Adabas for Linux, UNIX and Windows

Adabas ユーティリティ

バージョン 6.6

2017 年 10 月

ADABAS & NATURAL

このマニュアルは Adabas for Linux, UNIX and Windows バージョン 6.6 およびそれ以降のすべてのリリースに適用されます。

このマニュアルに記載される仕様は変更される可能性があります。変更は以降のリリースノートまたは新しいマニュアルに記述されます。

ドキュメント IDは: ADAOS-AADAOSUTILITIES-66-20200619JA

目次

Adabas ユーティリティ

このドキュメントでは、Adabas ユーティリティについて説明します。データベース管理者（DBA）は、Adabas ユーティリティを使用して、Adabas データベースを作成および保守します。各ユーティリティについて、次の情報が記述されています。

- ユーティリティの目的
- ユーティリティの機能概要
- ユーティリティの制御パラメータ
- ユーティリティの使用例（図解）

このドキュメントは、おもに DBA 向けに作成されています。一部の Adabas ユーティリティには、既存のデータベース情報を修正、削除するための機能があるので、注意する必要があります。ADAREP などのユーティリティは、ステータス情報のみを提供し、エンドユーザーが自由に使用できます。

注意: Adabas ユーティリティには、文書化されていない構文を使用して呼び出すことができる、文書化されていない機能も含まれています（これには、『*管理マニュアル*』で説明されている FDT 構文も含まれます）。Software AG では、このような文書化されていない機能を使用しないことを強く推奨しています。文書化されていない機能は、正しく動作する保証も、Adabas の一般的な動作に悪影響を及ぼさないという保証もありません。

「*概要*」では、利用できるユーティリティの要約とその目的について説明します。

次のドキュメントでは、ドキュメントごとにユーティリティを 1 つずつ詳しく説明します。

付録 A では、Adabas キットに収録されているデモユーティリティ入力ファイルについて説明します。

付録 B では、サンプルプログラム prilogc について説明します。このサンプルプログラムは、ニュークリアスパラメータ CLOGLAYOUT を 6 に設定した状態で作成されたコマンドログを出力するために使用します。

付録 C には、Adabas ユーティリティによって書き込まれるチェックポイントに関する情報が含まれています。

1 表記規則

このドキュメントは、次の規則に従って記載されています。

文字フォントの使用

他のマニュアルを参照するときは*イタリック体*で示します。

ドキュメントまたはドキュメントのセクションを参照するときは太字で示します。

ユーティリティの出力例とファイルの内容は、次のようなフォントで示します。

```
%ADADBM-I-OPENED,  ds DATA2, file DATA2.001 opened
%ADADBM-F-DSSTALL, allocation error DSST
```

ユーザー入力とユーティリティ出力の両方を示す例は、次のようなフォントで示します。

```
adadbm: add_container = data, size = 35
%ADADBM-I-OPENED,  ds DATA2, file DATA2.001 opened
%ADADBM-F-DSSTALL, allocation error DSST
```

構文形式

ユーティリティ制御パラメータの構文は次のとおりです。

大文字の項目はキーワードです。表示どおりに入力する必要があります。キーワードは大文字、小文字のどちらでも入力できます。

小文字の項目はユーザーが選択した値に変換する必要があります。項目が "number" の場合、10 進数を指定します。正の数か 0 だけを指定できます。負の数は指定できません。項目が "string" の場合、テキスト文字列、つまり英数字の文字をいくつでも指定できます。数字または文字列の場合は、"0x"、"0X" で始まる 16 進値を指定することもできます。"^x" または "^X"。16 進数で指定された数字の場合、先頭の 0 は省略できます。他の項目が出現することもあります。例えば "descriptor" とあれば、ディスクリプタ名を入力する必要があります。

角カッコ（[ ]）で囲まれた項目はオプションです。

中カッコ（{ }）で囲まれた項目は必須です。

垂線（|）は複数の項目を区切ります。項目のどれか 1 つだけを指定する必要があります。

省略記号（...）は、省略記号の直前の構文要素を繰り返し記述できることを示します。

コンマの後の省略記号（,...）は、省略記号の直前の構文要素を、コンマで区切って繰り返し記述できることを示します。

要素のリストに丸カッコを使用してキーワードを1つだけ指定する場合、指定する要素が1つだけであれば丸カッコを省略できます。

例 **1**

```
RB=0x33445566
```

ここでは、文字列の値が16進数の文字列として指定されています。これは、ASCII文字列"3DUf"と同等の出力可能な文字で構成されています。

例 **2**

```
DBID = number
```

この例では、キーワード "DBID" を大文字、小文字、またはその組み合わせで入力し、= 文字と 10 進数が後に続いています。例えば、次のように入力します。

```
DBID = 27
```

例 **3**

```
RABN = number [ - number]
```

この例では、キーワード "RABN" を大文字、小文字、またはその組み合わせで入力し、= 文字と 10 進数が後に続いています。ハイフン（-）とそれに続けて 10 進数を追加できますが、これは必須ではありません。この構文に当てはまる入力例を次に示します。

```
RABN = 25
RABN = 1000 - 1125
```

例 **4**

```
RABN = 0x400
```

この場合、値は 16 進数値として指定されています。これは以下の指定と同等です。

```
RABN = 1024
```

例 **5**

```
SORTSEQ = { descriptor | ISN }
```

この例では、キーワード "SORTSEQ" を大文字、小文字、またはその組み合わせで入力し、次に = 文字、ディスクリプタ値かキーワード "ISN" のどちらか一方が続いています。

```
SORTSEQ = ISN
```

例 **6**

```
number[-number] [,number[-number] ] ...
```

この例は、省略記号（...）の使用例です。省略記号は構文要素の "[,number[-number]]" の後に続きます。つまり、この構文要素を入力行に何回でも繰り返すことができます。この構文に当てはまる入力例を次に示します。

```
27
27-50
27-50,68
27,68-90
27-50,68-90
27-50,68-90,102,105,118-140,160
```

例 **7**

例 **4** の代わりとして、（,...）を使用して次の構文も指定できます。

```
{ number[-number] },...
```

この構文によって、例 **4** のすべての組み合わせが可能です。

例 **8**

```
(number[,number]...)
```

この構文に有効な入力例は次のとおりです。

```
(12,23,45)
(123)
123 - only one list element, so brackets can be omitted
```

大文字変換

指定されたパラメータ名は常に大文字に変換されます。

ほとんどのユーティリティ制御パラメータでは、指定されたパラメータ値も大文字に変換されますが、これは必ずしも望ましい動作ではありません。Adabas バージョン 6.1.6 以降では、制御パラメータ値の大文字変換の新しい規則が導入されました。

- 「*parameter=value*」を指定すると、値は大文字に変換されます。
- 「*parameter:value*」を指定すると、値は大文字に変換されません。

パラメータ名の後にコロンまたは等号を指定するオプションは、通常、パーサーを通じてすべてのパラメータに対して指定されますが、Software AG では、構文で明示的に記述されているパラメータに対してのみコロンを指定することを推奨しています。これは、明示的に記述されているパラメータに対してのみ、上記の動作が保証されていることが理由です。つまり、以前の Adabas

バージョンとの互換性の理由により、その他のパラメータの一部では大文字変換の処理が異なっています。

例

次の構文を想定します。

```
NAME{=|:}string
```

次のように指定した場合

```
NAME=Production
```

NAME のパラメータ値は "PRODUCTION" に設定されます。

次のように指定した場合

```
NAME:Production
```

NAME のパラメータ値は "Production" に設定されます。

ただし、フィールド、ディスクリプタ、参照制約の定義など、一部のユーティリティ入力は "パラメータ{=|:}値" として提供されません。定義の前にパラメータ LOWER_CASE_FIELD_NAMES が指定されていない限り、指定はデフォルトで大文字に変換されます。

注意: LOWER_CASE_FIELD_NAMES を指定すると、いずれの定義も大文字に変換されません。この場合は、フィールドオプションなどのキーワードを大文字で指定する必要があります。

例

LOWER_CASE_FIELD_NAMES を指定しない場合、フィールド定義

```
1,aa,8,a,de
```

は正しく、次と同等です。

```
1,AA,8,A,DE
```

ただし、LOWER_CASE_FIELD_NAMES を指定した場合、このフィールド定義は無効です。フィールド "aa" を定義するには、次を指定する必要があります。

```
1,aa,8,A,DE
```

フィールド指定

いくつかのユーティリティには、Adabas フィールドを指定できるパラメータが含まれています。フィールド指定の正確な構文は、対象とするパラメータによって異なります。例えば、ADAINV INVERT の場合は、フィールド名の指定で十分ですが、ADADBM ADD_FIELD の場合は完全なフィールド定義が必要です。

これらのフィールド指定は、END_OF_FIELDS パラメータで終了できます。

注意:

1. LOWER_CASE_FIELD_NAMES を指定した後では、END_OF_FIELDS パラメータを大文字で指定する必要があります。
2. ADACMP および ADADCU では、END_OF_FIELDS の代わりにピリオド（.）を指定できます。

ユーティリティのパラメータ指定の最後では、END_OF_FIELDS を省略できます。フィールド指定の後に他のパラメータを追加する場合は、END_OF_FIELDS が必要です。

注意: V6.3 SP3 より前の Adabas バージョンでは、一部のユーティリティで、パラメータ指定の最後に END_OF_FIELDS が必要でした。END_OF_FIELDS が見つからないと、機能は実行されませんでした。

制御パラメータに使用するシンボルの要約

各ユーティリティの説明の中に、そのユーティリティで使用できる制御パラメータの構文をまとめた表が記載されています。一部の制御パラメータの先頭には、文字 M または D が付いています。

文字 M は必須であることを示します。つまり、このパラメータをユーティリティの入力に指定しないとユーティリティは実行できなくなります。文字 M がない場合、そのパラメータは任意指定なので、指定しなくても構いません。

文字 D は、制御パラメータにデフォルト値があることを示します。つまり、このパラメータを明示的に指定しないと、パラメータに事前に設定されている値がそのユーティリティで使用されます。

パラメータの順序

ユーティリティのパラメータは、アルファベットの順にドキュメントに示されます。しかし、パラメータによっては、パラメータの指定順序に制限があります。通常、DBIDパラメータは最初に指定します。また、ユーティリティによっては多くの制限があります。

特に多機能ユーティリティの場合、機能を直ちにトリガするパラメータがいくつかあります。このようなパラメータの後に指定された他のパラメータは無視されます。ただし、後で別の機能を指定した場合は、これらのパラメータが使用されます。

例：

```
adavfy dbid=34 file=9 index file=11 data
```

パラメータINDEXによりインデックス検証がトリガされます。このパラメータには、先行する"file=9" の指定が影響します。DATA パラメータによりデータ検証がトリガされます。このパラメータには、先行する"file=11"の指定が影響します。このコマンドでは、ファイル9のインデックス検証とファイル 11 のデータ検証がトリガされます。

数値

数値は次の方法で指定できます。

- 番号
- number[K]（実際の値は指定した数値の 1024 倍）
- number[M]（実際の値は指定した数値の 1024*1024 倍）

number[K] および number[M] は、大きな数値を指定する場合のみ使用できます。

最大値

Adabasの制限により固定的な限界値がある場合にだけ、数値パラメータの最大値について説明します。符号ありまたは符号なしの4バイト整数が変数の格納に使用されるときの制限値については、説明しません。この場合、制限値が最大の可能な整数よりわずかに小さくなることがあるからです（例えば 4000 M）。

HTML ドキュメントでの構文図

各ユーティリティの最初に構文図があります。それらの構文図には、利用可能なキーワードとパラメータの詳細な記述へのリンクを含んでいます。構文図のハイパーリンクには、強調のために下線が引かれていますが、下線は構文の一部ではありませんので注意してください。

処理フロー

ユーティリティのドキュメントには、ユーティリティがアクセスするファイルを示す処理フロー図が含まれています。ユーティリティは、データベースコンテナにアクセスする際に、コンテナの場所を確認するために、*ADABAS.INI* および *DBnnn.INI* ファイルも読み取ります。シンプルな図を維持するため、*ADABAS.INIDBnnn.INI* は、修正された場合にのみ記述されています。

廃止されたパラメータ

場合によっては、Adabasの新しいバージョンがリリースされたときに、ユーティリティやニュークリアスのパラメータが廃止されることがあります。通常、ユーティリティまたはニュークリアスは、廃止されたパラメータを以前と同様に受け入れますが、次のような PAROBS 警告が表示されます。

```
%ADANUC-W-PAROBS, parameter NH has become obsolete
```

2

表記規則

規則	説明
太字	画面上の要素を表します。
モノスペースフォント	*folder.subfolder:service* という規則を使用して webMethods Integration Server 上のサービスの保存場所を表します。
大文字	キーボードのキーを表します。同時に押す必要があるキーは、プラス記号 (+) で結んで表記されます。
斜体	独自の状況または環境に固有の値を指定する必要がある変数を表します。本文で最初に出現する新しい用語を表します。
モノスペースフォント	入力する必要があるテキストまたはシステムから表示されるメッセージを表します。Program code.
{ }	選択肢のセットを表します。ここから1つ選択する必要があります。中カッコの内側にある情報のみを入力します。{ } 記号は入力しません。
|	構文行で相互排他的な2つの選択肢を区切ります。いずれかの選択肢を入力します。| 記号は入力しません。
[ ]	1 つ以上のオプションを表します。大カッコの内側にある情報のみを入力します。[ ] 記号は入力しません。
...	同じ種類の情報を複数回入力できることを示します。情報だけを入力してください。実際のコードに繰り返し記号 (...) を入力しないでください。

オンライン情報

Software AG マニュアルの **Web** サイト

マニュアルは、Software AG マニュアルの Web サイト (**http://documentation.softwareag.com**) で入手できます。このサイトでは Empower クレデンシャルが必要です。Empower クレデンシャルがない場合は、TECHcommunity Web サイトを使用する必要があります。

Software AG Empower 製品のサポート **Web** サイト

もしまだEmpowerのアカウントをお持ちでないのなら、こちらへempower@softwareag.com電子メールにて あなたのお名前、会社名、会社の電子メールアドレスをお書きの上、アカウントを請求してください。

いったんアカウントをお持ちになれば、Empower **https://empower.softwareag.com/** のeServiceセクションにてサポートインシデントをオンラインで開くことができます。

製品情報は、Software AG Empower 製品のサポート Web サイト (**https://empower.softwareag.com**) で入手できます。

機能および拡張機能に関するリクエストの送信、製品の可用性に関する情報の取得、製品のダウンロードを実行するには、Products に移動します。

修正に関する情報を取得し、早期警告、技術論文、Knowledge Base の記事を読むには、**Knowledge Center** に移動します。

もしご質問があれは、こちらの**https://empower.softwareag.com/public_directory.asp** グローバルサポート連絡一覧の、あなたの国の電話番号を選んで、わたくし共へご連絡ください。

Software AG TECHcommunity

マニュアルおよびその他の技術情報は、Software AG TECHcommunity Web サイト (**http://techcommunity.softwareag.com**) で入手できます。以下の操作を実行できます。

- TECHcommunity クレデンシャルを持っている場合は、製品マニュアルにアクセスできます。TECHcommunity クレデンシャルがない場合は、登録し、関心事の領域として [マニュアル] を指定する必要があります。
- 記事、コードサンプル、デモ、チュートリアルにアクセスする。
- Software AGの専門家によって承認されたオンライン掲示板フォーラムを使用して、質問したり、ベストプラクティスを話し合ったり、他の顧客が Software AG のテクノロジをどのように使用しているかを学んだりすることが可能です。
- オープンスタンダードや Web テクノロジを取り扱う外部 Web サイトにリンクできます。

データ保護

Software AG製品は、EU一般データ保護規則(GDPR)を尊重した個人データの処理機能を提供します。 該当する場合、適切な手順がそれぞれの管理ドキュメントに記載されています。

3 概要

この章では、Adabas データベースを管理するための機能である Adabas ユーティリティ全般について説明します。

ADABAS
データベースニュークリアスの開始（Windows のみ）

このユーティリティは、目的の環境でデータベースニュークリアスを開始するときに使用します。

ADABCK
データベースまたはファイルのバックアップとリストア

Adabasバックアップユーティリティは、データベース（または特定ファイル）の内容をシーケンシャルデータファイルにダンプするときに使用します。また、ダンプした内容をそのシーケンシャルデータファイルからリストアするときにも使用します。このユーティリティは、Adabas バックアップコピーをコピーするときにも使用できます。

ADACLP
コマンドログレポート

このユーティリティは、コマンドログの出力に使用します。

ADACMP
データの圧縮

圧縮ユーティリティは、ユーザーデータの圧縮に使用します。圧縮後のデータは、一括更新ユーティリティ ADAMUP の入力として使用できます。このユーティリティで想定している入力は、未加工データと、そのデータの構造を記述したデータ定義の組み合わせです。

ADACVT
データベースを以前のバージョンから変換したり、以前のバージョンへと変換したりします。

ADADBM
データベース更新

ADADBM ユーティリティには、データベース更新用に次の機能が備わっています。

- ADD_CONTAINER 機能は、新規コンテナファイルをアソシエータまたはデータストレージデータセットに追加します。
- ADD_FIELDS 機能は、ファイルの FDT の末尾に新しいフィールドを追加します。
- ALLOCATE 機能は、ファイルに割り当てられるノーマルインデックス、アッパーインデックス、アドレスコンバータ、またはデータストレージのスペースを増やします。DEALLOCATE 機能は逆の働きをする機能です。
- CHANGE 機能は、FDT 内のフィールドの標準長を変更します。
- CHANGE_FIELDS 機能は、フィールド定義を変更します。
- DEFINE_REFINT 機能は、新しい参照制約を定義します。
- DELCP 機能は、日付範囲を指定してチェックポイントファイルから古いチェックポイントレコードを削除します。
- DELETE 機能は、特定の Adabas ファイルまたはファイル範囲をデータベースから削除します。
- DELETE_DATABASE 機能は、データベースを削除します。指定されたキーワードに応じて、コンテナのみが削除されるか、データベースディレクトリとそのコンテンツが削除されます。
- DISPLAY 機能は、UCB を表示します。
- DROP_FIELDS 機能は、指定フィールドを存在しないものとしてマークします。つまり、これらのフィールドにはアクセスできなくなります。
- DROP_LOBFILE 機能は、ADAFDU ADD_LOBFILE の逆の機能です。
- DROP_REFINT 機能は、既存の参照制約をドロップします。
- EXTEND_CONTAINER 機能は、データベースに定義された最後のコンテナファイルを拡張します。
- NEW_DBID 機能は、使用しているデータベースの ID を変更します。
- NEWWORK 機能は、新規の Adabas WORK データセットを割り当てし、フォーマットします。
- PGM_REFRESH 機能は、アプリケーションプログラムの E1 コマンドで Adabas ファイルのリフレッシュの有効／無効を切り替えるために使用します。
- RECOVER 機能は失われたスペースを FST に返します。
- REDUCE_CONTAINER 機能は、データベースに定義された最後のコンテナファイルのサイズを縮小します。
- REFRESH 機能は、ファイルまたはファイル範囲をロードレコード数ゼロの状態にリセットします。
- REMOVE_CONTAINER 機能は、コンテナファイルを削除します。

- REMOVE_DROP 機能（後続の REFRESH と連携して使用される）は、ドロップされたフィールドを FDT から削除します。
- REMOVE_REPLICATION 機能は、すべてのレプリケーション処理を停止し、レプリケーションシステムファイルを削除します。
- RENAME 機能は、データベース名またはロードされたファイルの名前を変更します。
- RENUMBER 機能は、ロードされるファイルの番号を振り直すか、またはロードされるファイル同士の番号を交換します。
- REPLICATION_FILES 機能は、Adabas-Adabas レプリケーションに必要なシステムファイルを作成します。
- RESET 機能は、UCB からエントリを消去します。
- RESET_REPLICATION_TARGET 機能は、Adabas ファイルのレプリケーションターゲットフラグをリセットします。
- REUSE 機能は、Adabas によるデータストレージスペースまたは ISN の再利用を制御します。
- SECURITY 機能は、データベースのセキュリティモードを設定します。
- SYFMAX 機能は、指定されたファイル内のシステム生成マルチプルバリューフィールドに対して生成される値の最大数を指定します。

ADADCU
データの圧縮解除

ADADCU ユーティリティは、Adabas 以外のアプリケーションプログラムで使用するため、または圧縮ユーティリティ ADACMP への入力に使用するためにレコードの圧縮を解除します。圧縮解除するファイルは、このユーティリティへの入力に使用する前に、データベースからアンロードされている必要があります（アンロードユーティリティ ADAULD）。ADADCU では、レコード全体の圧縮解除、レコード内のフィールドの再配置、デフォルト長の変更、一部のタイプのフィールドの切り捨て、フォーマットの変更、新規フィールドを追加するためのスペースの割り当てが可能です。

ADADEV
ディスクスペース管理（UNIX のみ）

このユーティリティは、Adabas によって使用されるディスクスペースを管理するための複数の機能から構成されています。データベースのスペースを事前に割り当てるために使用します。

ADAELA
Event Analytics 管理

管理ユーティリティ ADAELA は、Event Analytics アドオンを構成します。

ADAELP
イベントログレポート

ADAELP ユーティリティは、Adabas Analytics が作成したイベントログからのイベントを出力します。

ADAERR
エラーファイルレポート

ADAERR ユーティリティは、さまざまなユーティリティによって生成されるエラーファイルの内容を表示します。

ADAFDU
ファイル定義

ファイル定義ユーティリティ ADAFDU は、データベース内にファイルを定義します。このユーティリティの機能は、FCB と FDT をデータベースにロードし、ASSO と DATA の要求スペースを指定されたファイルに割り当てるだけです。

ADAFIN
ファイル情報レポート

ADAFINユーティリティはファイルに関する情報を表示します。例えば、FDT、データストレージ内のブロックのディスクリプタ統計と占有率、ノーマルインデックスとアッパー／メインインデックスなどです。

ADAFRM
新規データベースのフォーマットと作成

フォーマットユーティリティ ADAFRM は、Adabas によって使用されるファイル（アソシエータ、データストレージ、WORK、TEMP、SORT）を割り当て、フォーマットします。ADADEVによって事前に割り当てられているファイルをフォーマットすることもできます。

ADAINV
インバーテッドリストの作成、削除、整合性チェック

インバートユーティリティ ADAINV は、データベースにロードされたファイルのインバーテッドリストの作成、再インバート、削除、または指定されたディスクリプタの整合性チェックを行います。

ADAMON
このユーティリティを使うと、Adabas ニュークリアスのパフォーマンスを監視し、端末でその状態を確認することができます。

ADAMUP
一括追加と削除

ADAMUP ユーティリティは、データベースのファイルに対する大量のレコードの追加または削除を行います。

ADANUC
データベースの開始、ニュークリアスパラメータの定義

ADANUC ユーティリティは、オンライン処理用にデータベースを開始し、ランタイム環境を定義します。

ADAOPR
オペレータユーティリティ

オペレータユーティリティは、Adabas ニュークリアスの操作に使用します。

ADAORD
データベースまたはファイルのリオーダ、ファイルのエクスポート／インポート

リオーダユーティリティ ADAORD は、データベースまたはデータベース内のファイルを再編成する機能（REORDER 機能）とデータベース間でファイルを移行する機能（EXPORT および IMPORT 機能）を提供します。

ADAPLP
プロテクションログの出力

このユーティリティはプロテクションログを出力します。

ADAPRI
Adabas ブロックの出力

ADAPRI ユーティリティは、メンテナンスまたは監査目的で、アソシエータ、データストレージ、WORK、TEMP、または SORT 内のブロックまたはブロック範囲の内容を出力します。

ADARBA
RBAC 管理

ADARBA ユーティリティは、RBAC セキュリティ定義の管理に使用されます。

ADAREC
データベースまたはファイルのリカバリ

このユーティリティは、データベースに加えられた更新を再適用します（REGENERATE 機能）。

ADAREP
データベースレポート

ADAREP ユーティリティはデータベースステータスレポートを生成します。このレポートには、データベースの現在の物理レイアウトおよび論理的な内容に関する情報が含まれます。

このレポートには、アソシエータおよびデータストレージに現在割り当てられているスペースの量と位置、アソシエータおよびデータストレージに利用可能な未使用のスペースの量と位置、データベースファイルのサマリ、チェックポイント情報、セキュリティ情報、データベース内の各ファイルに関する情報（スペース割り当て、利用可能なスペース、ロードされたレコードの数、MAXISN 設定、フィールド定義）が含まれます。

ADASCR

セキュリティ機能

セキュリティユーティリティ ADASCR は、ファイルプロテクションレベルとユーザーパスワードの作成、変更、削除を行い、設定または変更する個々のパスワードのレコードロック機能を（個々のデータベースファイルの値条件を使用して）有効にします。さらに、このユーティリティを使用して、ファイルとパスワードセキュリティ情報を表示することもできます。

ADATST

Adabas コマンドの発行

このユーティリティは、Adabas ニュークリアスにコマンドを発行します。

ADAULD

ファイルのアンロード

アンロードユーティリティ ADAULD は、データベースまたは Adabas バックアップコピーからファイルをアンロードし、圧縮ユーティリティ ADACMP で生成した場合と同じフォーマットで圧縮レコードを生成します。アンロードしたレコードは、圧縮解除ユーティリティ ADADCU の入力として、または一括更新ユーティリティ ADAMUP で使用することができます。レコードは、現在データストレージに格納されている順番、ディスクリプタの順番、または ISN の順番でデータベースからアンロードできます。ただし、レコードをバックアップコピーからアンロードする場合は、ユーティリティによって格納された順番でしかアンロードできません。

ADAVFY

データベース整合性チェック

このユーティリティは、データベースの整合性をチェックします。ロードされたファイルの各ファイルコントロールブロック（FCB）と各フィールド定義テーブル（FDT）とともにジェネラルコントロールブロック（GCB）が検証されます。インデックス構造およびデータストレージも検証されます。指定した場合、ADAVFY は失われた RABN を検索します。

4 ADABAS（データベースニュークリアスの起動）

この章では ADABAS ユーティリティについて説明します。

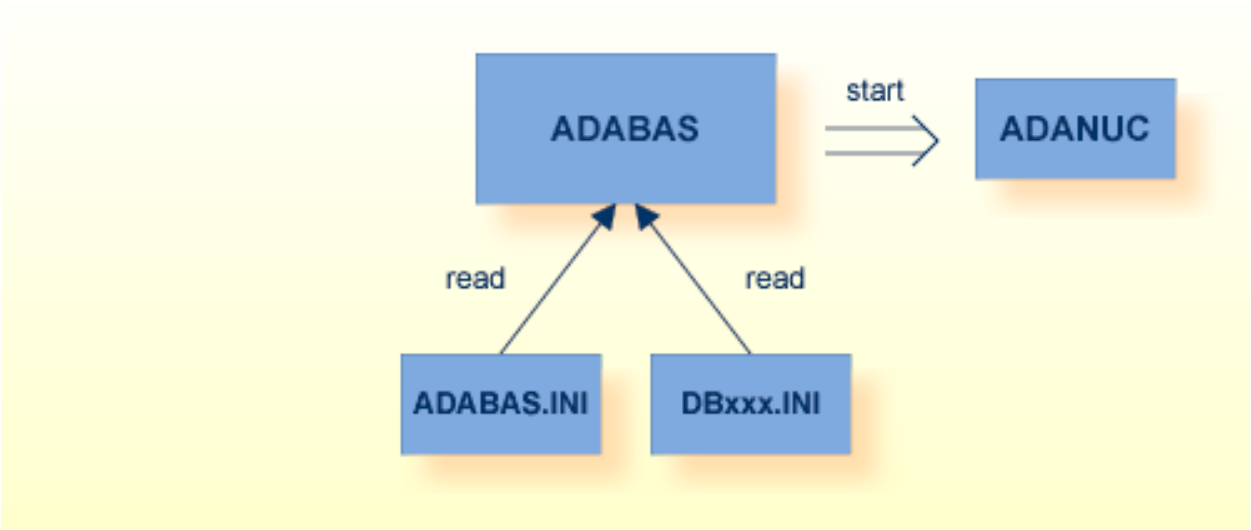
注意: このユーティリティは、Windows プラットフォームでのみ使用できます。

機能概要

ユーティリティ ADABAS を使用すると、データベース初期化ファイル（*DBxxx.INI*）で指定されたニュークリアスパラメータを使用して、データベースニュークリアスを起動できます。データベースニュークリアス ADANUC をパラメータなしで直接起動した場合、DBxxx.INI ファイルは評価されず、ニュークリアスパラメータのデフォルト値が使用されます。

注意: 制御パラメータと値は対話式では入力できません。ユーティリティの起動時にコマンドプロンプトで入力する必要があります。

処理フロー



1. ADABASは、グローバル初期化ファイル「%ADADATADIR%\ETC\ADABAS.INI」を読み取り、開始するデータベースのデータベース初期化ファイルを取得します。
2. ADABASは、データベース初期化ファイルを読み取り、開始するデータベースのニュークリアスパラメータを取得します。
3. ADABAS は、読み取ったパラメータを使用して ADANUC を開始します。

