE=FINAL ステートメントの構文, 91
TYPE=GEN ステートメントの構文, 91
TYPE=INIT ステートメントの構文, 91
使用, 89
ステートメントタイプ, 90
パラメータ, 91
MRO パラメータ, 99

N

Natural グループ ID, 100
NETOPT パラメータ, 100
NTGPID パラメータ, 100
NUBS パラメータ, 101

O

OPSYS パラメータ, 93, 101

P

PARMTYP パラメータ, 101
PRE パラメータ, 102
PREFIX パラメータ, 93
PURGE パラメータ, 102

R

RENT パラメータ, 102
RETRYX パラメータ, 102
REVIEW パラメータ, 102
RMI パラメータ, 103
RTXNAME パラメータ, 103

S

SAF パラメータ, 103
SAP ID 文字列パラメータ, 104
SAP パラメータ, 103
SAP ユーザー ID 生成サポートパラメータ, 103
SAPSTR パラメータ, 104

SVC パラメータ, 93
SVC ルーティング
データベース ID による, 85
SVCNO パラメータ, 104

T

TABNAME パラメータ, 94
TP オペレーティング環境パラメータ, 104
TPMON パラメータ, 104
TRUENM パラメータ, 105
TYPE=DSECT ステートメント
MDBSVC マクロ, 90
TYPE=FINAL ステートメント
MDBSVC マクロ, 90
構文, 91
TYPE=GEN ステートメント
MDBSVC マクロ, 90
構文, 91
TYPE=INIT ステートメント
MDBSVC マクロ, 90
構文, 91

U

UBPLOC パラメータ, 105
UES パラメータ, 105
USERX1 パラメータ, 105
USERX2 パラメータ, 105

X

XWAIT パラメータ, 106

Z

z/OS 環境での Adabas のインストール, 5

お

オペレーティングシステムパラメータ, 101

こ

コマンド再試行出口名, 103

さ

再試行コマンド出口フラグ, 102

た

ターゲットデータベース ID
デフォルト, 98

て

デフォルトのターゲットデータベース ID, 98

は

パラメータリストエリア, 101

ゆ

- ユニバーサルエンコーディングサポートパラメータ, 105
- ユーザー出口 1 のモジュール名, 99
- ユーザー出口 1 フラグ, 105
- ユーザー出口 2 のモジュール名, 99
- ユーザー出口 2 フラグ, 105
- ユーザー出口 4
 - 渡されたユーザーデータの長さ, 99
- ユーザー出口 4 に渡されたユーザーデータの長さ, 99
- ユーザーブロックプールの割り当てパラメータ, 105

り

- リエントラントグローバルモジュールフラグ, 102
- リンクグローバルモジュール名, 97