注意: データベース初期化ファイルの詳細については、「*拡張オペレーションセクション*」を参照してください。

制御パラメータ

次にあげる制御パラメータが使用可能です。

```
M     [DBID =] number
```

DBID

```
[DBID  =] number
```

このパラメータは、使用対象となるデータベースを選択するためのものです。

例：

データベース 20 を起動するには次のいずれかを入力します。

```
adabas dbid=20
```

または

```
adabas 20
```

5 ADABCK（データベースまたはファイルのダンプおよびリストア）

この章では ADABCK ユーティリティについて説明します。

機能概要

バックアップユーティリティ ADABCK は、Adabas バックアップコピーを作成して、データベースが壊れたときに備えます。ADABCK は定期的に使用してください。

このユーティリティは、データベースまたは選択ファイルをデータベースからダンプ、またはデータベースにリストアします。

ADABCK は、同じプラットフォーム上で作成された入力ファイルも、異なるエンディアンモードのプラットフォーム上で作成された入力ファイルも処理できます。ファイルが書き込まれたフォーマットは、リストア処理によって認識されます。リストア中に、エンディアンモードに依存するすべてのデータが、ターゲットプラットフォームの要件に合致するよう変換されます。

このユーティリティは、データベースの内部構造を利用して最適なパフォーマンスを提供します。未使用ブロックは読み取る必要がないため、ダンプ時に省略されます。そのようなブロックは、Adabas バックアップコピーに含まれていなくてもリストア中に再作成できます。

バックアップコピーはテープに直接割り当てることができ、このオプションは連続するテープをサポートします（『*Adabas Basics*』の「*ユーティリティの使用*」を参照）。

さらに、バックアップコピーは、バックアップデータのパイプ渡しをサポートするために、stdout に送ることができます（この機能は UNIX プラットフォームのみで使用可能です）。この機能は環境変数（BCK001）を -（マイナス）に設定すると有効になります。この場合、出力メッセージは stderr に送られます。RESTORE および OVERLAY 機能もこの方法で使用することができます。つまり、バックアップコピーは stdin から読み取ることができます。この場合、ADABCK コントロールステートメントはコマンド行に指定する必要があります『*Adabas Basics*』の「*ユーティリティの使用*」を参照してください。

利用できる機能は次のとおりです。

- COPY 機能は、Adabas バックアップコピーをコピーします。バックアップデータセットは、同じエンディアンモードのマシン上でのみ複製できます。異なるエンディアンモードのマシン上でバックアップを複製しようとすると、拒否されます。
- DUMP 機能は、データベースまたは選択ファイルをデータベースから Adabas バックアップコピーと呼ばれる 1 つまたは複数のシーケンシャルファイルにダンプします。ダンプの進行中に、ニュークリアスがアクティブな可能性があり、この場合はダンプ対象ファイルの同時更新が許されます。DUMP 機能は、プロセッサのエンディアンモードでデータを書き込みます。
- EXU_DUMP 機能は、データベースまたは選択ファイルをデータベースから Adabas バックアップコピーと呼ばれる 1 つまたは複数のシーケンシャルファイルにダンプします。ダンプの実行中には、ACC ユーザーのみがダンプ対象ファイルへのアクセスが許されます。
- IOSTAT 機能は、データ転送率および I/O 待ち時間に関する情報を表示します。

- OVERLAY 機能は、選択ファイルまたはデータベースをリストアします。リストア対象ファイルは、すでにデータベース内にロードされている可能性があります。その場合、ADABCK は、ファイルをリストアする前に黙示的な削除を実行します。
- READ_CHECK 機能は、Adabas バックアップコピーの読み取り可能性（すなわち、パリティエラーが存在しないこと）および完全性をチェックします。これらのチェックは、ダンプファイルが RESTORE または OVERLAY 機能によって読み取られることを確実にするためのものです。
- RESTORE 機能は、既存の Adabas バックアップコピーからデータベースまたは選択ファイルをリストアします。ターゲットデータベース内のファイルにセキュリティ定義がない場合には、そのファイルのリストア時に対応するエントリが（ファイルダンプ時の定義のまま）セキュリティテーブルにセットアップされます。
- リスト機能 CONTENTS、FILES、および SUMMARY は、Adabas バックアップコピーに関する情報を表示します。リスト機能を使用するとき、DBID を最初に入力する必要はありません。バックアップファイルが RAW セクションにあるときは例外です。この場合は DBID が必要になりますが、データベース自体は存在する必要がありません（UNIX プラットフォームのみ）。

DUMP、EXU_DUMP、OVERLAY および RESTORE 機能は相互に排他的であり、このユーティリティの 1 回の実行では、それらの機能の 1 つだけを実行できます。リスト機能は、READ_CHECK、RESTORE または OVERLAY 機能とだけ一緒に使用できます。

RESTORE 機能または OVERLAY 機能を実行し、データベースが小さすぎるかデータベースコンテナがない場合、ADABCK ではデータベースのサイズの拡大またはコンテナの作成が自動的に行われます。

注意:

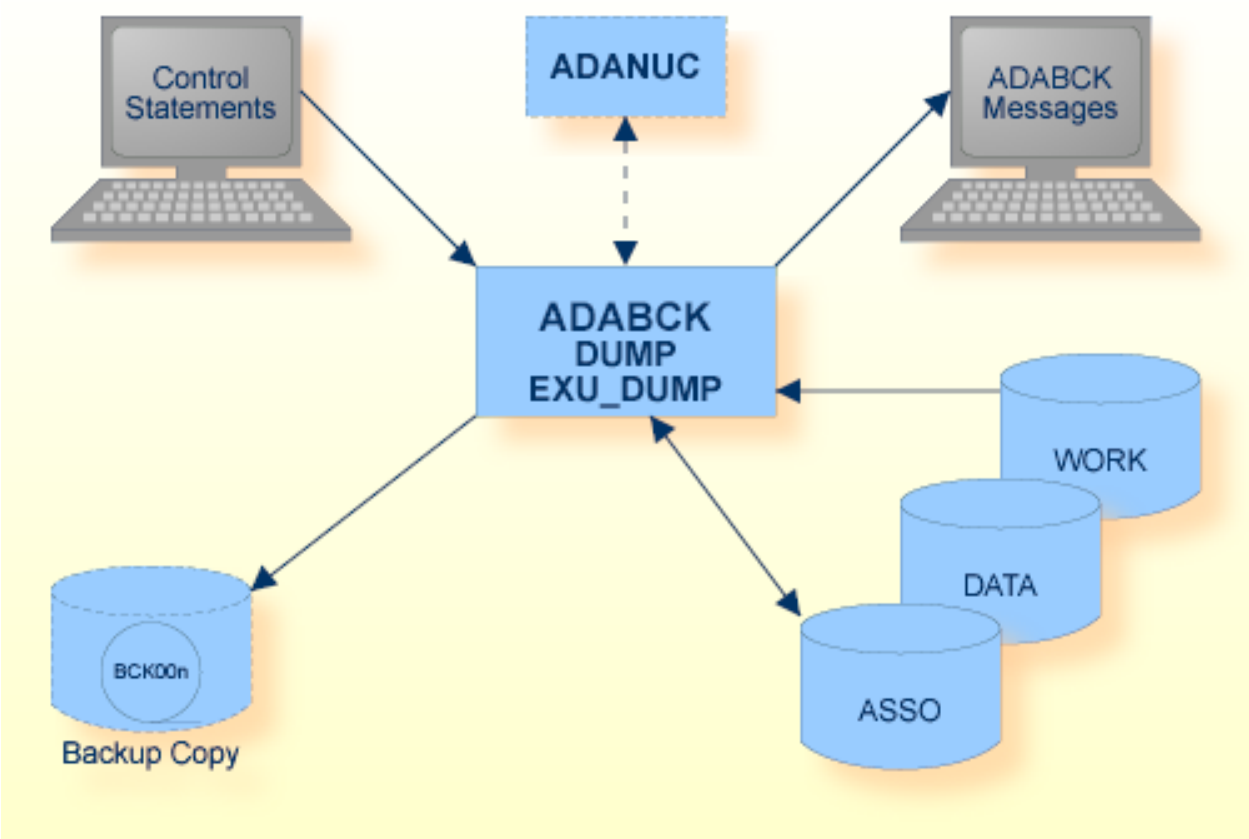
1. RESTORE および OVERLAY 機能は、以前の Adabas バージョンで作成されたバックアップファイルを処理できますが、通常、より新しい Adabas バージョンで作成されたバックアップファイルは処理できません。ただし、構造が変更されていない場合は、以前の Adabas バージョンでもリストアできます。例えば、文字セットエンコーディングを含むスーパーディスクリプタを使用していなければ、バージョン 6.6 のバックアップファイルをバージョン 6.5 でリストアできます。以前のバージョンで機能がサポートされていない場合は、構造レベルチェックが失敗します。
2. RESTORE オプション PARALLEL=MULTIPROCESS は、異なるエンディアンモードのプラットフォームで作成されたバックアップファイルに対してはサポートされません。ただし、複数のエクステントに分割されたバックアップはロードできます。

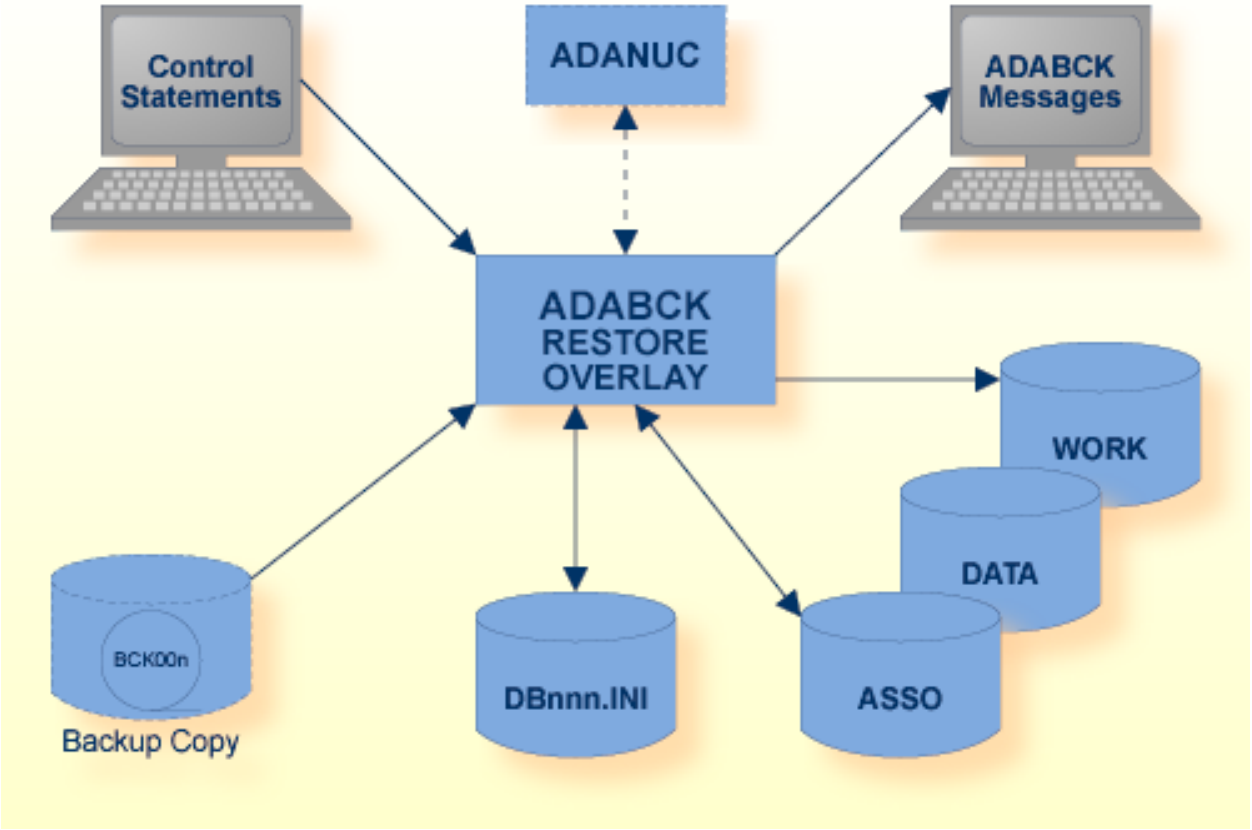
注意: Adabas INI ファイルを使用せず、代わりに環境変数を使用してコンテナファイル名を指定し、かつ ADABCK を開始する前に環境変数／論理名の割り当てを忘れた場合は、データベースのコピーがデータベースディレクトリに作成されます。Adabas ニュークリアスがアクティブなときにファイルオーバーレイまたはリストアを実行し、データベースを拡張する必要がある場合、データベースは ADABCK ではなくニュークリアスによって拡張されます。この場合 OPTION=AUTOEXPAND が指定されていなくても、ニューク

リアスによってデータベースが拡張されます。環境変数を使用してデータベースコンテナを指定する場合は、リストア／オーバーレイ用に新しいコンテナを作成する必要があるときに、以下の点を考慮する必要があります。ニュークリアスは、新しいコンテナの正しい環境変数設定で開始することが重要です。新しいコンテナはニュークリアスによって作成されるので、ADABCK プロセスの環境変数を指定しても効果がありません。

このユーティリティは単一機能ユーティリティです。

処理フロー


Control Statements
ADANUC
ADABCK Messages
ADABCK
DUMP
EXU_DUMP
WORK
DATA
ASSO
BCK00n
Backup Copy



データセット	環境 変数／ 論理名	記憶 媒体	追加情報
アソシエータ	ASSOx	ディスク	
バックアップコピー	BCK00n	ディスク、テープ（注 1 参照） stdin/ SYS$INPUT（注 2 参照）、 stdout/ SYS$OUTPUT（注 3 参照）	DUMP/EXU_DUMP 機能の出力、 その他の機能の入力
	BCKOUT	ディスク、テープ（注 1 参照）	COPY 機能の出力
データストレージ	DATAx	ディスク	
DBnnn.INI		ディスク	*Adabas 拡張オペレーション* マニュアル
コントロールステートメント	stdin SYS$INPUT		*ユーティリティマニュアル*

データセット	環境 変数／ 論理名	記憶 媒体	追加情報
ADABCK メッセージ	stdout/ SYS$OUTPUT（注4参照）、 stderr/ SYS$ERR（注5参照）		*メッセージとコード*
WORK	WORK1	ディスク	

注意:

1. このシーケンシャルファイルには、名前付きパイプを使用できます（UNIX プラットフォームのみ、詳細については、『*Adabas Basics*』の「*ユーティリティの使用*」を参照してください）。
2. DUMP または EXU_DUMP 以外の機能で（BCK001 のみ）使用します。
3. DUMP または EXU_DUMP の機能で（BCK001 のみ）使用します。
4. BCK001 が stdout/SYS$OUTPUT ではない場合。
5. BCK001 が stdout/SYS$OUTPUT の場合。

シーケンシャルファイル BCK00n は、複数エクステントを持つことができます。複数のエクステントを持つシーケンシャルファイルの詳細については、『*管理マニュアル*』の「*ユーティリティの使用*」を参照してください。

チェックポイント

次の表は、各機能に対するニュークリアス条件と記録チェックポイントを示しています。

機能	ニュークリアスのアクティブ化が必要	ニュークリアスの非アクティブ化が必要	ニュークリアスは不要	記録チェックポイント
CONTENTS			X	-
COPY			X	-
DUMP	X（注 1 参照）		X	SYNX
EXU_DUMP	X（注 1 参照）		X	SYNX
FILES			X	-
NEW_PLOG				SYNC
OVERLAY		X（注 2 参照）	X（注 3 参照）	SYNP
READ_CHECK			X	-
RESTORE		X（注 2 参照）	X（注 3 参照）	SYNP

機能	ニュークリアスのアクティブ化が必要	ニュークリアスの非アクティブ化が必要	ニュークリアスは不要	記録チェックポイント
SUMMARY			X	-

注意:

1. ニュークリアスは、AUTORESTART がこの機能の最後で保留状態のときのみ必要です。
2. データベースまたはシステムファイルのリストアの場合。
3. ファイルのリストアの場合。

制御パラメータ

次のコントロールパラメータを使用できます。

```
     CONTENTS

     COPY [= number]

M    DBID = number

     DUMP = {*|(number[-number][,number[-number]]...)}
            [,BLOCKSIZE = number [K|M]]
D           [{,DRIVES = number} |
D           {,[NO]DUAL } ]
            [,ET_SYNC_WAIT = number]
D           [,[NO]NEW_PLOG]
            [,REPLICATION]

     EXU_DUMP = {*|(number[-number][,number[-number]]...)}
          [[,BLOCKSIZE = number [K|M]]
D         [{,DRIVES = number} |
D         {,[NO]DUAL} ]
D         [,[NO]NEW_PLOG]
            [,REPLICATION]

     FILES = { * | (number[-number][,number[-number]]...)}

     IOSTAT

     OVERLAY = {*|(number[-number][,number[-number]]...)}
         [,FMOVE [=(number [,number [-number]]...)]]
         [,FORMAT = (keyword [,keyword])]
         [,KEEP_FILE_ALLOC]
         [,NEW_DBID = number]
         [,RENUMBER = (number[-number] [,number [-number]]...)]]
         [,REPLICATION]
```

```
    PARALLEL = keyword

    READ_CHECK

    RESTORE = {*|(number[-number][,number[-number]]...)}
        [,FMOVE [=(number [,number [-number]]...)]]
        [,FORMAT = (keyword [,keyword])]
        [,NEW_DBID = number]
        [,RENUMBER = (number[-number] [,number [-number]]...)]]
        [,REPLICATION]

    SUMMARY
```

CONTENTS

```
CONTENTS
```

このパラメータは、DUMP または EXU_DUMP パラメータで作成された Adabas バックアップコピー内のファイルのリストを表示するものです。

例

```
adabck cont
%ADABCK-I-STARTED,      30-OCT-2015 11:42:37, <Version number>
Files dumped on 30-OCT-2015 10:51:14

Database 34, GENERAL-DATABASE

File     4, Update-log      , loaded on 17-SEP-2014 14:44:19
File     9, EMPLOYEES       , loaded on  8-OCT-2008 17:59:40
File    14, miscellaneous   , loaded on 11-JUN-2015 13:22:19
File    17, Timezone        , loaded on 19-SEP-2014 11:44:42
File    19, LARGE           , loaded on  2-SEP-2014 15:37:18
File    51, PCA24SYSF1      , loaded on 14-APR-2014 16:55:22
File    91, ADAOS-2544      , loaded on  8-APR-2015 13:19:27
File    95, P299255         , loaded on 20-MAR-2014 11:35:30
File    98, ADAOS-4591      , loaded on 16-JUL-2015 10:03:16
File  1009, LOBFILE of 9    , loaded on  8-OCT-2008 17:59:40



%ADABCK-I-IOCNT, 1 IOs on dataset BCK001
%ADABCK-I-TERMINATED,   30-OCT-2015 11:42:37, elapsed time: 00:00:00
```

COPY

```
COPY [= number]
```

このパラメータは、既存のAdabasバックアップコピーから新規ファイルを作成するものです。入力ファイル（BCK0xx）および出力ファイル（BCKOUT）は、ディスクまたはテープ上のいずれにあっても構いません。ここでxxは、指定した番号、または番号が指定されない場合は01のいずれかです。

DBID

```
DBID = number
```

このパラメータは、使用対象となるデータベースを選択するためのものです。

DUMP

```
DUMP = { * | (number[-number][,number[-number]]...)}
       [,BLOCKSIZE = number [K|M]]
       [ {,DRIVES = number} |
       {, [NO]DUAL } ]
       [,ET_SYNC_WAIT = number ]
       [,[NO]NEW_PLOG ]
       [,REPLICATION] ↩
```

この機能は、ファイルレベルの場合、リスト内の番号で指定されたファイルをダンプします。LOBファイルが指定された場合は無視されますが、すべての基本ファイルに割り当てられたLOBファイルはダンプされます。アスタリスク'*'は、データベース全体をダンプすることを意味します。ダンプの実行中に、ダンプ対象ファイルに対する同時更新が許されます。

ニュークリアスが並行して稼働している（オンラインバックアップ）場合、ADABCKは、ADABCKが終了するまでに、ダンプされるファイルに影響するすべてのトランザクションが、全ユーザーによって完了されていることを確認する必要があります。これはET同期と呼ばれます。詳細については、『*管理マニュアル*』の「*ET同期*」セクションを参照してください。NONEW_PLOGオプションを指定してファイルレベルでダンプを実行すると、ET同期はファイルレベルで実行されます。そうでない場合は、データベース全体に対してET同期が実行されます。

参照制約を含むファイルを指定する場合、参照整合性を維持するために、参照制約を介してこれらのファイルに接続したすべてのファイルも指定する必要があります。

BLOCKSIZE = number[K|M]

このパラメータを指定すると、I/O 転送ブロックサイズを変更できます。PARALLEL を指定した場合、デフォルトブロックサイズは 512 KB になります。次の値を指定することができます。64KB、128KB、256KB、512KB、1MB、2MB、...12MB。指定されたブロックサイズは、次回の RESTORE 機能で使用されます。

DRIVES = number

このパラメータは、並行して処理する出力デバイスの最大数を制限します。バックアップファイルを複数のエクステントに分割するのに使用できます。出力は BCK0xx に送られます。

デフォルト値は 1、最大値は 10 です。

パラメータ DRIVES と DUAL は相互に排他的です。DUMP 機能の 1 回のコールで指定できるのはいずれか一方のみです。

[NO]DUAL

DUAL は、ダンプ情報の物理的なコピーを 2 つ作成することを指定します。出力は BCK001 と BCK002 に送られます。

デフォルトは NODUAL です。

パラメータ DUAL と DRIVES は相互に排他的です。DUMP 機能の 1 回のコールで指定できるのはいずれか一方のみです。

ET_SYNC_WAIT = number

このパラメータは、DUMP 機能の最後で ET ロジックユーザーが ET ステータスになるまで ADABCK が待機する時間（秒単位）を定義します。ET 同期の開始時に、トランザクションがすでに ET_SYNC_WAIT で指定された秒数（またはそれ以上）の間アクティブになっている場合、待機時間は 0 です。それ以外の場合は、トランザクションの待機時間（秒単位）は、パラメータに指定された値から、ET 同期が開始された時点でトランザクションがアクティブになっていた秒数を引いた値になります。待機時間が経過した時点でまだ終了していないトランザクションはロールバックされます。

このパラメータを省略すると、ET 同期は、通常の ADABAS タイムアウトロジック（ADANUC パラメータ TT）を使用して、すべてのオープントランザクションが終了するまで待機します。

最小値は 1 で最大値は 32767 です。

注意:

1. ADABCK の ET_SYNC_WAIT パラメータの指定を忘れ、オープントランザクションによって ADABCK がハングしている場合は、次のいずれかを実行して、ADABCK を続行できます。一時的に TT を小さな値に設定します。ADABCK が終了した後で、値を元に戻すことができ

ます。または、オープントランザクションがあるユーザーを停止します（ADAOPR STOP を使用）。

2. ET同期中にデータベースで更新が実行された場合、すべてのオープントランザクションがコミットまたはロールバックされたときに、変更されたすべてのブロックをデータベースコンテナおよびバックアップコピーに書き込む必要があります。したがって、ET同期の合計時間（IOSTATパラメータで表示可能）は、ET_SYNC_WAITパラメータで指定された時間よりも長くなる場合があります。

[NO]NEW_PLOG

このオプションは、DUMP 機能の最後でプロテクションログファイルをクローズして新しいログファイルを作成するかどうかを指定します。

データベースダンプのデフォルトはNEW_PLOGです。ファイルダンプの場合はNONEW_PLOGです。

注意: V6.3 SP1 Fix 13 以前では、ファイルダンプのデフォルトは NEW_PLOG でした。ほとんどの場合、この変更は影響を与えませんが、PLOG スイッチが実際に必要な場合は、NEW_PLOG を明示的に指定する必要があります。

REPLICATION

パラメータ REPLICATION は、Adabas を搭載した Adabas Event Replicator（Adabas レプリケーション）を使用している顧客にのみ関係します。

このパラメータは、Adabas に ADABCK を使用する場合に指定する必要があります（Adabas レプリケーションの初期状態処理）。このパラメータを指定する場合、ダンプされるファイルのレプリケーションのステータスは、自動的に更新されます。

詳細については、ADAOPR CHANGE_REPLICATION_STATUS を参照してください。

EXU_DUMP

```
EXU_DUMP = {*|(number[-number][,number[-number]]...)}
           [,BLOCKSIZE = number [K|M]]
           [ {,DRIVES = number} |
           {,[NO]DUAL} ]
           [,[NO]NEW_PLOG]
           [,REPLICATION] ↩
```

この機能は、ファイルレベルの場合、リスト内の番号で指定されたファイルをダンプします。LOBファイルが指定された場合は無視されますが、すべての基本ファイルに割り当てられたLOBファイルはダンプされます。アスタリスク'*'は、データベース全体をダンプすることを意味します。ダンプの実行中には、ACC ユーザーのみがダンプ対象ファイルへのアクセスが許可されます。ET 同期は必要ありません。

参照制約を含むファイルを指定する場合、参照整合性を維持するために、参照制約を介してこれらのファイルに接続したすべてのファイルも指定する必要があります。

BLOCKSIZE = number[K|M]

このパラメータを指定すると、I/O 転送ブロックサイズを変更できます。PARALLEL を指定した場合、デフォルトブロックサイズは 512 KB になります。次の値を指定することができます。64KB、128KB、256KB、512KB、1MB、2MB、...12MB。指定されたブロックサイズは、次回の RESTORE 機能で使用されます。

DRIVES = number

このパラメータは、並行して処理する出力デバイスの最大数を制限します。バックアップファイルを複数のエクステントに分割するのに使用できます。出力は BCK0xx に送られます。

デフォルト値は 1、最大値は 10 です。

パラメータ DRIVES と DUAL は相互に排他的です。DUMP 機能の 1 回のコールで指定できるのはいずれか一方のみです。

[NO]DUAL

DUAL は、ダンプ情報の物理的なコピーを 2 つ作成することを指定します。出力は BCK001 と BCK002 に送られます。

デフォルトは NODUAL です。

パラメータ DUAL と DRIVES は相互に排他的です。DUMP 機能の 1 回のコールで指定できるのはいずれか一方のみです。

[NO]NEW_PLOG

このオプションは、EXU_DUMP 機能の最後でプロテクションログファイルをクローズして新しいログファイルを作成するかどうかを指定します。

ダンプするファイルが 1 つだけの場合には、このオプションを使用しないでください。

EXU_DUMP=* の場合には、デフォルトは NEW_PLOG です。

REPLICATION

パラメータ REPLICATION は、Adabas を搭載した Adabas Event Replicator（Adabas レプリケーション）を使用している顧客にのみ関係します。

このパラメータは、Adabas に ADABCK を使用する場合に指定する必要があります（Adabas レプリケーションの初期状態処理）。このパラメータを指定する場合、ダンプされるファイルのレプリケーションのステータスは、自動的に更新されます。

詳細については、ADAOPR CHANGE_REPLICATION_STATUS を参照してください。

DUMP/EXUDUMP の例

例 **1**

データベースは、並行して 3 台の出力デバイスにダンプされます。

```
adabck db=34 parallel=multi_process dump=\* drives=3
%ADABCK-I-STARTED,      30-OCT-2015 11:05:25, <version number>
%ADABCK-I-DBOFF, database 34 accessed offline

Database dumped on 30-OCT-2015 11:05:25

Database 34, GENERAL-DATABASE

File     1, CHECKPOINT-FILE , loaded on  4-SEP-2014 13:52:43
File     2, SECURITY-FILE   , loaded on  4-SEP-2014 13:52:43
File     3, USER-DATA-FILE  , loaded on  4-SEP-2014 13:52:43
File     4, Update-log      , loaded on 17-SEP-2014 14:44:19
File     9, EMPLOYEES       , loaded on  8-OCT-2008 17:59:40
File    14, miscellaneous   , loaded on 11-JUN-2015 13:22:19
File    17, Timezone        , loaded on 19-SEP-2014 11:44:42
File    19, LARGE           , loaded on  2-SEP-2014 15:37:18
File    30, FILE-30         , loaded on 31-MAR-2015 13:40:33
File    51, PCA24SYSF1      , loaded on 14-APR-2014 16:55:22
File    80, P295170         , loaded on  4-AUG-2015 12:42:43
File    91, ADAOS-2544      , loaded on  8-APR-2015 13:19:27
File    95, P299255         , loaded on 20-MAR-2014 11:35:30
File    98, ADAOS-4591      , loaded on 16-JUL-2015 10:03:16
File   101, COLLATION-TESTS , loaded on 14-APR-2014 16:47:30
File   111, TESTOPT         , loaded on 29-NOV-2011 10:34:23
File   113, lob_LB          , loaded on 23-JUN-2015 15:48:48
File   146, XMA-REPOSITORY  , loaded on 10-DEC-2014 09:39:52
File   215, ADAOS-4647      , loaded on 27-APR-2012 15:52:49
File  1009, LOBFILE of 9    , loaded on  8-OCT-2008 17:59:40
File  1080, LOBFILE of 80   , loaded on  4-AUG-2015 12:42:43
File  1113, LOBFILE of 113  , loaded on 23-JUN-2015 15:48:48
File 22111, LOBFILE of 111  , loaded on 29-NOV-2011 10:34:23
```

```
%ADABCK-I-IOCNT, 2 IOs on dataset WORK
%ADABCK-I-IOCNT, 135 IOs on dataset DATA
%ADABCK-I-IOCNT, 275 IOs on dataset ASSO
%ADABCK-I-IOCNT, 41 IOs on dataset BCK001
%ADABCK-I-IOCNT, 34 IOs on dataset BCK002
%ADABCK-I-IOCNT, 80 IOs on dataset BCK003
%ADABCK-I-TERMINATED,   30-OCT-2015 11:05:26, elapsed time: 00:00:01
```

例 **2**

ファイル215がダンプされ、バックアップの2つの物理コピーが作成されます。ダンプの実行中は、ACC ユーザーのみファイル 30 へのアクセスが許されます。

```
adabck db=34 exu_dump=215 dual
%ADABCK-I-STARTED,      30-OCT-2015 10:45:43, <version number>
%ADABCK-I-DBON, database 34 accessed online

Files dumped on 30-OCT-2015 10:45:44

Database 34, GENERAL-DATABASE

File   215, ADAOS-4647      , loaded on 27-APR-2012 15:52:49


%ADABCK-I-IOCNT, 51 IOs on dataset DATA
%ADABCK-I-IOCNT, 29 IOs on dataset ASSO
%ADABCK-I-IOCNT, 40 IOs on dataset BCK001
%ADABCK-I-IOCNT, 40 IOs on dataset BCK002
%ADABCK-I-TERMINATED,   30-OCT-2015 10:45:44, elapsed time: 00:00:01
```

例 **3**

91～99、51、または4～19のファイル番号を持つデータベース内のすべての基本ファイルがダンプされます（指定したファイル範囲にない場合でも、対応する LOB ファイルが含まれます）。ADABCK は、ET ロジックユーザーが ET ステータスになるまで最大 10 秒間待機します。

```
adabck db=34 dump=\(91-99,51,4-19\) et_sync_wait=10
%ADABCK-I-STARTED,      30-OCT-2015 10:51:14, <version number>
%ADABCK-I-DBON, database 34 accessed online

Files dumped on 30-OCT-2015 10:51:14

Database 34, GENERAL-DATABASE

File     4, Update-log      , loaded on 17-SEP-2014 14:44:19
File     9, EMPLOYEES       , loaded on  8-OCT-2008 17:59:40
File    14, miscellaneous   , loaded on 11-JUN-2015 13:22:19
File    17, Timezone        , loaded on 19-SEP-2014 11:44:42
File    19, LARGE           , loaded on  2-SEP-2014 15:37:18
File    51, PCA24SYSF1      , loaded on 14-APR-2014 16:55:22
File    91, ADAOS-2544      , loaded on  8-APR-2015 13:19:27
```

```
File    95, P299255         , loaded on 20-MAR-2014 11:35:30
File    98, ADAOS-4591      , loaded on 16-JUL-2015 10:03:16
File  1009, LOBFILE of 9    , loaded on  8-OCT-2008 17:59:40


%ADABCK-I-IOCNT, 715 IOs on dataset DATA
%ADABCK-I-IOCNT, 1145 IOs on dataset ASSO
%ADABCK-I-IOCNT, 1195 IOs on dataset BCK001
%ADABCK-I-TERMINATED,   30-OCT-2015 10:51:16, elapsed time: 00:00:02
```

ファイル 1、2、4、6、8、10、11 および 13 がダンプされます。ADABCK は、ET ロジックユーザーが ET ステータスになるまで最大 5 秒待ちます。

FILES

```
FILES = { * | (number[-number][,number[-number]]...)}
```

このパラメータは、指定したファイルのステータス情報をダンプファイルに出力します。

IOSTAT

```
IOSTAT
```

このパラメータを指定すると、さまざまなデバイスのデータの転送率およびI/O（待ち）時間がADABCK 処理の最後に出力されます。

例：

```
adabck db=36 parallel=multi_process dump=\* drives=3 iostat
...
-----------------------------------------------------------------
Dump Method     :  parallel
Blocksizes      :  DB:              512 KB     BCK:           512 KB
DB I/O time     :  total:         27.09 sec    average:      8084 us
BCK  1 I/O time :  total:          1.16 sec    average:      7606 us
BCK  2 I/O time :  total:          0.00 sec    average:       944 us
BCK  3 I/O time :  total:          1.24 sec    average:      1375 us
Wait rates      :     waits    nowaits        rate   mreq
DB              :      1439       1898         43%    8
Transfer rate   :  15215 KB/sec
-----------------------------------------------------------------

%ADABCK-I-IOCNT, 2 IOs on dataset WORK
%ADABCK-I-IOCNT, 3147 IOs on dataset DATA
%ADABCK-I-IOCNT, 229 IOs on dataset ASSO
%ADABCK-I-IOCNT, 153 IOs on dataset BCK001
%ADABCK-I-IOCNT, 2 IOs on dataset BCK002
%ADABCK-I-IOCNT, 906 IOs on dataset BCK003
```

IOSTAT 統計には、次の情報が表示されます。

Dump Method
PARALLEL パラメータの設定に応じて、parallel または non-parallel になります。

DB I/O time
I/O 時間の合計（秒）と、ASSO および DATA コンテナへのアクセスのための I/O 処理あたりの平均時間（マイクロ秒）。

BCK n I/O time
I/O 時間の合計（秒）と、バックアップファイルへのアクセスのための I/O 処理あたりの平均時間（マイクロ秒）。

注意: 測定される I/O 時間は、I/O システム機能に必要とされた時間です。ディスクへのアクセスに実際に必要な物理I/O時間とは異なる場合があります。これは、オペレーティングシステムまたはストレージシステム内のキャッシュ、および非同期I/Oの使用が要因となります。

Wait rates（Dump Method が parallel の場合のみ）
並列バックアップ／リストアの場合、データベースコンテナの I/O は非同期で実行されます。待機レートには、待機処理が必要なASSOまたはデータI/Oの数が表示されます。mreq は、データベースコンテナに対するパラレル I/O 要求の最大数です。

注意: 実際のバックアップまたはリストアの I/O のみがカウントされます。ADABCK のスタートアップフェーズでは、いくつかの追加I/Oが必要になります。したがって、wait と nowait I/O の合計は、ASSO と DATA I/O の合計より少なくなります。

BF Sync Count（オンラインモードのバックアップのみ）
バッファフラッシュ中にオンラインモードでバックアップを実行する場合は、ニュークリアスと同期して、バッファフラッシュでディスクに書き込まれて変更されたデータベースブロックもバックアップファイルに書き込む必要があります。BF sync count とは、これらのバッファフラッシュ同期の数です。

ET sync time（オンラインモードのバックアップのみ）
オンラインモードでのバックアップの終了時に、ET 同期が必要です。つまり、すべての ET ロジックユーザーが ET ステータスになるまで ADABCK が待機する必要があります。ET sync time とは、この ET 同期に必要な時間です。

Transfer rate
これは、バックアップファイルに対して読み書きされたキロバイト（1 秒あたり）です。

注意:

1. 転送レートについては、純粋なバックアップ／リストア時間のみが考慮されます。バックアップ／リストアの準備に必要な時間は考慮されません。そのため、ADABCK の合計経過時間に基づいて転送レートを計算すると、表示された転送レートより高くなる場合があります。
2. 小規模なバックアップの場合、計算で丸め誤差が発生する可能性があります。したがって、バックアップが非常に小さい場合、転送レートは表示されません。これは、値が不正確になるためです。

3. 通常、多くのデータベースブロックは完全には入力されていません。正味データのみがバックアップファイルにコピーされるので、転送レートは、処理済みデータベーススペースを考慮した場合の転送レートよりも低くなります。

OVERLAY

```
OVERLAY = {*|(number[-number][,number[-number]]...)}
          [,FMOVE [=(number [,number [-number]]...)]]
          [,FORMAT = (keyword [,keyword] ) ]
          [,KEEP_FILE_ALLOC]
          [,NEW_DBID = number]
          [,RENUMBER = (number[-number] [,number [-number]]...)]]
          [,REPLICATION]
```

このパラメータは、引数リスト内の数値によって指定されるファイルをファイルレベルでリストアするものです。LOB ファイルが指定された場合は無視されますが、すべての基本ファイルに割り当てられたLOBファイルはリストアされます。リストア対象ファイルは、すでにデータベース内にロードされていても構いません。その場合、ファイルをリストアする前に、ADABCK は暗黙の削除を実行します。LOB グループのファイルが1つだけオーバーレイされている場合は、LOB グループのその他のファイルも削除されます。アスタリスク（*）は、データベースレベルでリストアを行うことを意味します。データベースコンテナファイルに対する排他制御が必要です。

他のファイルへの参照整合性制約がある場合でも、指定されたファイルのみがオーバーレイされます。これらの参照整合性制約は削除されます。

FMOVE [=(number [,number [-number]]...)]

このキーワードを指定すると、ADABCK は、オーバーレイするファイルまたは指定サブセットをバックアップと同じブロック範囲にリストアするのではなく、割り当て直します。このキーワードを使用すると、ファイルエクステントの数が可能な限り削減されます。

FORMAT = (keyword [,keyword])

キーワード ASSO と DATA のどちらか、または両方を指定できます。このパラメータは、アソシエータブロックやデータストレージブロックのフォーマットに使用します。ファイルレベルでリストアする際は、ファイルのエクステントの未使用エリアに含まれるブロックのみがフォーマットされます。

KEEP_FILE_ALLOC

このパラメータを指定すると、ファイルをバックアップと同じブロック範囲にリストアするのではなく、現在データベースに存在している状態のまま割り当てようと試みます。例えば、このキーワードは、バックアップが作成されてからファイルが再編成されたとき、またより多くのスペースが前もってファイルに割り当てられたときに使用することができます。バックアップに存在するファイルが、データベースで現在使用可能なブロックよりも多いブロックが必要な場合、残りのブロックは任意の位置に割り当てられます。このキーワードはファイルリストと一緒にしか使用することができません。

NEW_DBID = number

このパラメータは、リストアするデータベースの ID を変更するために使用します。データベース全体をリストアする場合にしか指定できません。

新しい ID を使用すると、アクティブなデータベースのバックアップコピーを別セットのコンテナファイルにリストアできます。新しい ID をアクティブなデータベースと同一にすることはできません。

このパラメータを省略した場合、データベース ID は変更されません。

RENUMBER = (number[-number] [,number [-number]]...)

RENUMBERは、ターゲットデータベースにオーバーレイされるファイルに番号を振り直すために使用されます。次の制限事項と要件が適用されます。

- OVERLAY ファイルリストで指定されたファイルと RENUMBER ファイルリストで指定されたファイルの間には 1：1 の関係が必要です。
- OVERLAY ファイルリストで範囲を指定する場合は、RENUMBER ファイルリストの対応する範囲が同じサイズになっている必要があります。
- 通常、OVERLAY ファイルリストには LOB ファイルを指定する必要はありません。ただし、LOB ファイルの番号も変更する場合は、LOB ファイルも指定する必要があります。
- ファイルは、OVERLAYファイルリストで複数回指定される場合があります（例：(11-55),(44-99)）。この場合、同じソースファイル番号に対して異なるターゲットファイル番号は指定できません。ファイルリストの例で、RENUMBER=(1011-1055,1044-1099) は正しい指定ですが、RENUMBER=(1011-1055,2044-2099) は正しくありません。
- 複数のファイルを同じターゲットファイル番号に番号変更することはできません。

REPLICATION

パラメータ REPLICATION は、Adabas を搭載した Adabas Event Replicator（Adabas レプリケーション）を使用している顧客にのみ関係します。

このパラメータは、Adabas に ADABCK を使用する場合に指定する必要があります（Adabas レプリケーションの初期状態処理）。このパラメータを指定する場合、Adabas ファイルが自動的にレプリケーションターゲットファイルとしてマークされます。

詳細については、ADAOPR CHANGE_REPLICATION_STATUS を参照してください。

PARALLEL

```
PARALLEL = keyword
```

このパラメータは、並行にデバイスを使用して処理の遅いデバイス（テープデバイスなど）にあるバックアップから作成／リストアを行うとき、その処理速度を早めるために指定します。キーワード MULTI_PROCESS を指定できます。PARALLEL=MULTI_PROCESS を指定すると、BLOCKSIZE パラメータのデフォルト値は 512 KB になります。

バックアップファイル数（DUMP または EXU_DUMP の ADABCK サブパラメータ DRIVES）が 1 より大きい場合にのみ、ADABCK 操作が並行して実行されます。

注意:

1. ADABCK RESTORE または ADABCK OVERLAY の場合は、OVERLAY または RESTORE パラメータの前に PARALLEL を指定する必要があります。
2. PARALLEL パラメータは、Windows プラットフォームでは使用できません。
3. ADABCK DUMP か ADABCK EXU_DUMP の出力を名前付きパイプに渡せば、ADABCK RESTORE または ADABCK OVERLAY の入力として直接使用できるようになるので、あるデータベースから別のデータベースにデータベースまたはファイルをコピーできます。
4. PARALLEL パラメータでは、READ_CHECK 機能のパフォーマンスは向上しません。
5. データを異なるエンディアンモードのコンピュータにリストアする場合、DUMP 処理に PARALLEL=MULTI_PROCESS を使用することは推奨されません。異なるエンディアンモードのコンピュータ上で PARALLEL=MULTI_PROCESS で作成されたバックアップファイルは、RESTORE 処理で拒否されます。

READ_CHECK

```
READ_CHECK
```

この機能は、Adabas バックアップコピーの読み取りの可能性（つまり、パリティエラーが存在しないこと）と完全性のチェックをします。これらのチェックは、ダンプファイルがこのユーティリティの RESTORE または OVERLAY 機能によるデータベースまたはファイルのリストアに使用できることを保証するために行われます。

RESTORE

```
RESTORE = {*|(number[-number][,number[-number]]...)}
          [,FMOVE [=(number [,number [-number]]...)]]
          [,FORMAT = (keyword [,keyword] ) ]
          [,NEW_DBID = number]
          [,RENUMBER = (number[-number] [,number [-number]]...)]]
          [,REPLICATION]
```

このパラメータは、リスト内の数値によって指定されるファイルをファイルレベルでリストアするものです。LOB ファイルが指定された場合は無視されますが、すべての基本ファイルに割り当てられたLOBファイルはリストアされます。リストア対象ファイルは、データベース内にロードしないでください。* は、データベースレベルでリストアを行うことを意味します。この場合、ファイルがすでにデータベース内にロードされていると、暗黙的に削除されるか、同じファイル番号を持つダンプ内のファイルが代わりに使用されるので、データベースコンテナファイルに対する排他制御が必要です。

他のファイルへの参照整合性制約がある場合でも、指定されたファイルのみがリストアされます。これらの参照整合性制約は削除されます。

注意:

1. RESTORE=* を使用するには、ダンプファイルが DUMP=* または EXU_DUMP=* を用いて作成されている必要があります。
2. 異なるエンディアンモードのプラットフォーム上で作成されたバックアップは、バックアップがオプション PARALLEL=MULTI_PROCESS を指定して作成されている場合はリストアされません。

FMOVE [=(number [,number [-number]]...)]

このキーワードを指定すると、ADABCK は、リストアするファイルまたは指定サブセットをバックアップと同じブロック範囲にリストアするのではなく、割り当て直します。このキーワードを使用すると、ファイルエクステントの数が可能な限り削減されます。

FORMAT = (keyword [,keyword])

キーワード ASSO と DATA のどちらか、または両方を指定できます。このパラメータは、アソシエータブロックやデータストレージブロックのフォーマットに使用します。ファイルレベルでリストアする際は、ファイルのエクステントの未使用エリアに含まれるブロックのみがフォーマットされます。

NEW_DBID = number

このパラメータは、リストアするデータベースの ID を変更するために使用します。データベース全体をリストアする場合にしか指定できません。

新しい ID を使用すると、アクティブなデータベースのバックアップコピーを別セットのコンテナファイルにリストアできます。新しい ID をアクティブなデータベースと同一にすることはできません。

このパラメータを省略した場合、データベース ID は変更されません。

REPLICATION

パラメータ REPLICATION は、Adabas を搭載した Adabas Event Replicator（Adabas レプリケーション）を使用している顧客にのみ関係します。

このパラメータは、Adabas に ADABCK を使用する場合に指定する必要があります（Adabas レプリケーションの初期状態処理）。このパラメータを指定する場合、Adabas ファイルが自動的にレプリケーションターゲットファイルとしてマークされます。

詳細については、ADAOPR CHANGE_REPLICATION_STATUS を参照してください。

RESTORE/OVERLAY の例

例 **1**

データベース全体が複数のバックアップデバイスから並行してリストアされます。データベースのバックアップのみを処理できます（DUMP=* または EXU_DUMP=* で作成されたバックアップ）。DUMP/EXUDUMP の例 1 のバックアップを使用します。ニュークリアスは非アクティブである必要があります。

```
adabck db=34 parallel=multi_process restore=\*
%ADABCK-I-STARTED,      30-OCT-2015 11:13:24, <version number>
%ADABCK-I-DBOFF, database 34 accessed offline

Restore database 34 dumped on 30-OCT-2015 11:10:29

Database 34, GENERAL-DATABASE

File     1, CHECKPOINT-FILE , loaded on  4-SEP-2014 13:52:43
File     2, SECURITY-FILE   , loaded on  4-SEP-2014 13:52:43
File     3, USER-DATA-FILE  , loaded on  4-SEP-2014 13:52:43
File     4, Update-log      , loaded on 17-SEP-2014 14:44:19
File     9, EMPLOYEES       , loaded on  8-OCT-2008 17:59:40
File    14, miscellaneous   , loaded on 11-JUN-2015 13:22:19
File    17, Timezone        , loaded on 19-SEP-2014 11:44:42
File    19, LARGE           , loaded on  2-SEP-2014 15:37:18
File    30, FILE-30         , loaded on 31-MAR-2015 13:40:33
File    51, PCA24SYSF1      , loaded on 14-APR-2014 16:55:22
File    80, P295170         , loaded on  4-AUG-2015 12:42:43
File    91, ADAOS-2544      , loaded on  8-APR-2015 13:19:27
File    95, P299255         , loaded on 20-MAR-2014 11:35:30
File    98, ADAOS-4591      , loaded on 16-JUL-2015 10:03:16
File   101, COLLATION-TESTS , loaded on 14-APR-2014 16:47:30
File   111, TESTOPT         , loaded on 29-NOV-2011 10:34:23
File   113, lob_LB          , loaded on 23-JUN-2015 15:48:48
File   146, XMA-REPOSITORY  , loaded on 10-DEC-2014 09:39:52
File   215, ADAOS-4647      , loaded on 27-APR-2012 15:52:49
File  1009, LOBFILE of 9    , loaded on  8-OCT-2008 17:59:40
File  1080, LOBFILE of 80   , loaded on  4-AUG-2015 12:42:43
File  1113, LOBFILE of 113  , loaded on 23-JUN-2015 15:48:48
File 22111, LOBFILE of 111  , loaded on 29-NOV-2011 10:34:23


%ADABCK-I-IOCNT, 1 IOs on dataset WORK
%ADABCK-I-IOCNT, 133 IOs on dataset DATA
%ADABCK-I-IOCNT, 244 IOs on dataset ASSO
%ADABCK-I-IOCNT, 41 IOs on dataset BCKO01
%ADABCK-I-IOCNT, 34 IOs on dataset BCKO02
%ADABCK-I-IOCNT, 80 IOs on dataset BCKO03
%ADABCK-I-TERMINATED,   30-OCT-2015 11:13:26, elapsed time: 00:00:02
```

例 **2**

ファイル215がリストアされます。このファイルがデータベースに存在していない必要があります。データベースバックアップとファイルバックアップが処理されます。複数のバックアップファイルがある場合、バックアップがオプションPARALLELで作成されていても、バックアップファイルは並行して処理されません。ニュークリアスはアクティブか非アクティブのいずれかになります。

```
adabck db=34 restore=215
%ADABCK-I-STARTED,      30-OCT-2015 11:18:34, <version number>

%ADABCK-I-DBOFF, database 34 accessed offline

Restore files from database 34 dumped on 30-OCT-2015 11:10:29

Database 34, GENERAL-DATABASE

File   215, ADAOS-4647      , loaded on 27-APR-2012 15:52:49


%ADABCK-I-IOCNT, 7 IOs on dataset DATA
%ADABCK-I-IOCNT, 28 IOs on dataset ASSO
%ADABCK-I-IOCNT, 41 IOs on dataset BCK001
%ADABCK-I-IOCNT, 34 IOs on dataset BCK002
%ADABCK-I-IOCNT, 80 IOs on dataset BCK003
%ADABCK-I-TERMINATED,   30-OCT-2015 11:18:34, elapsed time: 00:00:00
```

例 **3**

91～99、51、または 11～19 のファイル番号を持つバックアップファイル内のすべての基本ファイルがリストアされます（指定したファイル範囲にない場合でも、対応する LOB ファイルが含まれます）。ファイルがすでにデータベースに存在している場合は、上書きされます。データベースバックアップとファイルバックアップが処理されます。ニュークリアスはアクティブか非アクティブのいずれかになります。

```
adabck db=34 over=\(91-99,51,11-19\)
%ADABCK-I-STARTED,      30-OCT-2015 11:38:12, <version number>
%ADABCK-I-DBON, database 34 accessed online

Overlay files dumped on 30-OCT-2015 10:51:14

Database 34, GENERAL-DATABASE

File    14, miscellaneous   , loaded on 11-JUN-2015 13:22:19
File    17, Timezone        , loaded on 19-SEP-2014 11:44:42
File    19, LARGE           , loaded on  2-SEP-2014 15:37:18
File    51, PCA24SYSF1      , loaded on 14-APR-2014 16:55:22
File    91, ADAOS-2544      , loaded on  8-APR-2015 13:19:27
File    95, P299255         , loaded on 20-MAR-2014 11:35:30
File    98, ADAOS-4591      , loaded on 16-JUL-2015 10:03:16


%ADABCK-I-IOCNT, 619 IOs on dataset DATA
%ADABCK-I-IOCNT, 1122 IOs on dataset ASSO
%ADABCK-I-IOCNT, 1195 IOs on dataset BCK001
%ADABCK-I-TERMINATED,   30-OCT-2015 11:38:13, elapsed time: 00:00:01
```

SUMMARY

```
SUMMARY
```

このパラメータは、DUMP/EXU_DUMP 機能を以前に実行して作成された Adabas バックアップコピーにあるデータベースの一般情報と物理レイアウトを表示するものです。

例

```
adabck summary
%ADABCK-I-STARTED,      30-OCT-2015 12:01:46, <version number>

Database dumped on 30-OCT-2015 11:57:04

Database 34, GENERAL-DATABASE

Summary of Database 34          30-OCT-2015 12:01:46


DATABASE NAME                    WO-DB-34
DATABASE ID                         34
MAXIMUM FILE NUMBER LOADED       22111
SYSTEM FILES                         1 (CHK),     2 (SEC),     3 (USR)
ACTUAL FILES LOADED                 23
CURRENT PLOG NUMBER                 99
CURRENT CLOG NUMBER                 10


Container  Device      Extents in Blocks    Number of      Block    Total Size
  File     Type          from          to     Blocks        Size     (Megabytes)
-------------------------------------------------------------------------------

 ASSO1     file             1      35,840       35,840      4,096         140.00
 ASSO2     file        35,841     166,400      130,560      2,048         255.00
 ASSO3     file       166,401     205,120       38,720     32,768       1,210.00
 ASSO4     file       205,121     210,240        5,120     16,384          80.00
 ASSO5     file       210,241     944,832      734,592      8,192       5,739.00

 DATA1     file             1       8,891        8,891      4,096          34.73
 DATA2     file         8,892      24,379       15,488      8,192         121.00
 DATA3     file        24,380      34,619       10,240     16,384         160.00
 DATA4     file        34,620     388,763      354,144     32,768      11,067.00

 WORK1     file             1     207,872      207,872      4,096         812.00

-------------------------------------------------------------------------------
                                                                       19,618.73
                                                                       =========
```

```
%ADABCK-I-IOCNT, 2 IOs on dataset BCK001
%ADABCK-I-TERMINATED,   30-OCT-2015 12:01:46, elapsed time: 00:00:00
```

再スタートに関する考慮事項

ADABCK には再スタート機能は備えられていません。ADABCK が異常終了した場合、最初から再実行する必要があります。

1 つ以上のファイルに対して RESTORE/OVERLAY 機能を実行しているときに処理が中断されると、RABN の割り当ては不完全になりますが、ADADBM ユーティリティの RECOVER 機能を実行すれば完全な状態に戻すことができます。データベースに対して RESTORE/OVERLAY 機能を実行しているときに処理が中断されると、そのデータベースにはアクセスできなくなります。

6 ADACLP（コマンドログレポート）

この章では ADACLP ユーティリティについて説明します。

機能概要

ADACLP ユーティリティでは、132 文字の行幅でコマンドログが出力されます。

注意: ADACLP は、ADANUC パラメータ CLOGLAYOUT=5（5 がデフォルト値）で開始されたニュークリアスセッションのコマンドログのみを処理できます。詳細については、「*ADANUC*」、「**CLOGLAYOUT**」を参照してください。

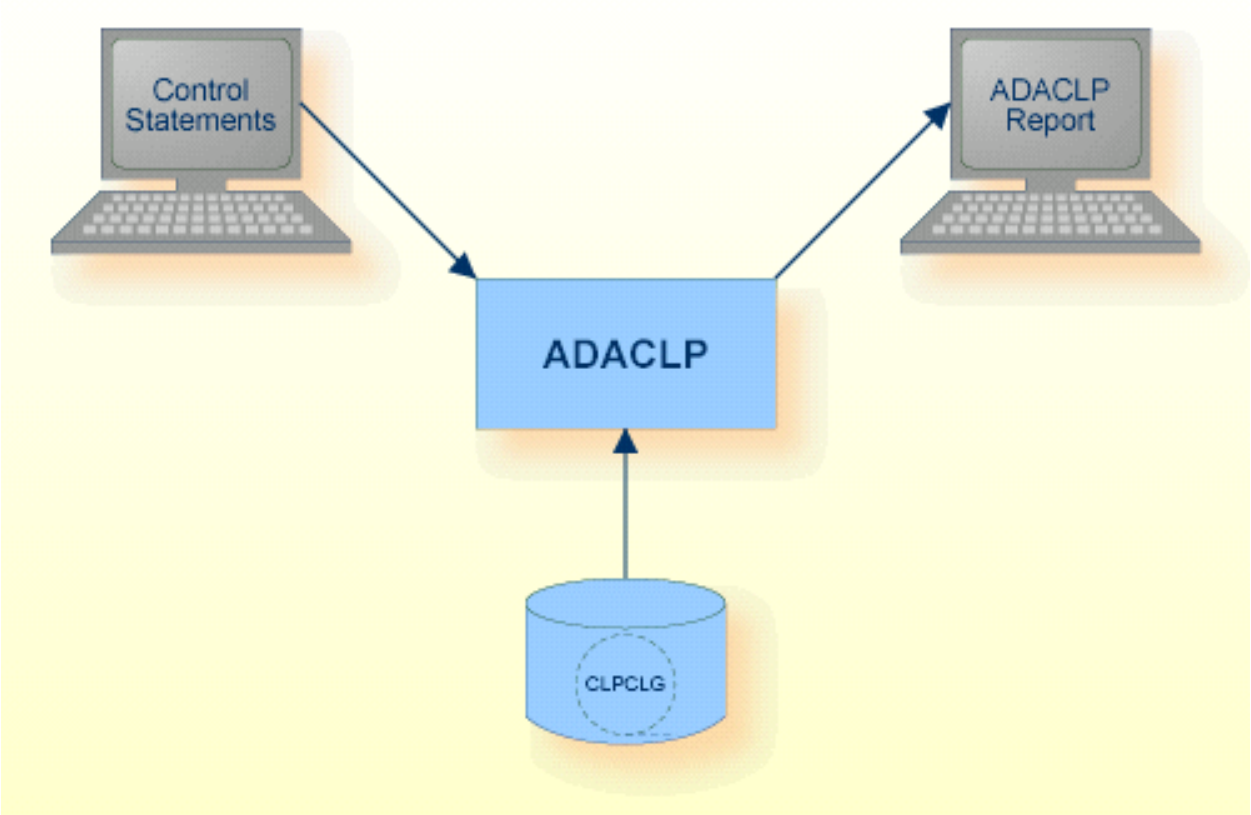
発行された Adabas コマンドごとに対して、コマンドログ内にレコードが書き込まれます。コマンドロギングは、ニュークリアスパラメータ LOGGING で Adabas を起動している間に、または ADAOPR パラメータ LOGGING でニュークリアスがすでにアクティブになっているときに有効にする必要があります。

注意: パフォーマンス上の理由で、Adabas ニュークリアスは、コマンドロギングが有効になっている場合に限り、コマンドの起動タイムスタンプを判断します。このため、コマンドロギングをオンに切り替えたとき、すでにアクティブになっているがまだ終了していない Adabas コマンドでは、コマンドの起動日付とコマンドの実行時間が表示されません。

ADACLP パラメータはいずれもコマンドログ情報のサブセットを選択します。

このユーティリティは単一機能ユーティリティです。

処理フロー



データセット	環境変数／論理名	記憶媒体	追加情報
コマンドログ	CLPCLG	ディスク、テープ（* 注参照）	ユーティリティマニュアル、ADACLP
コントロールステートメント	stdin/ SYS$INPUT		ユーティリティマニュアル
ADACLP レポート	stdout/ SYS$OUTPUT		メッセージおよびコード

注意: （*）このシーケンシャルファイルには、名前付きパイプを使用できます（OpenVMS にはない。詳細については、『*Adabas Basics*』の「*ユーティリティの使用*」を参照）。

シーケンシャルファイル CLPCLG には複数エクステントを持つことができます。複数のエクステントを持つシーケンシャルファイルの詳細については、『*Adabas Basics*』の「*ユーティリティの使用*」を参照してください。

チェックポイント

このユーティリティはチェックポイントを書き込みません。

制御パラメータ

次のコントロールパラメータを使用できます。

[NO]ADDITIONS_2

```
[NO]ADDITIONS_2
```

このパラメータは、コマンド ID の代わりにアディッション 2 フィールドを表示するのに使用されます。

デフォルトは、NOADDITIONS_2 です。

CLASS

```
CLASS = (keyword [,keyword]...)
```

このパラメータは、指定コマンドクラスに従うコマンドコードを持つログレコードを選択します。CLASS パラメータも COMMAND パラメータも指定されない場合には、すべてのレコードが選択対象となります。

CLASS パラメータと COMMAND パラメータは、相互に排他的です。

次のキーワードを使用できます。

キーワード	用途
CONTROL	open や close などの制御コマンドを選択します。
FIND	検索コマンドを選択します。
READ	読み込みコマンドを選択します。
UPDATE	更新コマンドを選択します。

例：

```
adaclp: class = find
```

コマンド S1、S2、S4、S8 および S9 のログレコードが選択されます。

CLOG

```
CLOG = (number [,number])
```

コマンドログがRAWセクションにある場合、このパラメータは必須です。コマンドログがファイルシステム内にある場合は任意です。CLOG 番号およびエクステンションカウントは指定することができます。エクステンションカウントを指定しない場合、Adabas は必要に応じて、後続のエクステントをオープンします。エクステンションカウントを指定すると、指定されたエクステントだけが処理されます。

注意: このパラメータは UNIX プラットフォームにのみ適用されます。

COMMAND

```
COMMAND = (keyword [,keyword]...)
```

このパラメータは、キーワードによって指定されたAdabasコマンドコードを持つログレコードを選択します。10個までのキーワードを定義できます。COMMANDパラメータもCLASSパラメータも指定しない場合には、全レコードが選択対象となります。

COMMAND および CLASS は相互に排他的です。

すべての有効な Adabas コマンド（A1～S9）をキーワードとして使用できます（詳細については『*コマンドリファレンス*』を参照）。

DATE

```
DATE = ([absolute-date] [,[absolute-date]])
```

このパラメータは、任意の日付文字列によって指定された範囲にあるログレコードを選択します。指定する日付文字列は、次の絶対日付および時刻の形式をとらなければいけません。

```
dd-mmm-yyyy[:hh:mm:ss]
```

日付と時刻の仕様の上位桁のゼロは、省略される可能性があります。指定されていない数字は、0 に設定されます（例：28-jul-2012 は 28-jul-2012:00:00:00 と同じです）。

デフォルトでは、すべてのログレコードが選択されます。

例：

```
adaclp: date = 8-aug-2012
```

8-AUG-2012 00:00:00 に記録されたログレコードが選択されます。

```
adaclp: date = (8-aug-2012:12,)
```

8-AUG-2012 12:00:00 以後に記録された全ログレコードが選択されます。

```
adaclp: date = (,8-aug-2012:12:34)
```

8-AUG-2012 12:34:00 までに記録された全ログレコードが選択されます。

DBID

```
DBID = number
```

このパラメータは、使用対象となるデータベースを選択するためのものです。CLOG を RAW デバイスで使用するときには、このパラメータを指定しなければなりません。

注意: このパラメータは UNIX プラットフォームにのみ適用されます。

DISPLAY

```
DISPLAY = (keyword [,keyword]...)
```

このパラメータは、コマンドログからの各種情報を表示するのに使用されます。次の各種パラメータを持つキーワードが使用可能です。ニュークリアスセッション中に対応するデータが記録された場合だけ、これらのキーワードについての情報が表示されます。

キーワード	説明
CB	■コマンドログレコード番号 ■コマンドの開始日時 ■コマンドの期間（マイクロ秒単位） ■対応する OP コマンドで指定されたユーザー ID ■ノード ID ■ログイン ID、ES_ID が指定されている場合は環境固有の ID ■コントロールブロックの選択したフィールド ■X 列の「*」は、ユーティリティまたは排他的なファイルの使用を示す ■コマンドを処理したスレッド

キーワード	説明
	■ I/O 統計 **注意:** バージョン 6.3 SP1 より前のバージョンで作成されたコマンドログでは、コマンドの継続時間がミリ秒単位で表示されます。
FB	フォーマットバッファを表示します。
FULL_CB	コントロールブロックの全フィールドを表示します。DISPLAY=CB のときに表示される他の情報はここでは表示されません。
IB	ISN バッファを表示します。
IO	I/O リストを表示します。
RB	レコードバッファを表示します。
SB	サーチバッファを表示します。
STATISTICS	選択レコードのコマンドの統計を表示します。
VB	バリューバッファを表示します。

デフォルトは DISPLAY = CB です。

ES_ID

```
ES_ID [ = number]
```

このパラメータによってログイン ID の代わりに環境固有の ID が表示されます。

数値が指定すると、指定した環境固有の ID（プロセス ID）に対する情報を備えたレコードだけが選択されます。

デフォルトでは、すべてのレコードが選択されます。

FILE

```
FILE = (number [- number] [,number [- number]]...)
```

このパラメータは、コマンド付きのログレコードの中から、番号または番号の範囲で指定したファイルを参照するものを選択します。最大 20 ファイルを指定することができます。

デフォルトでは、すべてのレコードが選択されます。

[NO]HEXADECIMAL

```
[NO]HEXADECIMAL
```

このパラメータを HEXADECIMAL に設定すると、レコードバッファとバリューバッファが 16 進形式で表示されます（DISPLAY=RB または DISPLAY=VB を指定した場合） 。

デフォルトは、NOHEXADECIMAL です。

LOGIN_ID

```
LOGIN_ID = string
```

このパラメータは、指定したログイン ID を持つレコードをすべて選択します。

デフォルトでは、すべてのレコードが選択されます。

NODE_ID

```
NODE_ID = string
```

このパラメータは、指定されたノードからのログレコードを選択します。

ADAOPR の処理時にパラメータ DISPLAY=UQ で示されたノード ID を使用しなければなりません。

このパラメータは、Entire Net-Work が導入されている場合にのみ有効です。

PAGE

```
PAGE = number
```

このパラメータは、プリント出力に使用されるページサイズを行数で定義します。

デフォルトは 59 行です。

RECORDS

```
RECORDS = number [-number]
```

このパラメータは、指定された範囲のログレコード番号に該当するログレコードを選択します。ログレコード番号は、ログが稼働し始めた後に、1 から開始されます。

デフォルトでは、すべてのレコードが選択されます。

RESPONSE

```
RESPONSE = (number [- number] [,number [- number]] ... )
```

このパラメータは、指定されたレスポンスコードまたはレスポンスコードの範囲に該当するレコードを選択します。

USER_ID

```
USER_ID = string
```

このパラメータは、"string" で指定されたユーザー ID を持つレコードを選択します。

デフォルトでは、全ユーザーが対象です。

例

```
user_id = *adarep
```

ADAREP ユーティリティから発行されたコマンドを示す全レコードが選択されます。

WIDTH

```
WIDTH = number
```

1 行の出力の幅を設定します。有効な値は 80 または 132 です。

デフォルトは、132 です。

複数選択条件の指定

複数の選択条件を指定する場合、それらは論理 AND で結合しなければなりません。

```
command = l3, file = 5
```

これは、ファイル 5 上の L3 コマンドがすべて選択されることになります。

7 ADACMP（データの圧縮）

この章では ADACMP ユーティリティについて説明します。

機能概要

圧縮ユーティリティ ADACMPは、ユーザーの未加工データを一括更新ユーティリティ ADAMUP が使用できる形式に圧縮するものです。

このユーティリティに対する入力データは、シーケンシャルファイルでなければいけません。LOB フィールドの値は別のファイルでも指定できます。

入力データの論理構造と特性は、フィールド定義テーブル（FDT）によって記述されます。これらのステートメントは、レベル番号、フィールド名、標準長および形式を、その他フィールドに割り当てられるすべての定義オプション（ディスクリプタ、ユニークディスクリプタ、マルチプルバリューフィールド、空値省略、固定ストレージ、ピリオディックグループ）とともに指定します。データベースのファイルのレイアウトおよび入力データの属性については、『*管理*マニュアル』の「*FDT のレコード構造*」を参照してください。

オプション SY（システム生成）が指定されていない入力レコードの各フィールドが圧縮されます。圧縮は、英数字フィールドの後方の空白および数値フィールドの上位桁のゼロを除去します。アンパック形式およびパック形式のフィールドは正しいデータかチェックされます。固定ストレージオプションで定義されたフィールドは圧縮されません。ユーザー出口が提供されているので、ユーザー作成ルーチンを使用して各入力レコードの追加編集ができます。

システム生成フィールドは、ADACMP パラメータ SYFINPUT に指定されたキーワードに応じて、再生成または圧縮解除されます。

このユーティリティは、次の 3 タイプの出力ファイルを作成します。

- 圧縮データ
- ディスクリプタ値
- エラーのあるレコード

すべてのディスクリプタ値の大きさは、実行の最後にリストされます。

エラーファイルにレコードが書き込まれると、ユーティリティはゼロ以外のステータスで終了します。

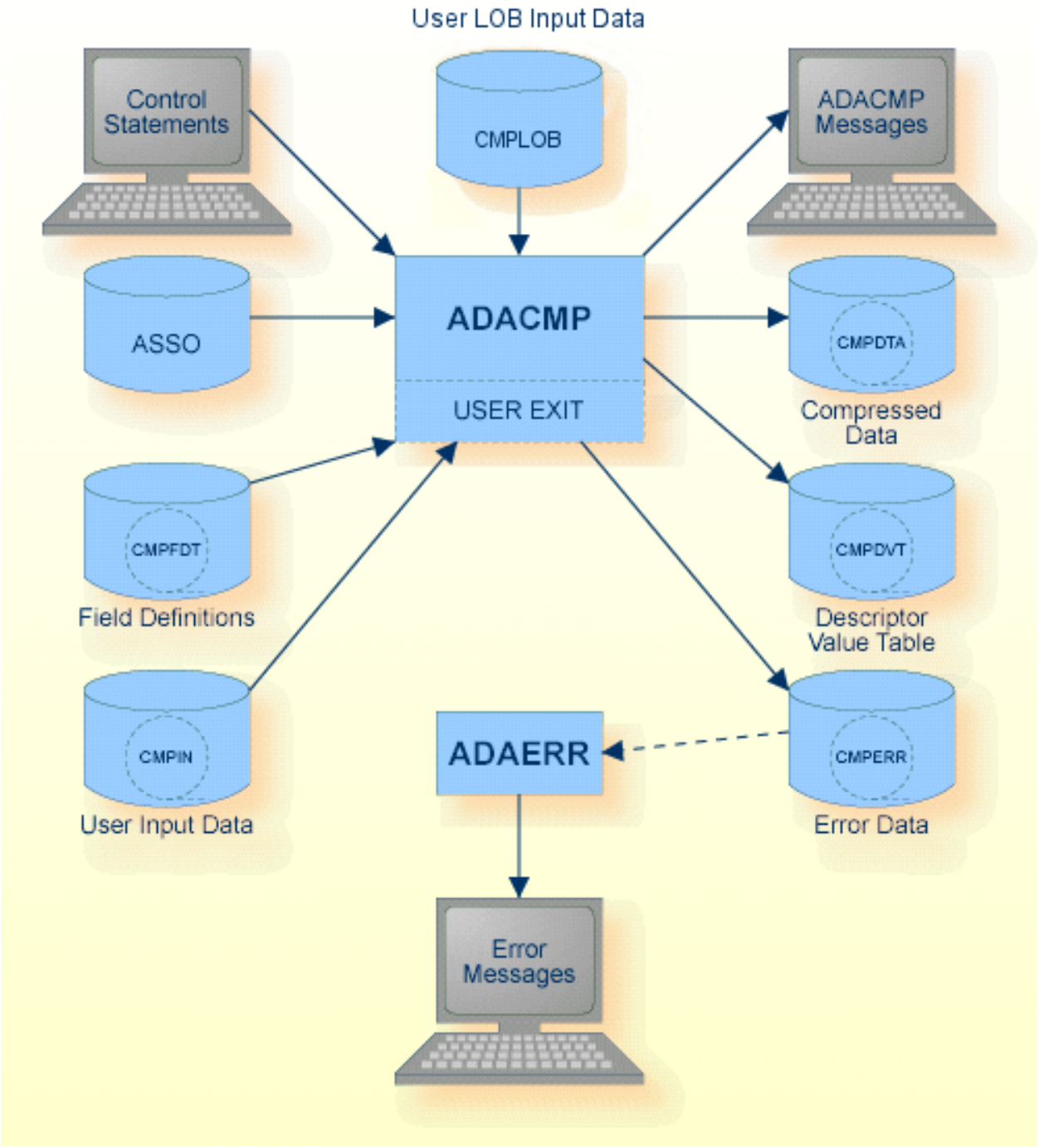
注意: ICU 3.2 照合ディスクリプタを含むファイルにデータを追加する場合は、注意してください。

- FDT をパラメータ DBID および FILE とともに指定すると、FDT はデータベースから変更されずに取得されます。これは、ICU バージョンが 3.2 のままであることを意味します。データをファイルに追加できますが、ICU のバージョンは 3.2 のままになります。

- CMPFDT ファイルから FDT を取得すると、CMPFDT ファイルから新しい FDT が作成され、ICU バージョンが 5.4 に設定されます。これは、データが空の場合、および NEW_FDT オプションを指定した場合にのみ、データをファイルに追加できることを意味します。使用される ICU のバージョンは 5.4 です。

このユーティリティは単一機能ユーティリティです。

処理フロー



シーケンシャルファイルCMPDTA、CMPDVTおよびCMPERRは複数のエクステントを持つことができます。エクステントが複数あるシーケンシャルファイルの詳細については、『*Adabas Basics*』の「ユーティリティの使用」に記載されている「*Adabas* シーケンシャルファイル」の

「*Adabas シーケンシャルファイルの複数エクステント*」を参照してください。CMPLOB はディレクトリであり、データベースに LOB 値として保存されている可能性があるファイルが含まれています。

データセット	環境変数／論理名	記憶媒体	追加情報
アソシエータ	ASSOx	ディスク	
圧縮データ	CMPDTA	ディスク、テープ（* 注参照）	ADACMP の出力
ディスクリプタバリューテーブル	CMPDVT	ディスク、テープ（* 注参照）	ADACMP の出力
拒否データ	CMPERR	ディスク、テープ（* 注参照）	ADACMP の出力
入力データ FDT	CMPFDT	ディスク、テープ（* 注参照）	ユーティリティマニュアル
ユーザー入力データ	CMPIN	ディスク（* 注参照）	ユーティリティマニュアル
ユーザー LOB 入力データ	CMPLOB	ディスク	ユーティリティマニュアル
ADACMP コントロールステートメント	stdin/SYS$INPUT		ユーティリティマニュアル
ADACMP メッセージ	stdout/SYS$OUTPUT		メッセージおよびコード

注意: （*）名前付きパイプをこのシーケンシャルファイルに使用できます（詳細については、『*管理マニュアル*』の「*ユーティリティの使用*」に記載されている「*Adabas シーケンシャルファイル*」、「*名前付きパイプの使用*」を参照）。

SINGLE_FILE オプションを設定した場合、ディスクリプタバリューテーブル（DVT）と圧縮ユーザーデータはともに論理名 CMPDTA に書き込まれます。

チェックポイント

このユーティリティはチェックポイントを書き込みません。

制御パラメータ

次のコントロールパラメータを使用できます。

```
     DBID = number
D    [NO]DST
     FDT
     FIELDS {uncompressed_field_definition | FDT}...[END_OF_FIELDS | . ]
     FILE = number
D    [NO]LOBS
D    [NO]LOWER_CASE_FIELD_NAME
D    MAX_DECOMPRESSED_SIZE = number [K|M]
D    MUPE_C_L = {1|2|4}
D    [NO]NULL_VALUE
D    NUMREC = number
D    RECORD_STRUCTURE = keyword
     SEPARATOR = character | \character
D    [NO]SHORT_RECORDS
D    [NO]SINGLE_FILE
     SKIPREC = number
D    SOURCE_ARCHITECTURE = (keyword[,keyword][,keyword])
D    SYFINPUT = keyword
D    TZ {=|:} [timezone]
D    [NO]USEREXIT
D    [NO]USERISN
D    WCHARSET = char_set
```

DBID

```
DBID = number
```

このパラメータは、FILE パラメータによって指定されるファイルが存在するデータベースを選択するものです。

[NO]DST

```
[NO]DST
```

DSTパラメータは、オプションTZを使用して日付／時刻フィールドに夏時間インジケータが提供されている場合に必要です。夏時間インジケータは、2 バイトの整数値（フォーマット F）として日付／時刻値の後に追加する必要があります。実際の時間（通常は 0 または 3600）を取得するために、この値には、標準時間に追加する秒数が含まれます。

DST パラメータが指定されていない場合、時刻が標準時刻に戻るまで、時間の時刻値を定義できません。

デフォルトは NODST です。

注意:

1. FIELDS パラメータが指定されている場合、DST パラメータは無視されます。この場合、サマータイムインジケータの付いたフィールドに D 要素を指定する必要があります。
2. フォーマット F のサマータイムインジケータに出力できない文字が含まれているため、DST パラメータは、RECORD_STRUCTURE = NEWLINE_SEPARATOR パラメータと互換性がありません。

例：

DT フィールドには次の定義があります：1,DT,8,P,DT=E(DATE_TIME),TZ

このフィールドには、次の値を指定する必要があります。

- 編集マスク DATE_TIME に対応する 8 バイトパック値としてのローカル日付／時刻値
- 夏時間インジケータ。通常は、2 バイトの固定小数点値として標準時間は 0、夏時間は 3600

```
Case 1 (DT has a date/time value with daylight saving time): 0x02009102502300000E10
Case 2 (DT has a date/time value with standard time): 0x0200910250230000000
```

FDT

```
FDT
```

このパラメータが最初のパラメータとして指定されている場合、または [NO]LOWER_CASE_FIELD_NAMES の後に 2 番目のパラメータとして指定されている場合、ADACMP はシーケンシャルファイル CMPFDT に含まれている FDT 情報を読み取り、FDT を表示します。

注意: または、FDT の代わりに DBID と FILE を使用して、最初のパラメータとして指定することも、[NO]LOWER_CASE_FIELD_NAMES（これは DBID と FILE の前に許可されます）の後に2番目のパラメータとして指定することもできます。この場合、ファイルの FDT が圧縮のベースとして使用されます。

FDT パラメータは複数回指定できますが、FDT または DBID と FILE パラメータを指定して、圧縮に FDT を使用するようにすでに決定している場合は、FDT パラメータを再度指定すると、FDT のみが表示され、FDT は CMPFDT で上書きされません。

FIELDS

```
FIELDS {uncompressed_field_definition | FDT}...[END_OF_FIELDS | . ]
```

このパラメータは、FDT のフィールドのサブセットおよび、それらのフォーマットと長さを指定するために使用します。FDT で指定されているすべてのフィールドが入力レコードに含まれている必要はありません。つまり、別のフォーマットか長さでフィールドを指定できるということです。構文と使用方法はフォーマットバッファの場合とほぼ同じです。ただし、非圧縮レコードに LOB 値自体ではなく、LOB 値を含むファイルの名前が含まれている場合に、（LOB 参照の）R要素も指定できる点だけが異なります。詳細については、『*管理マニュアル*』の「*データのロードとアンロード*」の「*非圧縮データフォーマット*」を参照してください。

指定リストの入力時に、圧縮解除するファイルの FDT を表示する FDT 機能を使用できます。指定リストは END_OF_FIELDS または「.」を入力することで終了または中断できます。「.」オプションは暗黙の END_OF_FIELDS であり、フォーマットバッファ構文と互換性があります。FIELDS または END_OF_FIELDS は常にそれ自体を行に入力する必要がありますが、「.」はそれ自体を行に入力することも、フォーマットバッファ要素の末尾に入力することも可能です。

END_OF_FIELDS パラメータを使用してフィールド定義を消去する場合、LOWER_CASE_FIELD_NAMES パラメータを使用するときに、このパラメータを大文字で指定する必要があります。また、LOWER_CASE_FIELD_NAMESを使用する場合は、FDTパラメータも大文字で指定する必要があります。

FILE

```
FILE = number
```

このパラメータは、FDT情報が読み取られるファイルを指定するものです。このパラメータは、DBID パラメータの後にしか指定できません。

[NO]LOWER_CASE_FIELD_NAMES

```
[NO]LOWER_CASE_FIELD_NAMES
```

LOWER_CASE_FIELD_NAMES が指定されている場合、Adabas フィールド名は、大文字に変換されません。NOLOWER_CASE_FIELD_NAMESが指定されている場合、Adabasフィールド名は、大文字に変換されます。デフォルトは、NOLOWER_CASE_FIELD_NAMES です。

FDT の小文字フィールド名を大文字に変換しない場合は、FDT パラメータの前に、最初のパラメータとしてこのパラメータを指定する必要があります。FIELDSパラメータの小文字フィールド名を大文字に変換しない場合は、FIELDSパラメータの前に、このパラメータを指定する必要があります。

注意: CMPFDTファイルに対してLOWER_CASE_FIELD_NAMESパラメータが指定されている場合、ファイル全体に対する大文字変換は行われません。フィールドフォーマットやフィールドオプションの小文字は、FDT 構文エラーの原因となります。この問題は、FIELDS パラメータの小文字でも発生します。

[NO]LOBS

```
[NO]LOBS
```

このパラメータでは、圧縮データをデータベースにロードした後で、LAフィールドとLBフィールドの値を LOB ファイルに保存するかどうかを指定します。

- パラメータDBIDとファイル番号を指定した場合、このパラメータは無視され、フィールドは次の説明のように処理されます。
- パラメータ DBID とファイル番号を指定せず、LOBS を指定した場合、LA フィールドと LB フィールドのフィールド値は、フィールドがディスクリプタとして定義されていなければ、LOB ファイルでの保存用に準備されます。
- パラメータ DBID とファイル番号を指定せず、NOLOBS を指定した場合、LA フィールドと LB フィールドのフィールド値は基本ファイルでの保存用に準備されます。この場合、LA フィールドと LB フィールドのフィールド値の長さは 16381 バイト以内にして、圧縮レコードが 32 KB DATA ブロックに収まる大きさにする必要があります。

ディスクリプタであるか、派生ディスクリプタ（スーパーディスクリプタなど）の親フィールドであるLAフィールドとLBフィールドは、NOLOBSパラメータで説明したように常に処理されます。

デフォルトの動作は次のとおりです。

- パラメータ DBID とファイル番号を指定し、ファイルが対応 LOB ファイルを含む基本ファイルである場合は、LOBS がデフォルトとなります。
- パラメータ DBID とファイル番号を指定し、ファイルが対応 LOB ファイルを含む基本ファイルでない場合は、NOLOBS がデフォルトとなります。
- パラメータ DBID とファイル番号を指定しない場合は、LOBS がデフォルトとなります。

MAX_DECOMPRESSED_SIZE

```
MAX_DECOMPRESSED_SIZE = number  [K|M]
```

このパラメータは、数字の後の「K」または「M」の仕様に応じて、非圧縮レコードの最大サイズをバイト、キロバイトまたはメガバイト単位で指定します。このパラメータは、無効なCMPINファイルを可能な限り早く認識するためのものです。

デフォルトは、65536 です。これは、最大値でもあります。

注意:

1. このパラメータには、別のファイルに保存されている LOB 値のサイズが含まれません。
2. このパラメータの正確な定義は、最大非圧縮レコードに必要な入力／出力バッファのサイズです。256バイトの倍数のみが入力／出力バッファに使用されます。これにより、次の256の倍数に切り上げられる最大非圧縮レコード以上の値（先行する長さフィールドを含む）を指定する必要があります。

MUPE_C_L

```
MUPE_C_L = {1|2|4}
```

非圧縮データにマルチプルバリューフィールドまたはピリオディックグループが含まれる場合、それらには MUPE_C_L バイトの長さのバイナリのカウントフィールドが先行します。

デフォルトは 1 です。

[NO]NULL_VALUE

```
[NO]NULL_VALUE
```

標準 FDT に従って圧縮するとき、入力データに NC オプションフィールドのステータス値が与えられる場合、パラメータ NULL_VALUE が必須になります。通常、このような入力データは NULL_VALUE オプションを設定した ADADCU によって生成されます。

デフォルトは NONULL_VALUE です。

例

フィールド AA の FDT 内での定義：1, AA, 2, A, NC

ケース 1（AA が NULL 値を持っていない場合）：入力レコード（16 進） = 00004142

ケース 2（AA が NULL 値を持っている場合）：入力レコード（16 進） = FFFF2020

NUMREC

```
NUMREC = number
```

このパラメータは、処理される入力レコード数を指定するものです。このパラメータが省略された場合は、入力ファイルに含まれているすべての入力レコードが処理されます。

このパラメータの使用は、入力データファイルに大量レコードがある場合、ADACMP の初期実行時の使用に適しています。一度 ADACMP を実行しておくと、データ定義エラーや不当な入力データによって大量のエラーレコードが発生する場合に、すべてのレコードに対する不要な処理を避けることができます。

また、このパラメータは、テスト用に小さいファイルを作成する場合にも役立ちます。

RECORD_STRUCTURE

```
RECORD_STRUCTURE = keyword
```

このパラメータは、環境変数 CMPIN に指定した入力ファイルに使用されるレコード区切りのタイプを指定するものです。次のキーワードを使用できます。

キーワード	説明
ELENGTH_PREFIX	CMPIN ファイルのレコードは、2 バイトの長さフィールド（フィールド長にはフィールド自体の長さは含まれない）によって区切られます。
E4LENGTH_PREFIX	非圧縮データファイル内のレコードは、4バイトの長さフィールド（長さフィールド自体の長さを除く）で区切られます。
ILENGTH_PREFIX	CMPIN ファイルのレコードは、2 バイトの長さフィールド（フィールド長にはフィールド自体の長さも含まれる）によって区切られます。
I4LENGTH_PREFIX	非圧縮データファイル内のレコードは、4バイトの長さフィールド（長さフィールド自体の長さを含む）で区切られます。
NEWLINE_SEPARATOR	CMPIN ファイルのレコードは改行文字で区切られています。このキーワードは、改行として解釈される文字がフィールド値に含まれない場合（アンパック形式のフィールド、英数字フィールド、Unicode フィールドのみが含まれる場合、および英数字フィールドと Unicode フィールドに出力可能文字のみが含まれる場合）に限って指定できます。このキーワードおよび USERISN パラメータは相互に排他的です。
RDW	CMPIN ファイルのレコードは、FTP *site rdw* オプションを使って IBM ホストから転送されたデータを含んでいます。ADACMP は *cvt_fmt* をはじめに使用しなくても、そのようなデータを処理することができます（以前のバージョンではそのようなデータの変換に、*cvt_fmt* という非サポートのツールを使っていました）。次にその例を示します。

キーワード	説明
	% ftp IBM-host ftp> binary 200 Representation type is Image ftp> site rdw 200 Site command was accepted ftp> get decomp % setenv CMPIN decomp % adacmp fdt record_structure=rdw source=(ebcdic,high)
RDW_HEADER	RDW と同じように、HEADER=YES を指定して、メインフレームで圧縮解除したデータの場合。
HEADER	HEADER=YESを指定してメインフレームで圧縮解除されたデータ（圧縮解除されたデータにブロックまたはレコード長に関する追加情報が含まれていない場合）。
VARIABLE_BLOCKED	BS2000 または IBM の可変ブロック形式。

デフォルトは ELENGTH_PREFIX です。

SEPARATOR

```
SEPARATOR = character | \character
```

このオプションを指定すると、未加工データレコードのフィールドは、指定した文字で区切られているものとして扱われます。Adabas ユーティリティにとって特別な意味を持たせる必要がなければ、文字をアポストロフィで囲む必要はありません。この場合には、それぞれのレコードの同一フィールドを別々の長さにすることができます。

FIELDSパラメータを使用して、フォーマットバッファを指定する場合、指定するフィールド名の順番は、フィールドがFDTに指定された順番と一致させなければなりません。異なった場合、不一致が生じます。

FDT にマルチプルバリューフィールド、またはピリオディックグループが含まれている場合、フォーマットバッファをFIELDSパラメータで指定する必要があります。ピリオディックグループのメンバは、ピリオディックグループインデックスと、FDT のフィールドの並び順に従って順序付ける必要があります（下の例 2 を参照）。

SEPARATORオプションを使用している入力ファイルにはバイナリデータは予期されないので、RECORD_STRUCTURE パラメータは NEWLINE_SEPARATOR に設定されます。

例 **1**

```
FDT:      1, AA, 2, U
          1, AB, 8, U
          1, AC, 2, A

CMPIN:    12;12345678;AA
          1234;5;BB

adacmp
fdt
separator=\;

or for UNIX

adacmp fdt  separator=\\\;

or

adacmp fdt  separator='\;'
```

上記の例では、2 レコードがデフォルトの FDT で圧縮されます。区切り文字はコロン（;）で、デフォルトのレコード構造は NEWLINE_SEPARATOR です。コロンはバックスラッシュで先行されていることに注意してください。そうでなければ、コメントの開始とみなされます。UNIX 環境において、直接コマンド行からパラメータを入力する場合、バックスラッシュとコロンの前に追加のバックスラッシュを付加するか、それらを引用符または二重引用符で囲む必要があります。そうしないと、特殊文字として扱われます。

例 **2**

```
FDT:      1, XX, PE
          2, AA, 8, A
          2, AB, 8, U
          1, YY, 2, A

Correct:  CMPIN:    aaaa,1,bbbb,2,yy

          Command:  adacmp fdt separator=, fields AA1,AB1,AA2,AB2,YY.
          First, the field values for the periodic group index 1 are
          specified, and then the field values for periodic group index 2.

Invalid:  CMPIN:   aaaa,bbbb,1,2,yy
          Command: adacmp fdt separator=, fields AA1-2,AB1-2,YY.
          The fields specification is invalid because the 2nd value of
          AA is specified before the 1st value of AB; you will get
          the error SEPINV.
```

上記の例では、フォーマットバッファに指定されたフィールドを持つ 1 レコードが圧縮されます。区切り文字はコンマ（,）です。

例 **3**

```
FDT:      1, AA, 8, A
          1, MA, 1, A, MU

CMPIN:    aaaa%2%A%B
          bbbb%3%C%D%E

adacmp  dbid=9  file=15  separator=%, fields "AA,MAC,1,U,MA1-N"
```

上記の例では、フォーマットバッファに指定されたフィールドを持つ 2 レコードが圧縮されます。オカレンスカウントまたはマルチプルバリューフィールド MA はそれぞれのレコードで異なります。区切り文字はパーセント（%）です。

[NO]SHORT_RECORDS

```
[NO]SHORT_RECORDS
```

SHORT_RECORDSを指定すると、空値を含んでいる非圧縮レコードの末尾のフィールドを省略することが可能です。

デフォルトは NOSHORT_RECORDS です。

フィールド全体を省略することだけが可能です。末尾の値を切り捨てることはできません。

例

英数字フィールド AA と AB を含んでいるファイルにパラメータを指定したと仮定します。

```
FIELDS
AA,20,AB,20
END_OF_FIELDS
SHORT_RECORDS
```

そのとき、次のレコードは認められます。

```
"Field AA          "
```

次のレコードは認められません。

```
"Field AA"
```

[NO]SINGLE_FILE

```
[NO]SINGLE_FILE
```

SINGLE_FILEオプションを設定すると、ADACMPは、ディスクリプタバリューテーブル（DVT）と圧縮データを別々のファイルではなく、1 つのファイル（CMPDTA）にまとめて書き込みます。

デフォルトは NOSINGLE_FILE です。

SKIPREC

```
SKIPREC = number
```

このパラメータは、圧縮を始める前にスキップされるレコード数を指定するものです。

SOURCE_ARCHITECTURE

```
SOURCE_ARCHITECTURE = ( keyword [,keyword [,keyword] ] )
```

このパラメータは、入力データレコードの形式（文字セット、浮動小数点フォーマット、およびバイト順）を指定するものです。次のキーワードを使用できます。

キーワードグループ	有効なキーワード
文字セット	ASCII EBCDIC
浮動小数点フォーマット	IBM_370_FLOATING IEEE_FLOATING VAX_FLOATING
バイト順	HIGH_ORDER_BYTE_FIRST LOW_ORDER_BYTE_FIRST

キーワードグループのキーワードを指定しない場合、このキーワードグループのデフォルトは、ADACMP を実行しているマシンのアーキテクチャに対応するキーワードになります。

注意: FDT は常に ASCII フォーマットの入力です。

例

圧縮される入力レコードがIBM形式の場合は、ユーザーは次のように指定する必要があります。

```
SOURCE_ARCHITECTURE = (EBCDIC, IBM_370_FLOATING, HIGH_ORDER_BYTE_FIRST)
```

SYFINPUT

```
SYFINPUT = keyword
```

このパラメータは、システム生成フィールドの圧縮に使用する入力を指定します。次のキーワードを使用できます。

キーワード	説明
SYSTEM	システム生成フィールド値は、ADACMP でシステムによって再生成されます。
USER	システム生成フィールド値は、圧縮解除されたファイルから取得されます。

デフォルトは SYFINPUT = USER です。

TZ

```
TZ  {=|:} [timezone]
```

指定されたタイムゾーンは、Olson データベース（**https://www.iana.org/time-zones**）として知られているタイムゾーンデータベースに含まれている有効なタイムゾーン名にする必要があります。タイムゾーンが指定されている場合、このタイムゾーンは、オプションTZを含む日付／時刻フィールドのタイムゾーン変換に使用されます。

デフォルトは、UTC です。これは、オプション TZ を含む日付／時刻フィールドに内部的に保存するために使用されます。変換する必要はありません。

空の値を指定する場合、日付／時刻フィールドが正しいことを確認するチェックマークは付いていません。

注意: タイムゾーン名はファイル名になります。プラットフォームに応じて、これらのファイル名は大文字小文字を区別する場合としない場合があります。また、タイムゾーン名も、プラットフォームに応じて大文字／小文字を区別する場合としない場合があります。

例：

```
tz:Europe/Berlin
```

これは、すべてのプラットフォームに適用されます。

```
TZ=Europe/Berlin
```

この仕様により、TZ は大文字の EUROPE/BERLIN に変換されます。これは、Windows では、ファイル名の大文字／小文字が区別されないため適用されます。ただし、Unix では、Unix ファイル名の大文字／小文字は区別されるため適用されません。

[NO]USEREXIT

```
[NO]USEREXIT
```

このオプションは、ユーザー出口が使用されるかどうかを指定します。USEREXIT を指定した場合、環境変数 ADAUEX_6／論理名 ADABAS$USEREXIT_6 は、ロード可能なユーザー作成ルーチンをポイントする必要があります。

詳細については、『*管理マニュアル*』の「*ユーザー出口とハイパー出口*」に関する説明を参照してください。

デフォルトは NOUSEREXIT です。

[NO]USERISN

```
[NO]USERISN
```

このパラメータを USERISN に設定すると、入力ファイルの各レコードに対する ISN は、ユーザーによって割り当てられます。

USERISN を指定した場合、ユーザーは、各レコードに割り当てる ISN を各データレコードの直前に先行する 4 バイトの 2 進数として指定します。

ISN は、任意の順番で割り当てることはできますが、（同一ファイル内で）ユニークでなければいけません。ISN は、ファイルに指定された最大レコード数（MAXISN）を超えないようにする必要があります。詳細については、ファイル定義ユーティリティ「*ADAFDU*」を参照してください。

ADACMP は、ISN の一意性のチェックや ISN が MAXISN を超えるかどうかのチェックは行いません。これらのチェックは、一括更新ユーティリティ ADAMUP によって実行されます（エラーが検出されると、ADAMUP の実行は終了し、エラーメッセージが出力されます）。

このパラメータを NOUSERISN に設定すると、ISN は Adabas によって割り当てられます。

デフォルトは NOUSERISN です。

WCHARSET

```
WCHARSET = char_set
```

このパラメータは、**http://www.iana.org/assignments/character-sets** に記載されているエンコード名に基づいて、非圧縮ファイルで使用するデフォルトエンコードを指定します。このサイトに記載されているほとんどの文字セットは、ICU（Adabas国際化サポートで使用）によってサポートされています。

デフォルトは UTF-8 です。

出力

ADACMP ユーティリティは、次の 3 つのファイルを出力します。

1. 圧縮データ
2. ディスクリプタ値
3. エラーのあるレコード

圧縮されたデータレコード

ADACMP が処理、修正および圧縮したデータレコードは、FDT 情報と一緒にシーケンシャルファイルに出力されます。このファイルは、一括更新ユーティリティ ADAMUP に対する入力として使用されます。

出力ファイルにレコードがない場合（入力ファイルにレコードがない場合やすべてのレコードがエラーになった場合）にも、その出力を一括更新ユーティリティ ADAMUP に対する入力として使用できます。

ディスクリプタバリューテーブルファイル

このファイルの内容は、ディスクリプタバリューテーブル（DVT）です。

圧縮データレコードとディスクリプタバリューテーブルは、SINGLE_FILE オプションが指定されると 1 つのファイルに書き込まれます。

拒否されたデータレコード

ADACMP によってエラーになったすべてのレコードは、ADACMP エラーファイルに書き込まれます。このエラーファイルの内容は、ADAERRユーティリティを使用して表示します。レコードに出力不能文字が含まれているため、標準オペレーティングシステム出力ユーティリティを使用してエラーファイルを出力しないでください。

詳細については、「*ADAERR*」ユーティリティの説明を参照してください。

レポート

CMPFDT を入力として使用している場合には、ADACMP レポートの先頭に、入力したフィールド定義が出力されます。構文エラーのあるすべてのステートメントは、そのステートメントのすぐ後にメッセージをが表示されます。

データ定義ステートメントの出力の後には、ディスクリプタの概要、処理された入力レコード数、エラーになった入力レコード数、および圧縮された入力レコード数が出力されます。

再スタートに関する考慮事項

ADACMP は再スタート機能を備えていません。中断した ADACMP の実行は、最初から再スタートしなければなりません。

ADACMP は、データベースを変更しないので、ADACMP の再スタート前にデータベースステータスに関して考慮する必要はありません。

8 ADACVT（以前のバージョンからデータベースを変換します）

この章では「ADACVT」ユーティリティについて説明します。

機能概要

ユーティリティ ADACVT は、既存のデータベースをその場所で変換します。変換は以下で説明するように実行されます。

ADACVT を使用すると、双方向にデータベースを変換することができます。

- 以前のバージョンから現在のバージョンへ。
- 現在のバージョンから以前のバージョンへ。

データベースを変換する場合

- ニュークリアスはアクティブにしてはいけません。または、
- ニュークリアスで AUTORESTART を保留にしてはいけません。
- ユーティリティコントロールブロック（UCB）にニュークリアスのユーティリティエントリがあってはいけません。

データベースを変換する前に、次の手順を実行することを強く推奨します。

- ADABCK を使用してデータベースのダンプを取得します。
- ADAVFY'S FIELD および INDEX 機能を使用して、データベースの一貫性を確認します。

注意: データベースの確認は、データベースのダンプを取得する前または後に行うことができます。

新しい Adabas バージョンは、データベースが ADABAS.INI ファイルで登録されている場合に最適に機能します。データベースが登録されていないが、対応する DBxxx.INI ファイルが使用できる場合は、ADACVT がデータベースを ADABAS.INI ファイルに追加します。

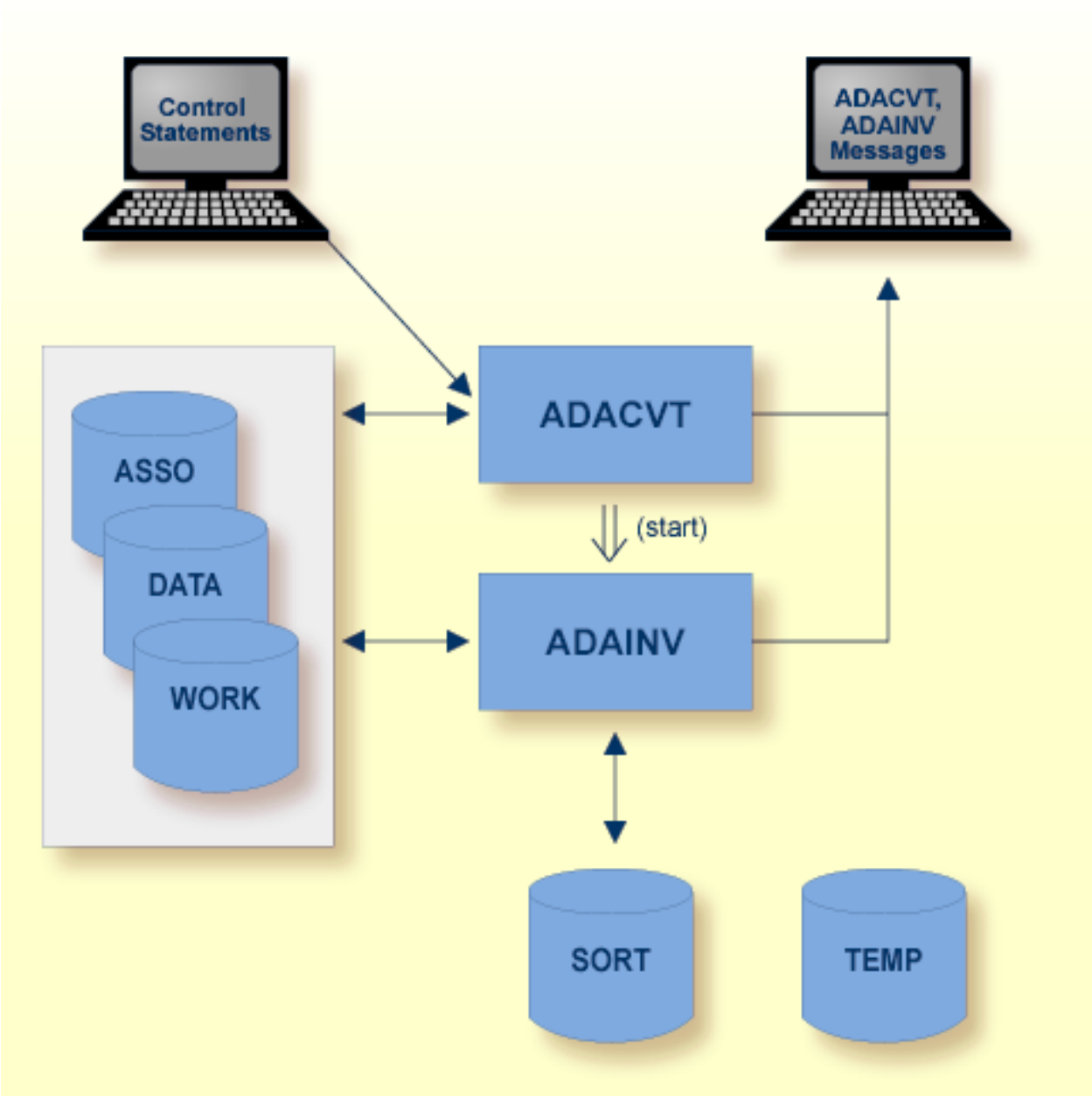
変換中に、新しい構造レベルと AUTOSTART オプションが ADABAS.INI に適合されます。

注意:

1. Adabas バージョン 6.6 で導入された機能を使用した場合は、下位変換はできません。新しい機能を使用して行われた修正は、下位変換を試みる前に削除する必要があります。
2. ICU バージョン 5.4 で作成された照合ディスクリプタを含むデータベースの下位変換を実行する場合は、ADACVT の実行に成功した後で、これらの照合ディスクリプタに対してターゲットバージョンの ADAINV を指定し、ADAINV REINVERT を実行する必要があります。

このユーティリティは単一機能ユーティリティです。

処理フロー


Control Statements
ADACVT, ADAINV Messages
ADACVT
(start)
ADAINV
ASSO
DATA
WORK
SORT
TEMP

データセット	環境 変数／ 論理名	記憶 媒体	追加情報
アソシエータ	ASSOx	ディスク	
データストレージ	DATAx	ディスク	
ソートストレージ	SORTx TEMPLOCx	ディスク	Adabas Basics マニュアル、一時ワークスペース
コントロールステートメント	stdin/ SYS$INPUT		
ADAINV メッセージ	stdout/ SYS$OUTPUT		メッセージおよびコード
一時ストレージ	TEMPx	ディスク	

チェックポイント

正常に完了すると、ADACVT が SYNP チェックポイントを書き込みます。

ADACVT 制御パラメータ

次のコントロールパラメータを使用できます。

```
    M    DBID = number

    M    CONVERT = {V630|V640|V650|V660}
```

DBID

```
DBID = number
```

このパラメータは変換するデータベースを選択します。

CONVERT

```
CONVERT =  {V630|V640|V650|V660}
```

このパラメータは、変換先バージョンの構造を指定するものであり、次の値を使用できます。

V630
バージョン 6.6 からバージョン 6.3 へデータベースを変換します。

V640
バージョン 6.6 からバージョン 6.4 へデータベースを変換します。

V650
バージョン 6.6 からバージョン 6.5 へデータベースを変換します。

V660
バージョン 6.1、バージョン 6.2、バージョン 6.3、バージョン 6.4、またはバージョン 6.5 からバージョン 6.6 へデータベースを変換します。

例 **1**：

```
ADACVT DBID=number CONVERT=V660
```

DBID = 数字のデータベースは、バージョン 6.1、6.2、6.3、6.4、または 6.5 からバージョン 6.6 に変換されます。

例 **2**：

```
ADACVT DBID=number CONVERT=V630
```

DBID = 数字のデータベースは、バージョン 6.6 で導入された新しい機能が使用されていなければ、バージョン 6.6 からバージョン 6.3 に変換されます。新しい機能を使用していた場合は、新しい機能を使用して行った変更を先に削除する必要があります。

再スタートに関する考慮事項

クラッシュ後に ADACVT を再スタートできない場合、元のデータベースをバックアップからリストアし、ユーティリティを再実行してください。

9 ADADBM（データベース更新）

この章では ADADBM ユーティリティについて説明します。

機能概要

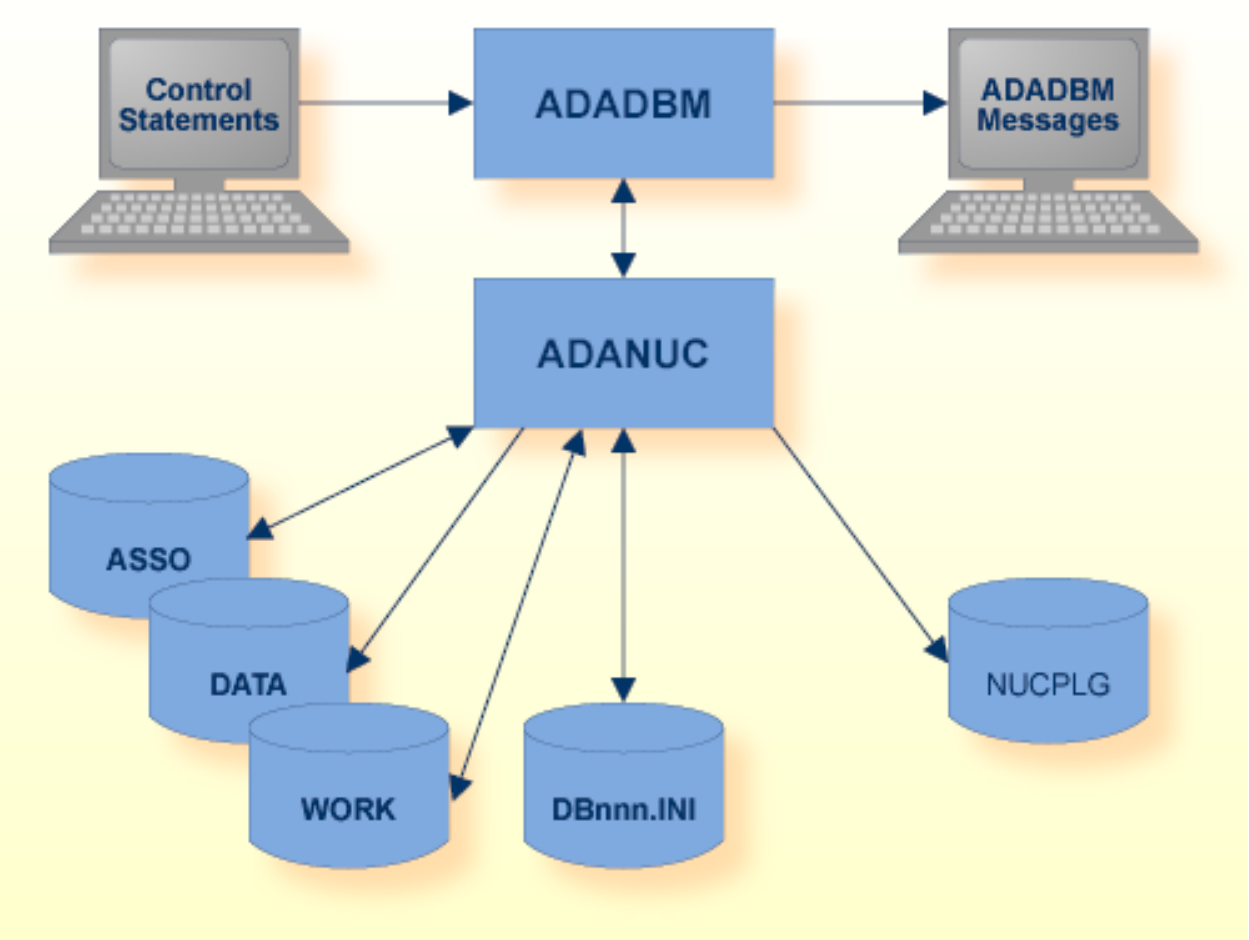
ADADBMユーティリティは、データベースに対する修正を行うために、次にあげる各機能から構成されています。

- ADD_CONTAINER 機能は、新規コンテナファイルをアソシエータまたはデータストレージデータセットに追加します。
- ADD_FIELDS 機能は、新規フィールドをファイルの FDT の最後に追加します。
- ALLOCATE NI、UI、AC、または DS 機能は、ファイルに割り当てられたノーマルインデックス、アッパーインデックス、アドレスコンバータ、またはデータストレージスペースを増やします。
- CHANGE 機能は、フィールド定義テーブル（FDT）内のフィールドの標準長を変更します。
- CHANGE_FIELDS 機能は、ファイル内の 1 つまたは複数の指定を変更します。
- DEALLOCATE 機能は、ALLOCATE とは逆の働きをする機能です。NI、UI、AC、または DS 機能は、ファイルにとって不要となったノーマルインデックス、アッパーインデックス、アドレスコンバータ、またはデータストレージスペースをフリースペーステーブル（FST）に戻します。
- DELCP 機能は、日付範囲を指定してチェックポイントファイルから古いチェックポイントレコードを削除します。
- DELETE 機能は、データベースから Adabas ファイルを 1 つまたは範囲で削除します。
- DELETE_DATABASE 機能は、データベースを削除します。指定されたキーワードに応じて、コンテナのみが削除されるか、データベースディレクトリとそのコンテンツが削除されます。
- DISPLAY 機能は、ユーティリティコミュニケーションブロック（UCB）を表示します。
- DROP_FIELDS 機能は、指定フィールドを存在しないものとしてマークします。これらは、それらにアクセスできなくなったことを意味します。
- DROP_LOBFILE 機能は、ADAFDU ADD_LOBFILE の逆の機能です。
- DROP_REFINT 機能は、既存の参照制約をドロップします。
- EXTEND_CONTAINER 機能は、データベースに定義された最後のコンテナファイルを拡張します。
- NEW_DBID 機能は、使用しているデータベースの ID を変更します。
- NEWWORK 機能は、新規の Adabas WORK データセットを割り当てし、フォーマットします。
- PGM_REFRESH 機能は、アプリケーションプログラム内での E1 コマンドによる Adabas ファイルのリフレッシュの有効／無効を切り替えます。
- RBAC_FILE 機能は、Adabas 認可モードに必要な RBAC システムファイルを作成します。

- RECOVER 機能は、フリースペーステーブルに欠如したスペースを戻します。
- REDUCE_CONTAINER 機能は、データベースに定義された最後のコンテナファイルのサイズを縮小します。
- REFRESH 機能は、1 つのファイルまたは指定範囲のファイルをレコードがまったくロードされていない状態にリセットします。
- REMOVE_CONTAINER 機能は、アソシエータまたはデータストレージデータセットからコンテナファイルを削除します。
- REMOVE_DROP 機能（後続の REFRESH と連携して使用される）は、ドロップされたフィールドを FDT から削除します。
- REMOVE_REPLICATION 機能は、すべてのレプリケーション処理を停止し、レプリケーションシステムファイルを削除します。
- RENAME 機能は、データベース名またはロードされたファイルの名前を変更します。
- RENUMBER 機能は、ロードされるファイルの番号を振り直すか、またはロードされるファイル同士の番号を交換します。
- REPLICATION_FILES 機能は、Adabas-Adabas レプリケーションに必要なシステムファイルを作成します。
- RESET 機能は、UCB からエントリを消去します。
- RESET_REPLICATION_TARGET 機能は、Adabas ファイルのレプリケーションターゲットフラグをリセットします。
- REUSE 機能は、Adabas によるデータストレージスペースや ISN の再使用を制御します。
- SECURITY 機能は、データベースのセキュリティモードを設定します。
- SYFMAX 機能は、指定されたファイル内のシステム生成マルチプルバリューフィールドに対して生成される値の最大数を指定します。

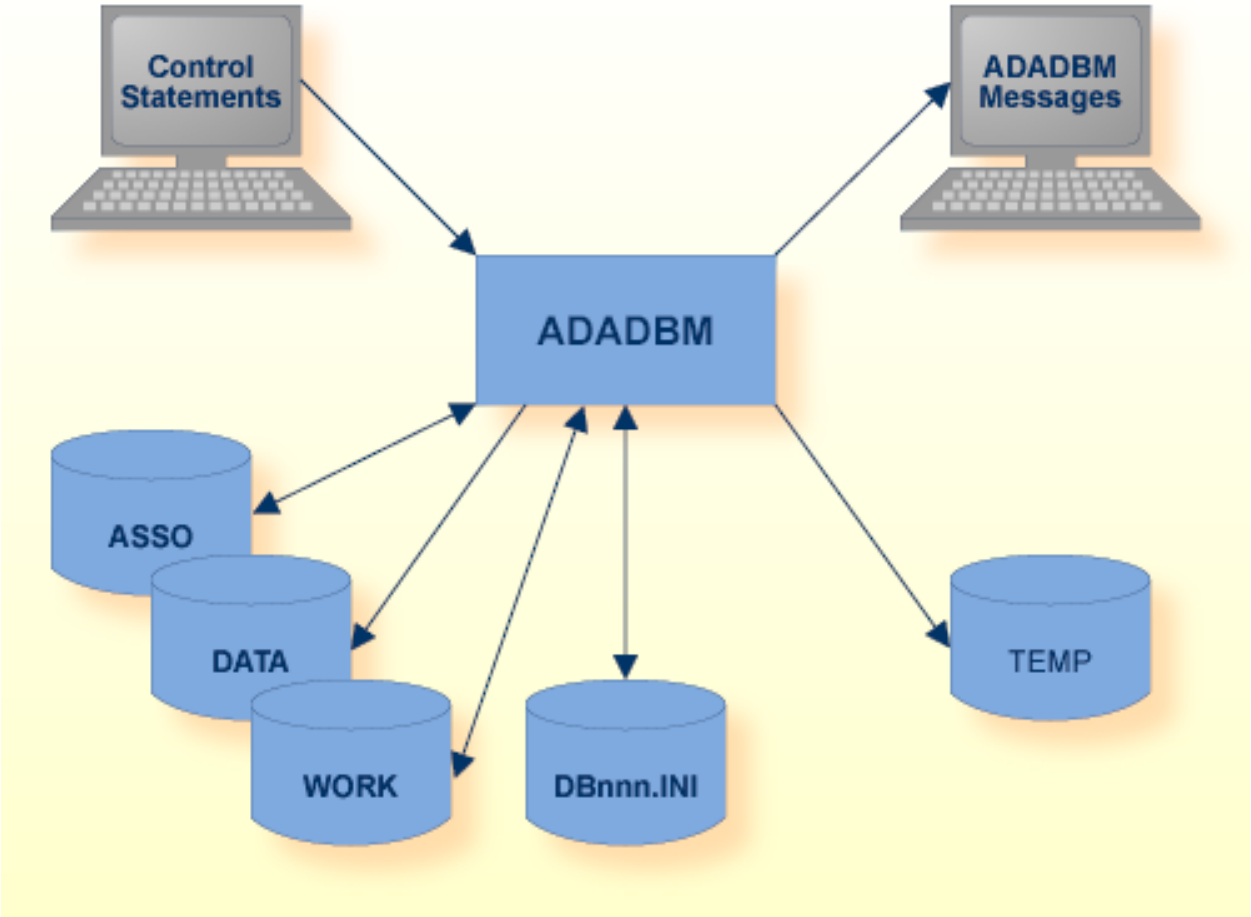
このユーティリティは多機能ユーティリティです。

処理フロー



Online Mode

Adabas ニュークリアスがアクティブな場合、ADADBM はデータベースコンテナを修正するためにニュークリアスを呼び出します。いくつかのタスクは、チェックポイントが書き込まれませんが、アクティビティがデータベースログに書き込まれます。リカバリの場合は、アクティブティが自動的に再実行されます。



Offline Mode

Adabas ニュークリアスがアクティブでない場合、ADADBM がデータベースコンテナを修正します。

データセット	環境変数／論理名	記憶媒体	追加情報
アソシエータ	ASSOx	ディスク	
データストレージ	DATAx	ディスク	
DBnnn.INI		ディスク	Adabas 拡張オペレーションマニュアル
コントロールステートメント	stdin		ユーティリティマニュアル
ADADBM メッセージ	stdout		メッセージおよびコード
プロテクションログ（オンラインモードのみ）	NUCPLG	ディスク	ユーティリティマニュアル：ADANUC、ADAPLP
一時ストレージ（オフラインモードのみ）	TEMP1	ディスク	NEWWORK 機能用の新しい WORK データセット。この機能の実行後には、WORK の環境変数／論理名

データセット	環境変数／論理名	記憶媒体	追加情報
			のポイント先を新しい WORK データセットに変更する必要があります。
WORK	WORK1	ディスク	

チェックポイント

次の表は、各機能に対するニュークリアス条件と記録チェックポイントを示しています。

機能	ニュークリアスのアクティブ化が必要	ニュークリアスの非アクティブ化が必要	ニュークリアスは不要	記録チェックポイント
ADD_CONTAINER			X	SYNP
ADD_FIELDS			X	SYNP（オフライン） SYNX（オンライン）
ALLOCATE			X	SYNP
CHANGE	X			-
CHANGE_FIELDS			X	SYNP（オフライン） SYNX（オンライン）
DEALLOCATE			X	SYNP
DELCP	X			SYNP
DELETE			X	SYNP（オフライン） SYNX（オンライン）
DELETE_DATABASE		X		-
DISPLAY			X	-
DROP_FIELDS			X	SYNP（オフライン） SYNX（オンライン）
DROP_LOBFILE			X	SYNP
EXTEND_CONTAINER		WORK の場合	ASSO または DATA の場合	SYNP
NEW_DBID		X（注 1 参照）		SYNP
NEWWORK		X（注 1 参照）		SYNP
PGM_REFRESH			X	SYNP
RECOVER			X	SYNP

機能	ニュークリアスのアクティブ化が必要	ニュークリアスの非アクティブ化が必要	ニュークリアスは不要	記録チェックポイント
REDUCE_CONTAINER			ASSO または DATA の場合	SYNP
REFRESH			X	SYNP
REMOVE_CONTAINER		X		SYNP
REMOVE_REPLICATION		X		SYNP（オフライン）
RENAME			X	SYNP
RENUMBER			X	SYNP
REPLICATION_FILES			X	SYNP（オフライン） SYNX（オンライン）（注 2 参照）
RESET			X	SYNX
RESET_REPLICATION_TARGET				
REUSE	X			SYNP
SECURITY		X		SYNP（オフライン）
SYFMAX			X	SYNP（オフライン） SYNX（オンライン）

注意:

1. この機能には、データベースコンテナファイルへの排他アクセスが必要です。
2. また、ADADBM または ADAFDU チェックポイントが（オフラインモードでも）生成され、削除または生成されたシステムファイル番号が示されます。

制御パラメータ

次のコントロールパラメータを使用できます。

```
      ADD_CONTAINER = keyword
D                    [,BLOCKSIZE=number[K] ]
                      ,SIZE = number [B|M]

     ADD_FIELDS = number {field_specification|FDT} ... [END_OF_FIELDS]

     ALLOCATE = keyword,  FILE = number [,RABN = number],
                SIZE = number [B|M]
```

```
     CHANGE = number, FIELD = string, LENGTH = number

     CHANGE_FIELDS = number {field_specification|FDT} ... [END_OF_FIELDS]

M    DBID = number

     DEALLOCATE = keyword,  FILE = number [,RABN = number],
                  SIZE = numberB

     DEFINE_REFINT = number  constraint_specification

     DELCP = { * | ([absolute-date] [,[absolute-date]]) }

     DELETE = (number [-number][,number[-number]]...)

     DELETE_DATABASE = keyword

     DISPLAY = UCB

     DROP_FIELDS = number {field_name|FDT} ... [END_OF_FIELDS]

     DROP_LOBFILE = number

     DROP_REFINT = number, NAME {=|:}constraint_name

     EXTEND_CONTAINER = keyword, SIZE = number [B|M]

D    [NO]LOWER_CASE_FIELD_NAMES

     NEW_DBID = number

     NEWWORK [,BLOCKSIZE=number[K] ], SIZE = number [B|M]

     PGM_REFRESH = keyword, FILE = number

     RBAC_FILE = number

     RECOVER

     REDUCE_CONTAINER = keyword, SIZE = number B

     REFRESH = (number [-number][,number[-number]]...)

     REMOVE_CONTAINER = keyword

     [NO]REMOVE_DROP

     REMOVE_REPLICATION

     RENAME = number, NAME {=|:} string

     RENUMBER = (number, number)
```

```
    REPLICATION_FILES = (file1, file2, file3, file4)

    RESET = UCB, IDENT = { (number [,number]...) | * }

    RESET_REPLICATION_TARGET = number

    REUSE = (keyword [,keyword]), FILE = number

    SECURITY = keyword

    SYFMAX = number, FILE = number
```

ADD_CONTAINER

```
ADD_CONTAINER = keyword
                [,BLOCKSIZE=number[K] ]
                ,SIZE = number [B|M]
```

ADD_CONTAINER パラメータは、使用されるキーワードに従って、新規コンテナファイルを既存のアソシエータまたはデータストレージデータセットに追加します。キーワードの値は ASSO または DATA のいずれかです。

新しいコンテナファイルは、現在のコンテナファイルと同じデバイスに割り当てるか、または別のデバイスタイプに割り当てることができます。

新規コンテナファイルの配置は、環境変数／論理名 ASSOx や DATAx に応じて変わります。これには、完全なパス名を持つ正式なファイル名を設定する必要があります。ASSOx も DATAx も設定しない場合には、新規コンテナファイルは現在のディレクトリ内に作成されます。

注意:

1. 新規 DATA コンテナを追加する場合、既存のアソシエータにデータストレージスペーステーブル（DSST）用のフリースペースが必要です。新規 DATA ブロックごとに 1 バイトずつ必要です。したがって、既存のアソシエータにフリースペースがなければ、最初に新規 ASSO コンテナを追加する必要があります。
2. Adabas ニュークリアスがアクティブな状態で、新しいコンテナを追加する場合、ADADBM でコンテナを追加することができませんが、ADANUC で追加することが可能です。この場合、コンテナファイルを定義するときに、注意が必要です。コンテナファイルを環境変数として定義している場合、現在のニュークリアスセッションに対応する環境変数／論理名を定義しているときにのみ、新しいコンテナを追加できます。このようにしない場合は、現在のニュークリアスセッションを終了してから新しいコンテナを追加する必要があります。一方、コンテナファイルを *DBnnn.INI* ファイルで指定している場合は、コンテナを追加する直前に、ファイルに新しいコンテナを入力できます。これは、コンテナが追加される前に、ニュークリアスがファイルを再び読み取ることが理由です（*DBnnn.INI* ファイルの詳細については、「*Adabas 拡張オペレーション*」を参照）。

3. Adabasニュークリアスがアクティブでないときに新しいコンテナを追加すると、この新しいコンテナファイルのエントリは DBnnn.INI ファイルに追加されます。

BLOCKSIZE = number[K]

新しいコンテナファイルのブロックサイズをバイト（または、Kが数値の後に指定される場合はキロバイト）で指定します。

指定した値以上かつ最も近い 1K の倍数に切り上げられます。最小ブロックサイズは 1K、最大ブロックサイズは 32K です。

BLOCKSIZEに対するデフォルト値は、データベース内に存在する対象データセットの最後のコンテナファイルのブロックサイズです。

SIZE = number [B|M]

このパラメータは、新規コンテナファイルに対して割り当てられるブロック数（B）またはメガバイト数（M）を指定するものです。デフォルトでは、サイズはメガバイト単位で認識されます。

例

```
adadbm: add_container=data, size=10
%ADADBM-I-CREATED, dataset DATA2 , file /FS/fs0395/Adabas/adadb/db076/DATA2 created
%ADADBM-I-FUNC, function ADD_CONTAINER executed
```

10 メガバイトの新規コンテナファイルがデータストレージに追加されます。ブロックサイズは DATA1 のブロックサイズと同じです。

ADD_FIELDS

```
ADD_FIELDS = number {field_specification|FDT}... [END_OF_FIELDS]
```

ADD_FIELDS パラメータは、number で定義されたファイルの末尾に 1 つまたは複数の新規フィールドを追加します。LOB ファイルを指定することはできません。この機能は END_OF_FIELDS の入力で完了します。

END_OF_FIELDS パラメータを使用してフィールド定義を消去する場合、LOWER_CASE_FIELD_NAMES パラメータを使用するときに、このパラメータを大文字で指定する必要があります。また、LOWER_CASE_FIELD_NAMESを使用する場合は、FDT パラメータも大文字で指定する必要があります。

注意: ADADBM では、派生ディスクリプタを追加できません。これを追加するには、代わりに、ユーティリティ ADAINV を使用する必要があります。

field_specification

この場合のフィールド指定リストは、ADAFDU での FDT 入力と同じ方法で入力されます。

```
level-number, name [,length] [,format] [(,option)...]
```

第 1 追加フィールドはレベル 1 のフィールドとなる必要があります。

NN オプションは使用することができません。Adabas ニュークリアスがアクティブで、NU または NC オプションが一緒に指定されるときに限り、DE は指定できます。それ以外では、ADAINV ユーティリティを使用して新規のフィールドディスクリプタステータスを指定します。UQ は DE オプションとともにのみ指定できます。

注意: システム生成フィールド（フィールドオプション SY が指定されたフィールド）をファイルに追加すると、データベース内にすでに存在していたレコード内では、これらのフィールドに空値が含まれます。これは、SY オプションが指定されていないフィールドと同じ動作です。

FDT

このパラメータは、フィールドを追加するファイルの FDT を表示します。

例

```
adadbm: add_fields=12
adadbm: fdt

Field Definition Table:

   Level  I Name I Length I Format I   Options      I Flags
-------------------------------------------------------------------------------
  1       I  AA  I   15   I    A   I DE,UQ,NU       I
  1       I  AB  I    4   I    F   I FI             I
  1       I  AC  I    8   I    A   I DE             I
  1       I  CD  I        I        I                I
   2      I  AD  I   20   I    A   I DE,NU          I      SP
   2      I  AE  I   20   I    A   I NU             I
   2      I  AF  I   10   I    A   I DE,NU          I
  1       I  AG  I    2   I    U   I NU             I      SP
  1       I  AH  I    1   I    A   I DE,FI          I
  1       I  AI  I    1   I    A   I FI             I
  1       I  AJ  I    6   I    U   I NU             I      SP
  1       I  AK  I        I        I                I
   2      I  AL  I    3   I    A   I NU             I
   2      I  AM  I    4   I    P   I NU,MU          I
-------------------------------------------------------------------------------
   Type   I Name I Length I Format I   Options      I Parent field(s)   Fmt
-------------------------------------------------------------------------------
```

```
 SUPER    I  AN  I    4   I    B   I NU             I AJ (   5 -   6 )   U
          I      I        I        I                I AJ (   3 -   4 )   U
-------------------------------------------------------------------------------
 SUPER    I  AO  I   22   I    A   I NU             I AG (   1 -   2 )   U
          I      I        I        I                I AD (   1 -  20 )   A
-------------------------------------------------------------------------------



adadbm: 01,dd,1,a
adadbm: 01,gr
adadbm: 02,g1,20,a,fi
adadbm: fdt

Field Definition Table:

   Level  I Name I Length I Format I   Options      I Flags
-------------------------------------------------------------------------------
  1       I  AA  I   15   I    A   I DE,UQ,NU       I
  1       I  AB  I    4   I    F   I FI             I
  1       I  AC  I    8   I    A   I DE             I
  1       I  CD  I        I        I                I
   2      I  AD  I   20   I    A   I DE,NU          I      SP
   2      I  AE  I   20   I    A   I NU             I
   2      I  AF  I   10   I    A   I DE,NU          I
  1       I  AG  I    2   I    U   I NU             I      SP
  1       I  AH  I    1   I    A   I DE,FI          I
  1       I  AI  I    1   I    A   I FI             I
  1       I  AJ  I    6   I    U   I NU             I      SP
  1       I  AK  I        I        I                I
   2      I  AL  I    3   I    A   I NU             I
   2      I  AM  I    4   I    P   I NU,MU          I
  1       I  DD  I    1   I    A   I                I
  1       I  GR  I        I        I                I
   2      I  G1  I   20   I    A   I FI             I
-------------------------------------------------------------------------------
   Type   I Name I Length I Format I   Options      I Parent field(s)   Fmt
-------------------------------------------------------------------------------
 SUPER    I  AN  I    4   I    B   I NU             I AJ (   5 -   6 )   U
          I      I        I        I                I AJ (   3 -   4 )   U
-------------------------------------------------------------------------------
 SUPER    I  AO  I   22   I    A   I NU             I AG (   1 -   2 )   U
          I      I        I        I                I AD (   1 -  20 )   A
-------------------------------------------------------------------------------



adadbm: end_of_fields
%ADADBM-I-FUNC, function ADD_FIELDS executed
```

ALLOCATE

```
ALLOCATE = keyword,  FILE = number [,RABN = number], SIZE = number [B|M]
```

指定されたキーワード（AC、DS、NI、UI）に応じて、ALLOCATE機能はノーマルインデックス（NI）、アッパーインデックス（UI）、アドレスコンバータ（AC）、またはデータストレージ（DS）を指定したサイズまで大きくすることが可能です。要求されたタイプでの拡張ができるかどうかを見るために、最終エクステントがチェックされます。現在のエクステントがまったく拡張不可能な場合には、新たな拡張が作成されます。

この機能は、自動拡張を無効にする方法をDBAに提供するものであり、エクステントの増減を自由に割り当てることができます。これは、大規模なレコード数を追加する場合に特に便利です。この機能に関しては、ファイルの排他制御は必要ありません。

FILE = number

このパラメータは、拡張対象ファイルを指定するものです。

RABN = number

このパラメータは、割り当て開始RABNを指定するものです。LOBファイルのNIまたはUIの割り当てでは、指定RABNのブロックサイズを16 KBより小さくする必要があります。LOBファイルのDS割り当てでは、指定RABNのブロックサイズを32 KBにする必要があります。

SIZE = number [B|M]

このパラメータは、拡張エリアのサイズを指定します。数値に"B"が付いている場合、大きさはブロック扱いとなり、これ以外はメガバイト扱いとなります。

例

```
adadbm: allocate=ni, file=11, size=100b
%ADADBM-I-ALLOC, 100 NI blocks allocated (611 - 710)


adadbm: allocate=ds, file=11, size=10
%ADADBM-I-DEALLOC, 2560 DS blocks allocated (245 - 2804)
```

CHANGE

```
CHANGE = number, FIELD = string, LENGTH = number
```

このパラメータは、数値部分に指定されたファイル内のフィールドの標準長を変更します。LOBファイルを指定することはできません。固定小数点フィールド（オプション FI）と浮動小数点フィールド（フォーマット G）の長さは変更できません。

フィールド長の変更は、データストレージ内での修正を必要としませんが、標準長を使用するプログラムに対しては影響を及ぼします。

オプション SY=OPUSER で定義されたフィールドは変更できません。

FIELD = string

このパラメータは、標準長を変更するフィールドを指定するものです。このフィールドは該当ファイルのフィールド定義テーブル内で定義されていなければなりません。

LENGTH = number

このパラメータは、フィールドの新たな標準長を定義するものです。

例

```
adadbm: change=12, field=ac, len=11
%ADADBM-I-FUNC, function CHANGE executed
```

CHANGE_FIELDS

```
CHANGE_FIELDS = number {field_specification|FDT}... [END_OF_FIELDS]
```

CHANGE_FIELDS 機能は、number で定義されたファイル内の 1 つまたは複数のフィールド指定を変更します。この機能は END_OF_FIELDS の入力で完了します。

END_OF_FIELDS パラメータを使用してフィールド定義を消去する場合、LOWER_CASE_FIELD_NAMES パラメータを使用するときに、このパラメータを大文字で指定する必要があります。また、LOWER_CASE_FIELD_NAMESを使用する場合は、FDT パラメータも大文字で指定する必要があります。

許可される変更は、ファイル内にレコードが存在するかどうかに応じて異なります。すべてのファイルに次の制限事項が適用されます。

- フィールドレベル番号は変更できません。
- グループはグループのままにする必要がありますが、レベル1で定義されている場合は、ピリオディックグループに変換できます。

- ピリオディックグループは、ピリオディックグループのままにするか、非ピリオディックグループに変換する必要があります。
- グループでもピリオディックグループでもないフィールドは、グループまたはピリオディックグループに変換しないでください。

空でないファイルには、次の追加の制限事項が適用されます。

- フィールド長：新しい長さは、新しいフィールドフォーマットおよびフィールドオプションと互換性がある必要があります。長さを変更すると、フォーマットバッファでフィールド長が指定されていない Adabas コマンドの動作が変化します。
- フィールドフォーマット：A は W に変更でき、その逆も可能です。フォーマットを A から W に変更した場合は、フィールドに UTF-8 値が含まれていることを確認するのはユーザーの責任です。フォーマットを W から A にした場合は、フィールドに UTF-8 値が含まれます。Adabasコマンドのフォーマットバッファで指定する形式は、AおよびWフィールドのフィールド定義のフォーマットと同になっている必要があります。したがって、既存のプログラムを適宜適合させることが必要になる場合があります。AとWの間で変更する場合を除き、フィールドフォーマットのその他の変更は許可されません。
- フィールドオプション：オプション DE、FI、HF、MU、UQ は、追加も削除もできません。

  次のフィールドオプションの変更が可能です。

古いフィールドオプション	新しいフィールドオプション	コメント
DT が設定されていません	DT が設定されます、TZ は設定されることも設定されないこともあります	データベース内の値が指定された日付／時刻編集マスクに準拠しているかどうかを確認するチェックは行われません。TZ を編集マスク名 DATE、TIME、およびNATDATEに設定することはできません。 **注意:** 通常、TZ オプションで定義されたフィールド値のセマンティクスとTZオプションなしで定義されたフィールド値は異なります。TZ 値のないフィールドは、通常、ローカル時間値を含み、TZ 値があるフィールドは UTC 値を含みます。フィールド値は自動的には更新されません。必要な更新が行われていることを確認するのはユーザーの責任です。
DT が設定されました	DT が設定されていません	フォーマットバッファのフィールドに日付／時刻編集マスクは指定できなくなりました。
HF が設定されています	HF が設定されていません	クロスプラットフォームコールの動作が変更されます。この変更を適用するには、ファイルが空になっている必要があります。
HF が設定されていません	HF が設定されています	
LA および LB が設定されていません	LA または LB が設定されています	可変長を含むフィールドにアクセスするコールの動作が変更されます。

古いフィールドオプション	新しいフィールドオプション	コメント
LA が設定されています	LAが設定されていません、LB が設定されました	
LB が設定されています	LBが設定されていません、LA が設定されています	ファイルに LOB ファイルが定義されていない場合、またはフィールドが派生ディスクリプタの親のディスクリプタの場合にのみ許可されます。可変長を含むフィールドにアクセスするコールの動作が変更されます。
NB が設定されていません	NB が設定されています	
NC および NN が設定されました	FI、NC、NU、および NN が設定されていません	この変更の後、N1/N2 コマンドのフォーマットバッファでフィールドが必須ではなくなります。指定しない場合、フィールドには Adabas 空値が設定されます。
NC および NN が設定されました	NC が設定されています、NN が設定されていません	この変更の後、N1/N2 コマンドのフォーマットバッファでフィールドが必須ではなくなります。指定しない場合、フィールドには SQL 空値が指定されます。
NU が設定されています	NC が設定されています	空の値は空値に変換されます。NU と NC は相互に排他的なので、「NC が設定されている -> NU が設定されていない」になります。
NV が設定されています	NV が設定されていません	クロスプラットフォームコールの動作が変更されます。
NV が設定されていません	NV が設定されています	
SY が設定されていません	SY が設定されています	A1、N1、および N2 コマンドの動作が変更されます。フィールドフォーマットは、SY オプションと互換性がある必要があります。既存値の妥当性を確認するチェックは行われないことに注意してください。
SY が設定されています	SY が設定されていません	A1、N1、および N2 コマンドの動作が変更されます。
TR が設定されていません	TR が設定されています	
TZ が設定されていません、DT が設定されています	TZ が設定されています、DT は変更されません	日付／時刻の編集マスクを指定すると、データベース内の値が UTC からローカル時間に変換されます。
TZ が設定されています	TZ が設定されていません	日付／時刻の編集マスクを指定したときに、データベース内の値が UTC からローカル時間に変換されなくなります。

field_specification

この場合のフィールド指定リストは、ADAFDU での FDT 入力と同じ方法で入力されます。

```
level-number, name [,length] [,format] [(,option)...]
```

第 1 追加フィールドはレベル 1 のフィールドとなる必要があります。

FDT

このパラメータは、フィールドを追加するファイルの FDT を表示します。

DBID

```
DBID = number
```

このパラメータは、使用対象となるデータベースを選択するためのものです。

注意: ニュークリアスのシャットダウンを要求または許可するユーティリティ機能は、そのデータセットに対して論理割り当てを必要とします。

例

```
adadbm: dbid=76
%ADADBM-I-DBOFF, database 76 accessed offline


adadbm: dbid=76
%ADADBM-I-DBON, database 76 accessed online
```

DEALLOCATE

```
DEALLOCATE = keyword,  FILE = number [,RABN = number],
             SIZE = numberB
```

DEALLOCATE = AC、DS、NI、または UI

指定されたキーワード（AC、DS、NI または UI）に応じて、このパラメータはアドレスコンバータ（AC）、データストレージ（DS）、ノーマルインデックス（NI）、またはアッパーインデックス（UI）に与えられているスペースを解放します。

エクステントに過度のスペースが割り当てられている場合、DBA は自動または手動でこのスペースをフリースペーステーブル（FST）に戻すことができます。

割り当ての解除は、1 回の実行につき 1 つのエクステントに限定されます。複数エクステントからスペースを解放するには、DEALLOCATE を複数回コールする必要があります。

FILE = number

このパラメータは、ファイルを指定します。

RABN = number

このパラメータは、解放するスペースの最初の RABN を指定するものです。このパラメータが省略された場合には、最終エクステントの末尾からスペースが解放されます。

SIZE = numberB

このパラメータは、解放するスペースのエリアの大きさをブロック単位で指定します。

例

```
adadbm: deallocate=ni, file=11, size=110b
SIZE=110B
        ^
%ADADBM-E-VALUP, value has to be less-equal 100
%ADADBM-I-ABORTED,      14-NOV-2002 14:44:01, elapsed time: 00:00:00
```

```
adadbm: deallocate=ni, file=11, size=100b
%ADADBM-I-DEALLOC, 100 NI blocks deallocated (611 - 710)


adadbm: deallocate=ni, file=11, size=10b
%ADADBM-I-DEALLOC, 10 NI blocks deallocated (323 - 332)
```

DEFINE_REFINT

```
DEFINE_REFINT = number constraint_specification
```

この機能は、外部キーを含むファイルnumberに参照制約を追加します。制約の構文は、ADAFDU の FDT ファイルで使用される構文と同じで、『*管理マニュアル*』の「*FDT レコード構造*」、「参照制約」で説明されています。この制約は、プライマリファイルのFDTにも含まれるので、制約名がプライマリファイルで定義されていないようにする必要があります。

プライマリファイルとして指定されたファイルが PGM_REFRESH = YES で定義されている場合、参照制約の追加は許可されません。

参照整合性に違反があると、制約の追加は失敗します。参照整合性を確立するための、ファイルデータに対する更新は実行されません。

DELCP

```
DELCP = { * | ([absolute-date] [,[absolute-date]]) }
```

この機能は、チェックポイントファイルからチェックポイントレコードを削除します。

アスタリスク（*）が入力されると、全チェックポイントレコードが削除されます。

例

```
adadbm: delcp=13-NOV-2006:15:09:48
%ADADBM-I-DELCP, 1 record deleted from CHECKPOINT file

adadbm: delcp=(13-NOV-2006:15:09:48,)
%ADADBM-I-DELCP, 81 records deleted from CHECKPOINT file

adadbm: delcp=(,14-NOV-2006:14:37:24)
%ADADBM-I-DELCP, 41 records deleted from CHECKPOINT file

adadbm: delcp=(14-NOV-2006:14:37:25,14-NOV-1996:14:38:15)
%ADADBM-I-DELCP, 42 records deleted from CHECKPOINT file

adadbm: delcp=*
%ADADBM-I-DELCP, 20 records deleted from CHECKPOINT file
```

DELETE

```
DELETE = (number [-number][,number[-number]]...)
```

DELETEパラメータは、データベースから1つのファイルまたはある範囲のファイルを削除し、フリースペーステーブル（FST）に該当ファイルが割り当てられていた全スペースを戻します。LOBファイルが指定された場合は無視されますが、LOBファイルが割り当てられている基本ファイルをすべて指定した場合には、その LOB ファイルも削除されます。削除するファイルと指定されていない別のファイルとの間に、参照制約が存在していない必要があります。システムファイルの削除は許可されていません。

注意: Adabas-to-Adabas レプリケーションの使用を停止する必要があり、そのためにレプリケーションシステムファイルを削除する場合は、DELETE FUNCTION *ではなく* ADADBM REMOVE_REPLICATION を使用する必要があります。

ADADBMはファイル削除に対する確認要求を行いません。したがって、ファイル番号を指定する際には十分な注意が必要です。

例

```
adadbm: delete=(4-11,14)
%ADADBM-I-DELETED, file 11 deleted
%ADADBM-I-DELETED, file 14 deleted
```

DELETE_DATABASE

```
DELETE_DATABASE = keyword
```

DELETE_DATABASE機能は、データベースを削除します。指定されたキーワード（CONTAINERまたはFULL）に応じて、コンテナのみが削除されたり、データベースディレクトリとその内容が削除されたりします。

キーワードCONTAINERを指定すると、ADABAS.INIファイルの[DB_LIST]セクションにあるコンテナファイルとDBIDエントリが削除されます。キーワードFULLを指定すると、データベースディレクトリとそのすべての内容が削除されます。

例：

```
adadbm: dbid=12 delete_database=container
```

DBID 12のデータベースのコンテナが削除されます。

DISPLAY

```
DISPLAY = UCB
```

DISPLAY機能は、ユーティリティコミュニケーションブロックを表示します。また、この機能は、AUTORESTARTが保留状態でも実行が可能です。

例：

```
adadbm: display=ucb

       Date/Time        Entry Id  Utility  Mode Files
       ---------        --------  -------  ---- -----
14-NOV-2006 14:38:40         233   adaopr  UTO  11
14-NOV-2006 14:38:42         234   adabck  ACC  *
```

次の項目を表示します。

- ［DATE/TIME］には、ファイルがロックされた日付と時刻が表示されます。
- ［ENTRY ID］には、エントリに割り当てられたIDが表示されます。
- ［UTILITY］には、ユーティリティの名前が表示されます。

- ［MODE］には、ファイルがアクセスされているモードが表示されます。
- ［FILES］には、ロックされているファイルの数が表示されます。

DROP_FIELDS

```
DROP_FIELDS = number {field_name|FDT}... [END_OF_FIELDS]
```

DROP_FIELDS 機能は、"number" で定義されたファイルから 1 つ以上のフィールドをドロップします。指定されたフィールドはもはや存在しないものとしてマークされ、それらにアクセスすることはできません。LOB ファイルを指定することはできません。この機能は END_OF_FIELDS の入力で完了します。

END_OF_FIELDS パラメータを使用してフィールド定義を消去する場合、LOWER_CASE_FIELD_NAMES パラメータを使用するときに、このパラメータを大文字で指定する必要があります。また、LOWER_CASE_FIELD_NAMES を使用する場合は、FDT パラメータも大文字で指定する必要があります。

グループまたはピリオディックグループを指定する場合、グループまたはピリオディックグループに属しているすべてのフィールドがドロップされます。ディスクリプタまたはディスクリプタから派生したフィールドを指定することはできません。そのようなフィールドをドロップしたい場合は、最初に対応するすべてのディスクリプタを ADAINV で解放する必要があります。

DROP_FIELDS 機能が実行された後、ドロップされたフィールドの名前を、例えば ADADBM の ADD_FIELDS 機能を使用して再定義することができます。

注意:

1. DROP_FIELDS機能は、フィールドを物理的に削除するわけではありません。繰り返しフィールドをドロップしたり追加したりしないでください。ファイルのデータレコードや FDT がオーバーフローする可能性があります。
2. ファイルのフィールドが見かけ上は同じでも、ドロップしたフィールドが違っていれば、そのファイルにデータをロードすることはできません。

FDT

このパラメータは、フィールドがドロップされるファイルの FDT を表示します。

DROP_LOBFILE

```
DROP_LOBFILE = number ↩
```

割り当て先の LOB ファイルが空になっている LOB ファイルの基本ファイルを削除するためのファイル番号を数値で指定する必要があります。

割り当て先の LOB ファイルが空でない場合、DROP_LOBFILE は許可されません。

DROP_REFINT

```
DROP_REFINT = number, NAME {=|:} constraint_name
```

この機能は、外部キーを含むnumberで指定されたファイルから参照制約を削除します。この制約は、プライマリファイルの FDT からも削除されます。

EXTEND_CONTAINER

```
EXTEND_CONTAINER = keyword, SIZE = number [B|M]
```

EXTEND_CONTAINER機能は、使用されたキーワードに従ってデータベース用に定義された最後のアソシエータ、データストレージ、または WORK コンテナファイルを拡張します。キーワードの値は、ASSO、DATA、または WORK です。

注意: WORK コンテナはオフラインモードでのみ拡張することができます。

SIZE = number [B|M]

このパラメータはブロック単位（B）またはメガバイト単位（M）で拡張エリアのサイズを指定します。デフォルトでは、サイズはメガバイト単位です。

[NO]LOWER_CASE_FIELD_NAMES

```
[NO]LOWER_CASE_FIELD_NAMES
```

LOWER_CASE_FIELD_NAMES が指定されている場合、Adabas フィールド名は、大文字に変換されません。NOLOWER_CASE_FIELD_NAMESが指定されている場合、Adabasフィールド名は、大文字に変換されます。デフォルトは、NOLOWER_CASE_FIELD_NAMES です。

このパラメータは、ADD_FIELDS、CHANGE_FIELDS、またはDEFINE_REFINTパラメータの前に指定する必要があります。

NEW_DBID

```
NEW_DBID = number
```

この機能は、使用されているデータベースの ID の変更に使用されます。新しい ID は、他のアクティブデータベースによって使用されているときには、このパラメータを使用することはできません。

例

```
adadbm: new_dbid=77
%ADADBM-I-FUNC, function NEW_DBID executed
```

NEWWORK

```
NEWWORK [,BLOCKSIZE = number[K] ], SIZE = number [B|M]
```

この機能では既存の WORK1 コンテナファイルが削除され、新しい WORK1 コンテナファイルで置き換わります。新規 WORK1 コンテナファイルを割り当て、その後、必要に応じてフォーマットします。

新規 WORK を作成する前に、ニュークリアスおよびデータベースを使用している全ユーティリティを正常終了させなければいけません。この機能は、現在の WORK を必要とするため、現在の WORK は NEWWORK を実行する前に削除してはいけません。この機能を使用する場合、新規 WORK ファイルには TEMP1 を指示する必要があります。

注意: 新規 WORK の参照先をディスクセクションやファイルシステムに設定できます。TEMP1 が WORK1 と同じディスクセクションをポイントする場合、ADADBM は既存の WORK ファイルを拡張／縮小しようとします。いずれの場合も、新規 WORK コンテナファイルの名前は WORK1 です。この機能が正常に完了すると、古い WORK1 は削除されます。

BLOCKSIZE = number[K]

新しいコンテナファイルのブロックサイズをバイト（または、Kが数の後に指定される場合はキロバイト）で指定します。

Adabas は、指定された値を 1024 の倍数に切り上げます。

指定できる最小ブロックサイズは 3072 で、最大ブロックサイズは 32768 です。

これらの最小値と最大値に加えて、ASSO と WORK のブロックサイズには、通常次のサイズ制限が適用されます。

MAX (ASSOBLS) < WORKBLS

MAX(ASSOBLS) は、ASSO の最大ブロックサイズを表わし、WORKBLS は WORK ブロックサイズを表わします。

BLOCKSIZE に対するデフォルト値は、古い WORK ファイルのブロックサイズです。

SIZE = number [B|M]

このパラメータは、新規 WORK ファイルに対して割り当てられるブロック数またはメガバイト数を指定するものです。デフォルトでは、サイズはメガバイト単位です。最小値は 200 ブロック、またはメガバイト単位での同等の値です。

PGM_REFRESH

```
PGM_REFRESH = keyword, FILE = number
```

このパラメータは、アプリケーションプログラムの E1 コマンド（ISN=0、CID=空白）で Adabas ファイルをリフレッシュ可能にしたり、不可能にしたりします。LOB ファイルを指定することはできません。キーワードには、YES または NO を指定します。参照制約のプライマリファイルとなっているファイルには、PGM_REFRESH = YES を設定できません。

FILE = number

このパラメータには、リフレッシュを可能／不可能にするファイル番号を指定します。

RBAC_FILE

```
RBAC_FILE = number
```

この機能は RBAC システムファイルを作成し、初期セキュリティ定義をロードします。

例

```
adadbm: rbac_file=200
```

RECOVER

```
RECOVER
```

この機能は、アソシエータまたはデータストレージ内の失われたスペースをフリースペーステーブル（FST）に戻します。

スペースは、Adabas ユーティリティが正常終了しないと失われます。

例

```
adadbm: recover
%ADADBM-I-FUNC, function RECOVER executed
```

REDUCE_CONTAINER

```
REDUCE_CONTAINER = keyword, SIZE = number B
```

REDUCE_CONTAINER 機能は、使用されたキーワードに従って、データベースに定義されたアソシエータまたはデータストレージコンテナ末尾のフリースペースを割り当て解除します。キーワードに使用できる値は、ASSO または DATA です。

縮小する分のブロック数は、指定したコンテナの最後の部分では使用されていない必要があります。1つ以上のコンテナエクステントのスペース全体を解放すると、そのコンテナエクステントは削除されます。ADADBMをオンラインで実行している場合、ADADBMは、コンテナエクステントが削除されることを通知するメッセージを表示しません。このメッセージはニュークリアスログに出力されます。

コンテナの末尾にあるフリースペースが、要求されたブロックよりも少ない場合、コンテナの末尾にあるすべてのフリースペースが割り当て解除され、次の警告が表示されます。

```
%ADADBM-W-PREDCONT, not all requested blocks removed
```

SIZE = number B

このパラメータは、コンテナを減らすサイズをブロック単位で指定します。

REFRESH

```
REFRESH = (number [-number][,number[-number]]...)
```

この機能は、"number" に指定されたファイルをレコードがロードされていない状態にリセットします。ノーマルインデックス、アドレスコンバータおよびデータストレージに対する最初のエクステントだけは保持されます。その他のエクステントはフリースペーステーブル（FST）に戻されます。アッパーインデックスは再構築され、未使用のアッパーインデックスのエクステントはフリースペーステーブルに戻されます。LOB ファイルが指定された場合は無視されますが、LOB ファイルが割り当てられている基本ファイルをすべて指定した場合には、その LOB ファイルもリフレッシュされます。参照整合性制約のプライマリファイルは、参照制約の外部ファイルもリフレッシュされている場合にのみリフレッシュできます。

ADADBMは、リフレッシュされるファイルの確認を要求しません。したがって、ファイル番号を指定する際には十分な注意が必要です。

この機能は、テスト環境でのテストファイルをクリアする場合に便利です。ファイルを削除してから再びロードするよりもこの方法のほうが効率的です。

ADAM 機能を使用しているファイルはリフレッシュできません。

REMOVE_DROP 機能が設定されている場合、ドロップするフィールドは FDT から削除されます。

例

```
adadbm: refresh=13
%ADADBM-I-REFRESH, file 13 refreshed
```

REMOVE_CONTAINER

```
REMOVE_CONTAINER = keyword
```

この機能は、使用されるキーワードに従って最後のデータベースコンテナファイルを既存のアソシエータまたはデータストレージデータセットから削除します。キーワードに使用できる値は、ASSO または DATA です。

削除対象のコンテナファイルは、この機能が実行されるときに使用中であってはいけません。すなわち、ファイル内のすべてのブロックがフリーでなければいけません。

コンテナファイルは、ファイルシステムからまたは RAW ディスクセクションから削除されます。

コンテナファイルが削除できるためには、ニュークリアスおよびデータベースを使用しているすべてのユーティリティが正常に終了していなければなりません。

注意: コンテナを削除すると、DBnnn.INI ファイル内で、このコンテナファイルに対応するエントリが削除されます。

例

```
adadbm: remove_container=data
%ADADBM-I-DMCONREM, container DATA2 removed
```

REMOVE_DROP

```
[NO]REMOVE_DROP
```

REMOVE_DROP を指定すると、後続の REFRESH 機能がドロップされたフィールドを FDT から削除します。

NOREMOVE_DROP を指定すると、後続の REFRESH 機能はドロップされたフィールドを FDT から削除しません。

デフォルトは NOREMOVE_DROP です。

例

```
adadbm: remove_drop
adadbm: refresh=2
%ADADBM-I-REFRESH, file 2 refreshed
adadbm: refresh=3
%ADADBM-I-REFRESH, file 3 refreshed
adadbm: noremove_drop
adadbm: refresh=4
%ADADBM-I-REFRESH, file 4 refreshed
```

ファイル 2 はリフレッシュされ、ドロップされたフィールドは FDT から削除されました。ファイル 3 はリフレッシュされ、ドロップされたフィールドは FDT から削除されました。ファイル 4 はリフレッシュされ、ドロップされたフィールドは FDT から削除されませんでした。

REMOVE_REPLICATION

```
REMOVE_REPLICATION
```

この機能は、すべてのレプリケーション処理を停止し、すべてのレプリケーションシステムファイルを削除します。

注意: この機能は、Adabas を搭載した Event Replicator（Adabas レプリケーション）を使用している顧客にのみ関連します。

RENAME

```
RENAME = number, NAME {=|:} string
```

この機能は、ファイルまたはデータベースの名前を変更します。numberには名前の変更を行うファイル番号を指定します。number にゼロを指定すると、データベース名が変更されます。

NAME {=|:} string

"string" の部分には、指定したファイルまたはデータベースの新しい名前を指定します。等号を指定する場合、「文字列」に指定された値が大文字に変換されます。コロンを指定した場合、大文字の変換は実行されません。

例

```
adadbm: rename=11, name=employee-file
%ADADBM-I-FUNC, function RENAME executed
```

RENUMBER

```
RENUMBER = (number, number)
```

この機能は、ロードされたAdabasファイルのファイル番号を変更します。しかし、そのファイルの新しい番号がロード済みファイルに使用されていた場合には、そのファイル間でファイル番号の交換が行われます。

指定部分の最初のnumberには、現在ファイルに割り当てられている番号を指定します。そして2 番目の number には、新たにファイルに割り当てる番号を指定します。

例：

```
adadbm: renumber=(12,14)
%ADADBM-I-RENUM, File 12 renumbered to 14
%ADADBM-I-RENUM, File 14 renumbered to 12
```

REPLICATION_FILES

```
REPLICATION_FILES = (file1, file2, file3, file4)
```

この機能は、Adabas-Adabas レプリケーションに必要なすべての初期化ステップを実行し、レプリケーションシステムファイルを作成します。

file1
メタデータファイル。

file2
レプリケーショントランザクションファイル。

file3
レプリケーションコマンドファイル。

file4
レプリケーションコマンドファイルの LOB ファイル。

注意:

1. この機能は、Adabasを搭載したEvent Replicator（Adabas レプリケーション）を使用している顧客にのみ関連します。

2. Adabas-Adabas レプリケーションの初期化後は、Adabas Event Replicator をインストールした後にのみ、Adabasニュークリアスが機能します（具体的にはレプリケーション出口が必要です）。
3. レプリケーションファイルに必要なスペースは、約1MB の ASSO スペース（ブロックサイズが 16 KB 未満の小さな ASSO ブロック）と 5MB の DATA スペース（ブロックサイズは 32 KB）です。更新の負荷が高い場合、またはレプリケーションのステータスが記録になっている場合は、レプリケーションシステムファイルが大きくなる可能性があります。これは、レプリケーション対象ファイルのすべての更新処理が、ターゲットデータベースに適用されるまで、レプリケーションシステムファイルに保存されることが理由です。

RESET

```
RESET = UCB, IDENT = { (number [,number]...) | * }
```

UCB

この機能は、1つまたは複数のエントリをユーティリティコミュニケーションブロック（UCB）から削除するものです。またこのオプションは、AUTORESTART が保留状態でも使用することができます。

UCBは、データベース内の特定のリソース（データベース全体、1つまたは複数のファイル等）へのアクセスを制御するのに使用されます。またUCBは、Adabasユーティリティのデータベース処理および、ユーティリティが関与するリソース関連の情報のセーブを行います。

エントリは、ユーティリティのリソースへのアクセスが認められるごとに UCB 内に作成されます。このエントリには、ユーティリティおよびユーティリティがロックするリソースに関する情報が含まれます。リソースが必要なくなると、ユーティリティは自動的にエントリを削除します。UCB の内容の表示方法については、このユーティリティの DISPLAY=UCB の記述部分を参照してください。

ある特別な状況においては（ADAMUP の異常終了など）、UCB 内にエントリが残り、リソースがロックされたままとなります。しかし、この RESET 機能を使用し、UCB から 1 つないし複数のエントリを削除することにより、そのリソースは解放されます。

UCB エントリをリセットすると、ユーザーキューから関連するエントリを取り除き、ニュークリアスがアクティブな場合、フリースペーステーブルに失われたブロックを返します。その他の場合は、RECOVER機能を使用すれば、リソースをフリースペーステーブルに返すことができます。

IDENT = { (number [,number]...)| * }

このパラメータは、削除対象となるエントリの固有IDを指定します。アスタリスク（*）を指定すると、すべてのエントリが削除対象となります。

RESET UCB 機能をオフラインで使用する場合、* だけを指定することができます。

例

```
adadbm: reset=ucb, ident=233
%ADADBM-I-RESUCB1, 1 entry deleted from UCB


adadbm: reset=ucb, ident=(235,234)
%ADADBM-I-RESUCB, 2 entries deleted from UCB


adadbm: reset=ucb, ident=*
%ADADBM-I-RESUCB1, 1 entry deleted from UCB
```

RESET_REPLICATION_TARGET

```
RESET_REPLICATION_TARGET = number
```

この機能は、Adabasファイルのレプリケーションターゲットフラグをリセットします。その後に、再び通常のファイルとして処理されます。0を指定すると、すべてのレプリケーションターゲットファイルのレプリケーションターゲットフラグがリセットされます。ファイル番号を指定すると、このファイル番号を持つファイルのレプリケーションターゲットフラグがリセットされます。

注意:

1. この機能は、Adabasを搭載したEvent Replicator（Adabasレプリケーション）を使用している顧客にのみ関連します。
2. この機能を実行した後は、このレプリケーションターゲットへのレプリケーションができなくなります。このレプリケーションターゲットへのレプリケーションがまだアクティブな場合、レプリケーションソースでの新しい更新トランザクションによって、レプリケーションのステータスがエラーに設定されます。このレプリケーションターゲットにデータを再度レプリケーションする場合は、新しい初期状態処理が必要です。

REUSE

```
REUSE = (keyword [,keyword]), FILE = number
```

REUSE 機能は、Adabas によるデータストレージスペースまたは ISN の再利用を制御します。

指定したファイルのファイルコントロールブロック（FCB）は、新規レコードの追加または更新レコードの移動時に使用される割り当て方法を指示するように修正されます。

有効なキーワードは、[NO]DS および [NO]ISN です。

DS キーワードを指定すると、Adabas は十分なスペースを持つブロックを割り当てるために、データストレージスペーステーブル（DSST）を検索します。この場合、最初に検索された十分なスペースを持つブロックが使用されます。

NODS キーワードは、新たに追加されたレコードすべてと別のブロックに移動するレコード（更新によってレコードの拡張が発生した場合）をまとめて、ファイルに割り当てられたデータストレージエクステント内の最終使用ブロック内に移動させます。このブロックに十分なスペースがない場合、次のブロックが使用されます。

DS と NODS は相互に排他的です。デフォルトは REUSE = DS です。

ISN キーワードを設定すると、Adabas は新規レコードに対して削除レコードの ISN を再利用します。

NOISN を設定すると、Adabas は削除レコードの ISN を新規レコードに対して再利用しません。新規レコードにはそれぞれ次に大きい未使用 ISN が割り当てられます。

NOISN と ISN は相互に排他的です。デフォルトは REUSE = NOISN です。

FILE = number

このパラメータは、ファイルを指定します。

例

```
adadbm: reuse=nods, file=11
%ADADBM-I-FUNC, function REUSE executed


adadbm: reuse=(ds,isn), file=12
%ADADBM-I-FUNC, function REUSE executed
```

SECURITY

```
SECURITY = keyword
```

SECURITY機能は、データベースのセキュリティモードを設定します。キーワードは、ACTIVEまたはWARNのいずれかになります。

ACTIVE

ACTIVEキーワードは、セキュリティ機能を有効にします。ACTIVEは、承認されたユーザーのみがデータベースにアクセスできることを意味しています。認証エラーなどのセキュリティ違反は、監査証跡で「エラー」として処理されます。

セキュリティ違反の場合、データベースへのアクセスは拒否されます。

重要: データベースセキュリティを有効にした後は、セキュリティを無効にできなくなります。

WARN

WARNキーワードは、セキュリティ機能を有効にします。WARNは、すべてのユーザーがデータベースにアクセスできることを意味しています。認証エラーなどのセキュリティ違反は、監査証跡で「警告」としてプロトコル化されます。セキュリティ違反の場合、データベースへのアクセスは拒否されません。

このモードは、セキュアデータベースを使用するためにアプリケーションを移行することを意図しています。「*ニュークリアスユーザー出口21*」も参照してください。

重要: セキュリティモードWARNは、モードACTIVEにのみ変更できます。

WARN

WARNキーワードは、セキュリティ機能を有効にします。WARNは、すべてのユーザーがデータベースにアクセスできることを意味しています。認証エラーなどのセキュリティ違反は、監査証跡で「警告」としてプロトコル化されます。セキュリティ違反の場合、データベースへのアクセスは拒否されません。

このモードは、セキュアデータベースを使用するためにアプリケーションを移行することを意図しています。「*ニュークリアスユーザー出口21*」も参照してください。

重要: セキュリティモードWARNは、モードACTIVEにのみ変更できます。

デフォルトモード

デフォルトでは、セキュリティは有効になっていません。

SYFMAX

```
SYFMAX = number, FILE = number
```

このパラメータは、指定されたファイル内のシステム生成マルチプルバリューフィールドに対して、生成される値の最大数を指定します。明示的な最大値はありませんが、値が高く定義されると、レコードオーバーフローが発生することに注意してください。圧縮データレコードは1つのDATA ブロック内に収まる必要もあり、SYFMAX 値がシステム生成マルチプルバリューフィールドに対して定義されています。SYFMAX 値が下げられ、この新しい値よりも、レコードに含まれるシステム生成フィールド値の数が多くなった場合は、このレコードの次の更新処理で余分な値が削除されます。

FILE = number

このパラメータは、ファイルを指定します。

再スタートに関する考慮事項

ADADBMは再スタート機能を備えていません。しかし、各機能の終了に際してシステムは、その機能が正常終了したかどうかを調査します。正常終了しなかった場合には、その機能は再スタートする必要があります。

10 ADADCU（データの圧縮）

この章では ADADCU ユーティリティについて説明します。

機能概要

圧縮解除ユーティリティ ADADCU は、ADACMP、ADAMUP および ADAULD ユーティリティによって生成されたレコードを圧縮解除します。

圧縮解除ユーティリティ ADADCU の出力は、標準オペレーティングシステムのファイルアクセスメソッドを使用しているプログラムに対する入力として使用できます。

また、データ構造を変更したい時やファイルのデータ定義を変更したいときに、圧縮ユーティリティ ADACMP に対する入力としても使用できます。このユーティリティによって作成された圧縮解除出力ファイル（DCUOUT ファイル）が空の場合は、警告メッセージが発行されます。

ADADCU では、ユーティリティ ADAULD と ADACMP の SINGLE オプションで作成されたファイルも圧縮解除されますが、ユーティリティが判別するので、パラメータは必要ありません。

ADADCU には次の機能が装備されています。

- 圧縮解除すると、レコードが完全に復元され、FDT で記述されているフォーマットと長さに戻ります。出力レコードの各マルチプルバリューフィールドまたはピリオディックグループの先頭には、1 バイトのカウントフィールドが付きます。
- LOB フィールドの値は別のファイルに保存することもでき、生成されるファイルの名前は非圧縮レコードに配置されます。
- 複数のフィールドを圧縮解除できます。

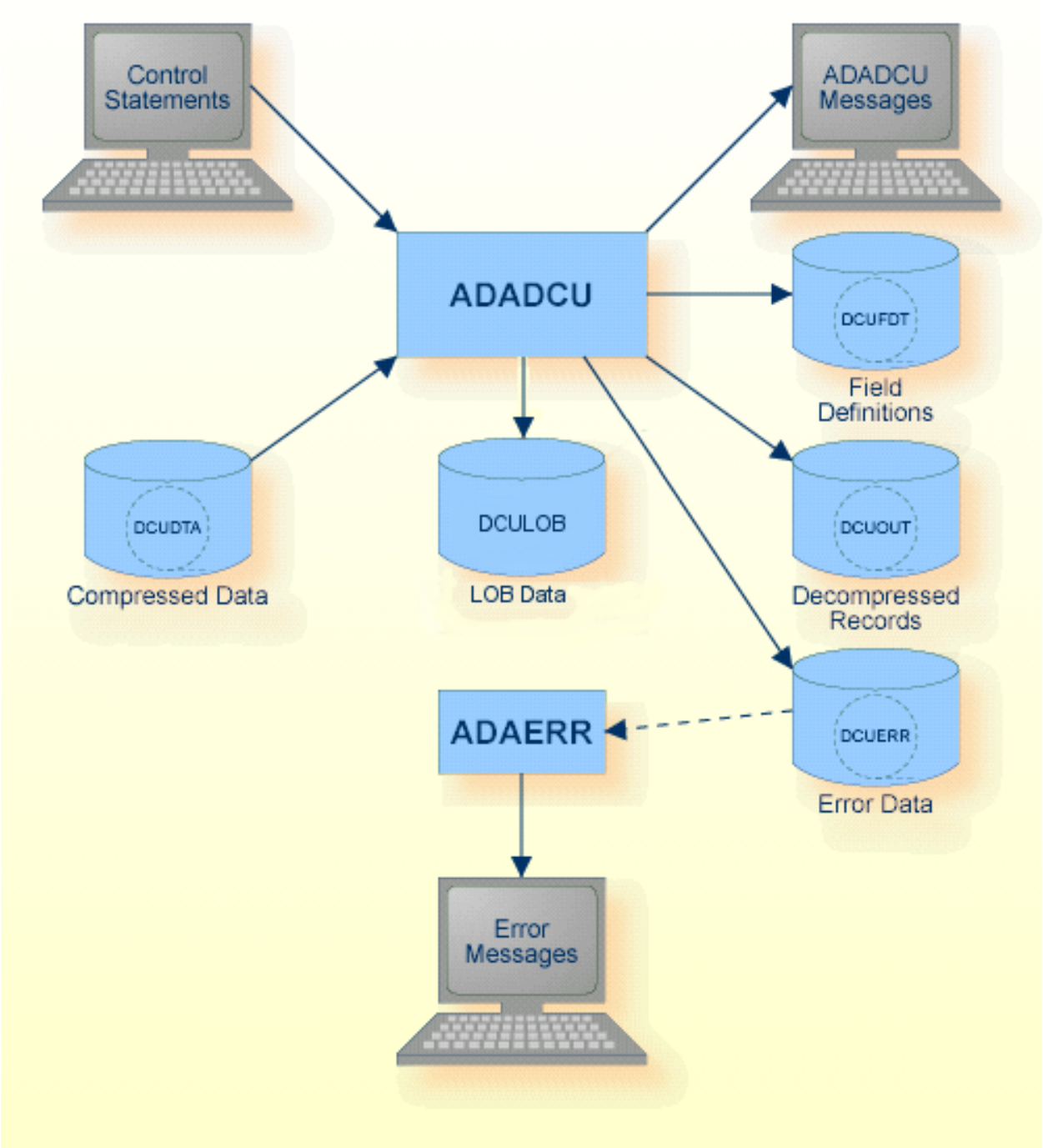
複数のフィールドを圧縮解除する場合、フィールドをレコード内で再配置できます。つまり、レコード構造を次のように変更できます。

- フィールド長を変更できます。
- フィールドフォーマットを変更できます。
- リテラルエレメントまたは空白エレメントを使用して後続の新規フィールドの追加に対してスペースを割り当てできます。

エラーファイルにレコードが書き込まれると、ユーティリティはゼロ以外のステータスで終了します。

このユーティリティは単一機能ユーティリティです。

処理フロー



DCULOB は、LOB の値が別のファイルで保存されるディレクトリです。シーケンシャルファイル DCUDTA および DCUERR は複数エクステントを持つことができます。複数のエクステント

を持つシーケンシャルファイルの詳細については、『Adabas Basics』の「ユーティリティの使用」を参照してください。

データセット	環境 変数／ 論理名	記憶 媒体	追加情報
圧縮 データレコード	DCUDTA	ディスク、テープ（* 注参照）	ADACMP または ADAULD の出力
拒否データ	DCUERR	ディスク、テープ（* 注参照）	ADADCU の出力
出力データ FDT	DCUFDT	ディスク、テープ（* 注参照）	ADADCU の出力 ユーティリティマニュアル
非圧縮レコード	DCUOUT	ディスク、テープ（* 注参照）	ユーティリティマニュアル
LOB データ	DCULOB	ディスク	ユーティリティマニュアル
コントロールステートメント	stdin/ SYS$INPUT		ユーティリティマニュアル
ADADCU メッセージ	stdout/ SYS$OUTPUT		メッセージおよびコード

注意: （*）このシーケンシャルファイルには、名前付きパイプを使用できます（OpenVMS にはない。詳細については、『*Adabas Basics*』の「*ユーティリティの使用*」を参照）。

チェックポイント

このユーティリティはチェックポイントを書き込みません。

制御パラメータ

次のコントロールパラメータを使用できます。

```
D    [NO]DCUFDT

D    [NO]DST

     FDT

     FIELDS {field_specification | FDT},...[END_OF_FIELDS | .]

D    [NO]LOWER_CASE_FIELD_NAMES

D    MAX_DECOMPRESSED_SIZE = number  [K|M]

D    MUPE_C_L = {1|2|4}

     MUPE_OCCURRENCES

D    [NO]NULL_VALUE

D    NUMREC = number

D    RECORD_STRUCTURE = keyword

     SKIPREC = number

D    TARGET_ARCHITECTURE = (keyword[,keyword[,keyword]])

D    [NO]TRUNCATION

     TZ {=|:} [timezone]

D    [NO]USERISN

     WCHARSET = char_set
```

[NO]DCUFDT

```
[NO]DCUFDT
```

このパラメータを DCUFDT に設定すると、非圧縮レコードの FDT 情報がシーケンシャルファイル DCUFDT に書き込まれます。デフォルトは NODCUFDT です。

FIELDS パラメータ（以下を参照）を使用した場合、フィールドは FIELDS に指定した順番でシーケンシャルファイルDCUFDTに書き込まれます。したがって、DCUFDTのフィールドは、オリジナルの FDT とは異なる順番になることがあります。

[NO]DST

```
[NO]DST
```

DSTパラメータは、オプションTZを使用して日付／時刻フィールドに夏時間インジケータが提供されている場合に必要です。夏時間インジケータは、2 バイトの整数値（フォーマット F）として日付／時刻値の後に追加されます。この値には、実際の時間（通常は 0 または 3600）を取得するための、標準時間に追加する秒数が含まれています。

このパラメータは、時刻が標準時刻に戻る前の時間にTZオプションが指定された日付／時刻値が含まれているレコードがある場合に必要です。このパラメータがないと、これらの値はエラーファイルに書き込まれます。

デフォルトは NODST です。

注意:

1. FIELDS パラメータが指定されている場合、DST パラメータは無視されます。この場合、サマータイムインジケータの付いたフィールドに D 要素を指定する必要があります。
2. フォーマット F のサマータイムインジケータに出力できない文字が含まれているため、DST パラメータは、RECORD_STRUCTURE = NEWLINE_SEPARATOR パラメータと互換性がありません。

FDT

```
FDT
```

このパラメータは、圧縮レコードを含むファイルの FDT を表示します。

FIELDS

```
FIELDS {field_specification | FDT},...[END_OF_FIELDS | . ]
```

このパラメータは、FDT のフィールドのサブセットおよび、それらのフォーマットと長さを指定するために使用します。FDT で指定されているすべてのフィールドが、作成される非圧縮レコードに含まれている必要はない、つまり別のフォーマットか長さでフィールドを圧縮解除できるということです。構文と使用方法はフォーマットバッファの場合とほぼ同じです。ただし、非圧縮レコードに LOB 値自体ではなく、LOB 値を含むファイルの名前が含まれている場合に、（LOB 参照の）R 要素も指定できる点だけが異なります。詳細については、『*管理マニュアル*』の「*データのロードとアンロード*」の「*非圧縮データフォーマット*」を参照してください。

指定リストの入力時に、圧縮解除するファイルの FDT を表示する FDT 機能を使用できます。指定リストは END_OF_FIELDS または「.」を入力することで終了または中断できます。「.」オプションは暗黙の END_OF_FIELDS であり、フォーマットバッファ構文と互換性があります。FIELDS または END_OF_FIELDS は常にそれ自体を行に入力する必要がありますが、「.」はそれ自体を行に入力することも、フォーマットバッファ要素の末尾に入力することも可能です。FIELDS の入力によって任意のオプションまたはパラメータの設定後に処理を続行できます。

END_OF_FIELDS パラメータを使用してフィールド定義を消去する場合、LOWER_CASE_FIELD_NAMES パラメータを使用するときに、このパラメータを大文字で指定する必要があります。また、LOWER_CASE_FIELD_NAMES を使用する場合は、FDT パラメータも大文字で指定する必要があります。

例

```
adadcu: fields
adadcu: ; This is a comment line
adadcu: AA,AB,6,A,AC,P ;  - inline comment -
adadcu: AD,AF,CBC,CB1-N . ; implicit END_OF_FIELDS
```

フィールド AA はデフォルトの長さとフォーマットで出力され、フィールド AB は 6 バイトの英数字、フィールド AC はパックのデフォルト長さで出力されます。さらにフィールド AD および AF は、デフォルトの長さとフォーマットで出力され、その後には、フィールド CB の 1 バイトのバイナリマルチプルフィールドカウントと CB のすべてのオカレンスが続きます。

[NO]LOWER_CASE_FIELD_NAMES

```
[NO]LOWER_CASE_FIELD_NAMES
```

LOWER_CASE_FIELD_NAMES が指定されている場合、Adabas フィールド名は、大文字に変換されません。NOLOWER_CASE_FIELD_NAMESが指定されている場合、Adabasフィールド名は、大文字に変換されます。デフォルトは、NOLOWER_CASE_FIELD_NAMES です。

このパラメータは、FIELDS パラメータの前に指定する必要があります。

MAX_DECOMPRESSED_SIZE

```
MAX_DECOMPRESSED_SIZE = number  [K|M]
```

このパラメータは、数字の後の「K」または「M」の仕様に応じて、非圧縮レコードの最大サイズをバイト、キロバイトまたはメガバイト単位で指定します。このパラメータの目的は、誤って巨大な非圧縮レコードファイルが作成されることを防止することです（ファイルに LOB データが含まれることを考慮しなかった場合に発生します）。

デフォルトは、65536 です。これは、最大値でもあります。

注意: このパラメータの正確な定義は、最大非圧縮レコードに必要な入力／出力バッファのサイズです。256 バイトの倍数のみが入力／出力バッファに使用されます。これにより、次の256の倍数に切り上げられる最大非圧縮レコード以上の値（先行する長さフィールドを含む）を指定する必要があります。

MUPE_C_L

```
MUPE_C_L = {1|2|4}
```

データにマルチプルバリューフィールドまたはピリオディックグループが含まれる場合、圧縮解除データにおいて、それらにはMUPE_C_Lバイトの長さのバイナリのカウントフィールドが先行します。

デフォルトは 1 です。

MUPE_OCCURRENCES

```
MUPE_OCCURRENCES
```

このパラメータは、マルチプルフィールドおよびピリオディックグループのリストと、それらの最大オカレンス数を出力するために使用します。圧縮解除データが非常に大きくなることがあるので、出力した情報は重要です。範囲指定が大きすぎる場合、非圧縮レコードのサイズの限界を超えることがあります。

例

ファイルの FDT は次のような圧縮レコードを含みます。

```
1,AA,4,A,NU
1,PE,PE
2,PA,2,A,NU
2,PB,2,A,NU,MU
1,MM,2,U,NU,MU
1,X1,4,B
```

次に示すように、MUPE_OCCURRENCES の出力は、フォームのような形態になります。

```
Name   Max occurrence
--------------------
PE                  4
PB                  8
MM                 12

%ADADCU-I-DCUREC, Number of decompressed records: 5023
%ADADCU-I-DCUIR, Number of incorrect records: 0
```

ファイルは次のように圧縮解除できます。

```
adadcu fields "AA,PA1-4,PB1-4(1-8),MM1-12,P,X1" ↩
```

注意: MU フィールドを含んでいるピリオディックグループのオカレンスが非常に多い場合、レコードは不正であるとみなされ、内部のオーバーフローを起こします。ピリオディックグループを含むこのレコードを圧縮解除することはできません。

[NO]NULL_VALUE

```
[NO]NULL_VALUE
```

このパラメータは、レコードが NC パラメータフィールドおよびそれらのステータス値（S 要素）を含んでいる場合に、標準 FDT でレコードを圧縮解除するために、使用することができます。1つ以上のフィールドに空値がある場合、必要となります。それ以外の場合は、これらのレコードはエラーファイルに格納されます。

例

フィールド AA の FDT エントリが1,AA,2,A,NC の場合には、NULL_VALUE の効果は次のようになります。

- NULL_VALUE：第 1 出力レコード（16 進）00004141（AA に値がある）、第 2 出力レコード（16 進）FFFF2020（AA は空値）。
- NONULL_VALUE：第 1 出力レコード（16 進）4141（AA に値がある）、第 2 出力レコード（16 進）AA は空値。そのため、レコードはエラーファイルに格納されます。

デフォルトは NONULL_VALUE です。

NUMREC

```
NUMREC = number
```

このパラメータは、入力ファイルから読み込んで圧縮解除するレコード数を指定するものです。このパラメータが省略され、SKIPREC が指定されないと、全レコードが処理されます。

例

```
adadcu: numrec = 100
```

100 レコードが読み込まれて圧縮解除されます。

RECORD_STRUCTURE

```
RECORD_STRUCTURE = keyword
```

このパラメータは、論理名 DCUOUT で指定された出力ファイル内で使用されるレコード区切りのタイプを、指定するものです。次のキーワードを使用できます。

キーワード	説明
ELENGTH_PREFIX	DCUOUT ファイルのレコードは、2 バイトの長さフィールド（フィールド長にはフィールド自体の長さは含まれない）によって区切られます。区切り文字がないため、このフォーマットの使用は一切の制限を受けません。
E4LENGTH_PREFIX	非圧縮データファイル内のレコードは、4バイトの長さフィールド（長さフィールド自体の長さを除く）で区切られます。
ILENGTH_PREFIX	DCUOUT ファイルのレコードは、2 バイトの長さフィールド（フィールド長にはフィールド自体の長さも含まれる）によって区切られます。区切り文字がないため、このフォーマットの使用は一切の制限を受けません。
I4LENGTH_PREFIX	非圧縮データファイル内のレコードは、4バイトの長さフィールド（長さフィールド自体の長さを含む）で区切られます。

キーワード	説明
NEWLINE_SEPARATOR	DCUOUT ファイル内のレコードは、改行文字によってて区切られます。 DCUOUT ファイルを ADACMP の入力として使用する場合に、このキーワードを指定できるのは、出力のフィールド値に改行文字が含まれていないときに限ります。つまり、アンパック形式のフィールド、英数字フィールド、Unicode フィールドだけが存在する場合と、英数字フィールドと Unicode フィールドに出力可能な文字のみが含まれる場合がこれに該当します。 このキーワードと USERISN パラメータは相互に排他的です。
RDW	DCUOUT ファイル内のレコードは、FTP *site rdw* オプションを使って IBM ホストへ転送可能な形式にフォーマットされます。
RDW_HEADER	RDW と同じように、HEADER=YES を使用してメインフレームで圧縮できる非圧縮レコードの場合。
VARIABLE_BLOCKED	レコードは、ブロックとして格納されます。各レコードは4バイト（長さバイトの長さを含む）のフィールド長で始まります。

デフォルトは ELENGTH_PREFIX です。

SKIPREC

```
SKIPREC = number
```

このパラメータは、圧縮解除を始める前にスキップされるレコード数を指定するものです。

TARGET_ARCHITECTURE

```
TARGET_ARCHITECTURE = (keyword[,keyword[,keyword]])
```

このパラメータは、出力データレコードの形式（文字セット、浮動小数点フォーマットおよびバイト順）を指定するものです。次のキーワードを使用できます。

キーワードグループ	有効なキーワード
文字セット	ASCII EBCDIC
浮動小数点フォーマット	IBM_370_FLOATING IEEE_FLOATING VAX_FLOATING
バイト順	HIGH_ORDER_BYTE_FIRST LOW_ORDER_BYTE_FIRST

キーワードグループのキーワードを指定しない場合、このキーワードグループのデフォルトは、ADADCU を実行しているマシンのアーキテクチャに対応するキーワードになります。

注意: FDTは、常に ASCII フォーマットで出力されます。

例

出力レコードを IBM 形式に圧縮解除する場合は、ユーザーは次のように指定しなければなりません。

```
TARGET_ARCHITECTURE = (EBCDIC, IBM_370_FLOATING, HIGH_ORDER_BYTE_FIRST)
```

[NO]TRUNCATION

```
[NO]TRUNCATION
```

このパラメータは、英数字フィールド値の桁落ちを有効または無効にするものです。

NOTRUNCATIONがデフォルトです。この場合は、英数字フィールド値が切り詰められたすべてのレコードは、エラーファイルに書き込まれます。

数値の切り詰めはできず、値は標準長または指定長内でなければなりません。数値の桁落ちが発生すると、該当レコードはエラーファイルに書き込まれます。

TZ

```
TZ  {=|:} [timezone]
```

指定されたタイムゾーンは、Olson データベース（**https://www.iana.org/time-zones**）として知られているタイムゾーンデータベースに含まれている有効なタイムゾーン名にする必要があります。タイムゾーンが指定されている場合、このタイムゾーンは、オプションTZを含む日付／時刻フィールドのタイムゾーン変換に使用されます。

デフォルトは、UTC です。これは、オプション TZ を含む日付／時刻フィールドに内部的に保存するために使用されます。変換する必要はありません。

空の値を指定する場合、日付／時刻フィールドが正しいことを確認するチェックマークは付いていません。

注意: タイムゾーン名はファイル名になります。プラットフォームに応じて、これらのファイル名は大文字小文字を区別する場合としない場合があります。また、タイムゾーン名も、プラットフォームに応じて大文字／小文字を区別する場合としない場合があります。

例：

```
tz:Europe/Berlin
```

これは、すべてのプラットフォームに適用されます。

```
TZ=Europe/Berlin
```

この仕様により、TZ は大文字の EUROPE/BERLIN に変換されます。これは、Windows では、ファイル名の大文字／小文字が区別されないため適用されます。ただし、Unix では、Unix ファイル名の大文字／小文字は区別されるため適用されません。

[NO]USERISN

```
[NO]USERISN
```

このパラメータは、それぞれの非圧縮レコードと一緒に ISN を出力するかどうかを指示するものです。レコードに現在割り当てられている ISN を圧縮解除データと一緒に出力するかまたはそれを省略するかどうかをユーザーは指定できます。ユーザーが同じ ISN を持つファイルを再ロードしたい場合には、USERISN オプションを設定する必要があります。

このパラメータは、RECORD_STRUCTURE=NEWLINE_SEPARATOR が指定された場合には指定することができません。

このパラメータが省略されると、ISN は各レコードと一緒には出力されません。

NOUSERISN がデフォルトです。

例

```
adadcu: userisn
```

ISN が各レコードと一緒に出力されます。

WCHARSET

```
WCHARSET = char_set
```

このパラメータは、**http://www.iana.org/assignments/character-sets** に記載されているエンコード名に基づいて、非圧縮ファイルで使用するデフォルトエンコードを指定します。このサイトに記載されているほとんどの文字セットは、ICU（Adabas 国際化サポートで使用）によってサポートされています。

入力および出力データ

ADADCUに対する入力は、アンロードユーティリティADAULDや圧縮ユーティリティADACMPなどによって出力された圧縮レコードが存在するファイルでなければなりません。

ADADCU は、FIELDS 指定にしたがって各入力レコードを圧縮解除し、その結果のレコードを論理名 DCUOUT ファイルに書き込みます。このレコードは可変長形式で書き込まれます。デフォルトでは、レコードは2バイトの長さフィールド（長さフィールド自体はフィールド長に含まれない）によって区切られます。詳細に関しては、このセクションのパラメータ RECORD_STRUCTURE に関する説明を参照してください。

USERISN が指定されると、データレコードの先頭には 4 バイトの 2 進数形式で ISN が付きます。

ADADCU の出力

シーケンシャルファイル DCUFDT（非圧縮レコードのフィールド定義情報）は、ファイル定義ユーティリティ ADAFDU または圧縮ユーティリティ ADACMP に対する入力として使用できます。

拒否されたデータレコード

ADADCU によって拒否されたすべてのレコードは、ADADCU エラーファイルに書き込まれます。このエラーファイルの内容は、ADAERR ユーティリティを使用して表示します。レコードに出力不能文字が含まれているため、標準オペレーティングシステム出力ユーティリティを使用してエラーファイルを出力しないでください。

詳細については、ADAERR ユーティリティの説明を参照してください。

再スタートに関する考慮事項

ADADCUは再スタート機能を備えていません。中断されたADADCUの実行は、最初から再実行しなければなりません。

ADADCU はデータベースを更新しないので、中断された ADADCU の実行を再実行する前にデータベースのステータスに関して考慮する必要はありません。

11 ADADEV（ディスクスペース管理）

この章では ADADEV ユーティリティについて説明します。

注意: このユーティリティは UNIX プラットフォームでのみ使用できます。

機能概要

ADADEVユーティリティは、Adabasによって使用されるディスクスペースを、RAWディスクI/O インターフェイスを介して管理するために、いくつかの機能を提供します。

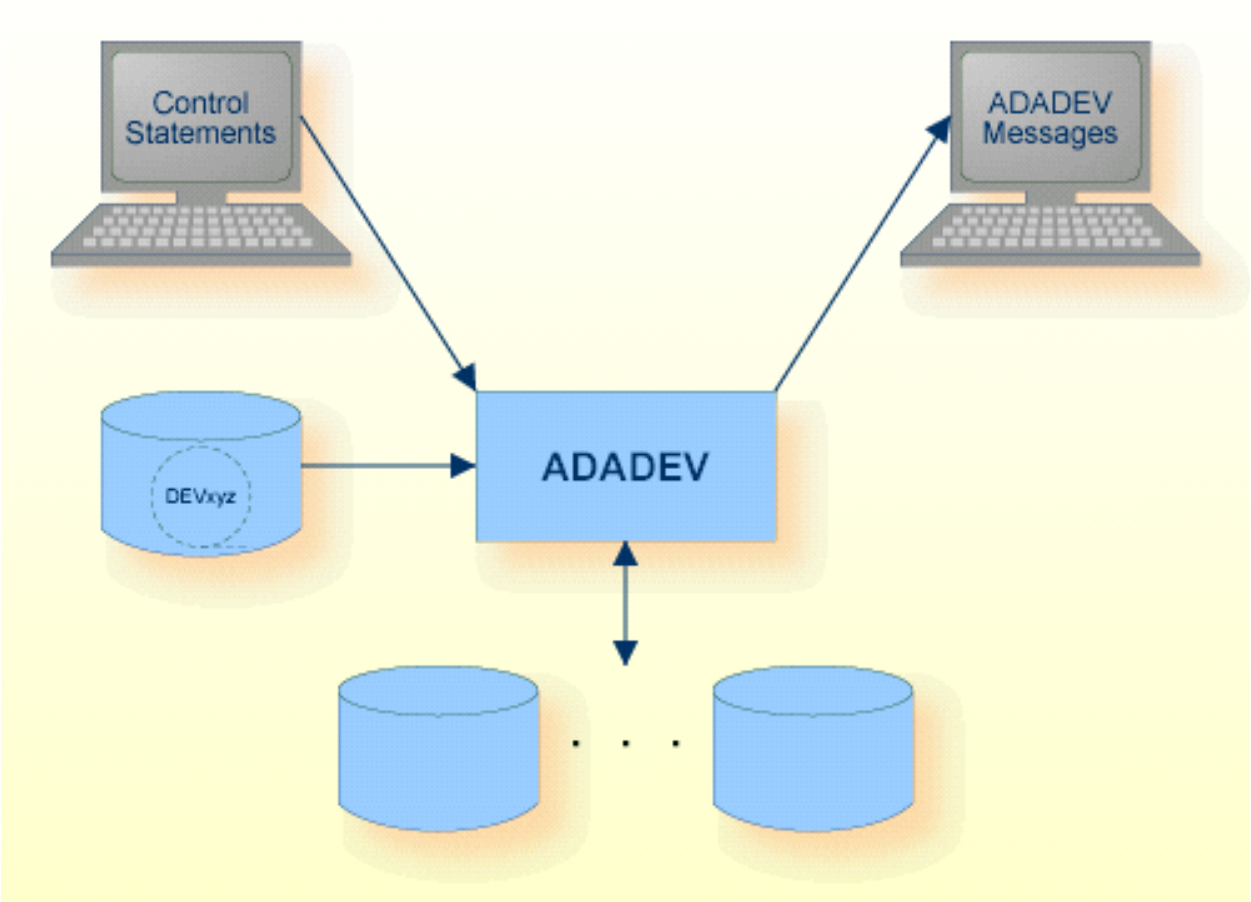
ADADEV は指定されたディスクセクションデバイスに対する READ/WRITE アクセスを必要とします。システムのRAWディスクセクション使用については、『*Adabas インストールガイド*』の「*Adabas のインストール*」を参照してください。

Adabas が使用する各ディスクセクションは、一度 ADADEV で初期化する必要があります。Adabas コンテナファイルまたはシーケンシャルファイル用にスペースをあらかじめ割り当てておく必要はありませんが、そうしておくと役に立つ場合もあります。ADAFRM でコンテナのエクステントを生成する場合、またはAdabasユーティリティでシーケンシャルファイルを生成する場合には、ディスクスペースは自動的に割り当てられます。スペースがあらかじめ割り当てられていない場合、最適アルゴリズムがコンテナのエクステントに使用されます。Adabas シーケンシャルファイルの場合は、その領域が1 MB 以上ある場合、最大有効フリースペースの半分が割り当てられます。そのスペースだけでは足りなくなった場合、フリースペース領域が隣接していれば自動拡張が実行されます。

RAW セクションあたりのコンテナエクステントとシーケンシャルファイルの数は 338 に制限されます。

このユーティリティは多機能ユーティリティです。

処理フロー



データセット	環境 変数	記憶媒体	追加情報
コントロールステートメント	stdin		ユーティリティマニュアル
ADADEV メッセージ	stdout		メッセージおよびコード
Adabas シーケンシャルファイル	DEVxyz（注意 1 を参照）	ディスク、テープ（注意 2 を参照）	

注意:

1. xyz は、PLG、CLG、00n、OUT、ERR、LOG、EXP、DTA、DVT のいずれか。
2. このシーケンシャルファイルには、名前付きパイプを使用できます（詳細については、『Adabas Basics』の「ユーティリティの使用」を参照）。

シーケンシャルファイル DEVPLG、DEVCLG、DEV00n、DEVOUT、DEVERR、DEVEXP、DEVDTA、DEVDVT および DEVLOG は複数エクステントを持つことができます。複数のエクステントを持つシーケンシャルファイルの詳細については、『*Adabas Basics*』の「*ユーティリティの使用*」を参照してください。

チェックポイント

このユーティリティはチェックポイントを書き込みません。

制御パラメータ

次のコントロールパラメータを使用できます。

```
     ALLOCATE = keyword[,START_SECTOR = number]
                [,BLOCKSIZE = numberKB] ,SIZE = number [B|M]

     CHANGE = (keyword,keyword)

     COMBINE = keyword, DESTINATION = string

     COPY = keyword, DESTINATION = string

     DBID = number

     DEALLOCATE = {*|keyword}

     FREE_SPACE

     INITIALIZE

     LAYOUT

D    [NO]MOUNTCHECK

     MOVE = keyword, DESTINATION = string

     NEW_DBID = (container-name, new-dbid)

     REALLOCATE = {*|keyword}

     RESET

     RESIZE

M    SECTION = string
```

```
    UNLOCK = keyword
```

ALLOCATE

```
ALLOCATE = keyword[,START_SECTOR = number]
           [,BLOCKSIZE = numberKB] ,SIZE = number [B|M]
```

この機能は、指定されたキーワードに応じて Adabas コンテナファイルまたは Adabas シーケンシャルファイルに対するスペースを割り当てます。スペースは、開始セクタが指定されていない場合でも、最適なアルゴリズムを用いて割り当てられます。キーワードとして、値 ASSOx、DATAx、WORKx、TEMPx および SORTx（x は 1～最大エクステント数の間の数。『*Adabas Basics*』の「*ユーティリティの使用*」を参照）、PLG、CLG、BCK、BCKOUT、RECOUT、ERR、MUPLOG、MUPTMP、ORDEXP、DTA または DVT を使用できます。DBID パラメータは ASSO、DATA、WORK のファイルまたは PLG、CLG、BCK、BCKOUT、RECOUT に対してスペースが割り当てられる前に指定されなければいけません。

DBA はこの機能を使用して、あらかじめコンテナファイルまたは Adabas シーケンシャルファイルを割り当てることができます。パフォーマンスの上でも、このパラメータを使用することにより、（開始セクタを指定することで）フラグメントを回避し、今後作成するデータベースに対するスペースを確保することができます。

ADADBM または ADAFRM がディスクセクションでのコンテナファイルの作成に使用される場合、あらかじめ割り当てられたスペースは、使用可能であれば確保されます。あらかじめ割り当てられたスペースがなければ、最適なアルゴリズムを使用して領域確保が行われます。コンテナファイルにあらかじめ割り当てるサイズと、コンテナファイルの作成時のサイズは同にする必要があります。

START_SECTOR = number

このパラメータは、割り当てを開始するセクタを指定するものです。

BLOCKSIZE = numberKB

ブロック単位で割り当てを行っている場合、コンテナブロックサイズを指定するために、32 KB までの値を使用することができます。デフォルトは、ASSO コンテナファイルに対して 2 KB、他のすべてのコンテナファイルに対して 4 KB、および Adabas のシーケンシャルファイルに対して 1 KB です。

SIZE = number [B|M]

このパラメータは、割り当てられるエリアをブロック単位またはメガバイト単位で指定するものです。数値に"B" を付加するとサイズはブロック単位になります。デフォルトでは、サイズはメガバイト単位で認識されます。

CHANGE

```
CHANGE = (keyword, keyword)
```

このパラメータは、第1キーワードに指定されたコンテナファイルまたは Adabas シーケンシャルファイルのタイプを第 2 キーワードに指定したタイプに変更します。

キーワードの組み合わせは次のとおりです。

第 1キーワード	第 2キーワード
WORK1	TEMP1、SORT1、SORT2
TEMP1	WORK1、SORT1、SORT2
SORT1	WORK1、TEMP1、SORT2
SORT2	WORK1、TEMP1、SORT1
DTA	MUPLOG
MUPLOG	DTA
RECOUT	PLG（* 注意を参照）
BCKOUT	BCK（* 注意を参照）

注意: （*）RECOUT[BCKOUT] が PLG.n[BCK00n] のコピーであれば、新しいシーケンシャルファイルの名前は対応する名前になります。

DBID パラメータは、キーワード WORK1、RECOUT、BCKOUT が使用される前に設定する必要があります。

自動再スタートが保留状態でなければ、任意のデータベースの WORK1 コンテナを SORT または TEMP コンテナに変更することができます。

COMBINE

```
COMBINE = keyword, DESTINATION = string
```

このパラメータは複数の Adabas シーケンシャルファイルのエクステントを 1 つのエクステントに結合します。キーワードは PLG.n、CLG.n、BCK00n、BCKOUT、RECOUT.n、ERR、MUPLOG、ORDEXP、DTA、DVT の値を取ることができます。またキーワードの後に、エクステントのラベル（m）を続けることができます。場合によっては、DBID パラメータの設定が必要になります（詳細については ALLOCATE を参照）。

COMBINE パラメータは任意のエクステントから開始できますが、最後のエクステント（エンドオブファイル）で終了しなければなりません。後続のエクステントは対話方式で定義することも、既定の環境変数で定義することもできます。DEVxyzを設定している場合、最初のエクステントはデフォルトのパス名から取られます。DEVxyz を設定していない場合、Adabas は最初のエクステントを探すために現在のディスクセクションを見ます。

"string" は、ディスクセクションの RAW インターフェイスに相当するデバイスファイルのパス名、テープデバイスのパス名、ファイルシステムの非エクステントファイルの名前（パス名）、またはピリオドです。

詳細については、『*Adabas Basics*』の「*ユーティリティの使用*」を参照してください。

例 **1**：

データベース 100 の PLOG 2 は、同じディスクセクション /dev/rdsk/c4d0s2 にすべて割り当てられる 3 エクステントで構成されます。

環境変数	設定
DEVPLG	/dev/rdsk/c4d0s2 /dev/rdsk/c4d0s2

ADADEV コマンドは次のとおりです。

```
adadev: section=/dev/rdsk/...
adadev: dbid=100
adadev: combine=plg.2(1)
adadev: destination=PLOG_2
```

PLOG 2 のすべてのエクステントは 1 つのファイルにまとめられ、PLOG_2 という名前で、ファイルシステムに書き込まれます。DEVPLG には同じセクションを 2 つ指定して、いつでも設定を解除したり、元に戻したりできるようにする必要があります。

例 **2**：

この例では、データベース 100 の PLOG 3 は 9 つのエクステントから構成されており、5 から 9 のエクステントは 4 つのディスクセクションにわたって配分されています。

環境変数	設定
DEVPLG	/dev/rdsk/c3d0s2 # contains PLG.3(5) and PLG.3(9) EOF
DEVPLG2	/dev/rdsk/c4d0s2 # contains PLG.3(6)
DEVPLG3	/dev/rdsk/c5d0s2 # contains PLG.3(7)
DEVPLG4	/dev/rdsk/c6d0s2 # contains PLG.3(8)

ADADEV コマンドは次のとおりです。

```
adadev: section=/dev/rdsk/...
adadev: dbid=100
adadev: combine=plg.3(5)
adadev: destination=PLOG.3(5)
```

PLOG 3 の 5 から 9 のエクステントはエクステント番号 5 と EOF のラベルを付けて PLOG 3 内の 1 つのエクステントに格納されます。結合した PLOG のエクステントは PLOG.3(5) という名前で現在のディレクトリ内に作成されます。

COPY

```
COPY = keyword, DESTINATION = string
```

この機能は、現在の位置から指定した位置（DESTINATION=文字列）にコンテナファイル（ASSO1、DATA1 など）またはシーケンシャルファイル（RECOUT、DVT など）をコピーします。

コンテナファイルに対して有効なキーワードは ASSOx、DATAx および WORKx です。x は『*Adabas Basics*』の「*ユーティリティの使用*」の説明にあるように、1～最大エクステント数の範囲の数字です。

Adabas シーケンシャルファイルのキーワードは PLG.n、CLG.n、BCK00n、BCKOUT、RECOUT.n、ERR、MUPLOG、ORDEXP、DTA、DVT です。Adabas シーケンシャルファイルのキーワードの n はエクステント番号を示しています。

コンテナファイルをコピーする場合、その前にまずコピー元の場所として適切な環境変数（例えば ASSO1）を設定する必要があります。

現在のディスクセクション内に Adabas シーケンシャルファイルをコピーする場合、SECTION パラメータの指定（例えば SECTION=/dev/c5d0s2）に従って、環境変数 DEVxyz（xyz は PLG、CLG、00n、OUT、ERR、LOG、EXP、DTA、DVT の値を取る）を Adabas シーケンシャルファイルまたはデバイスファイルに設定する必要があります。

DESTINATION キーワードには RAW ディスクセクションのパス名、テープデバイスのパス名、ファイルシステム内の非エクステントファイルのパス名、または現在のディスクセクション内にファイルをコピーすることを示す名前付きパイプまたはピリオド（.）を指定します。

場合によっては、COPY 機能を実行する前に、個別のファイルであることを明示するために DBID を指定する必要があります。DBID の指定が必要なファイルのリストについては、ALLOCATE 機能パラメータに関する説明を参照してください。

例：

```
adadev: section=/dev/rlv02
adadev: dbid=23
adadev: copy=WORK1, destination=/FS/fs0395/SAG/ada/db023/WORK1.023
```

WORK1 は RAW デバイスからファイルシステムにコピーされます。

環境変数 WORK1 は、値として /FS/fs0395/SAG/ada/db023/WORK1.023 を持ちます。

```
adadev: section=/dev/rlv02
adadev: dbid=023
adadev: copy=WORK1, destination=.
```

WORK1 はファイルシステムから現在選択している RAW デバイスにコピーされます。

DBID

```
DBID = number
```

このパラメータは、ASSO、DATA、WORK のコンテナファイルのデータベースまたは PLG、CLG、BCK、BCKOUT、RECOUT の Adabas シーケンシャルファイルのデータベースを指定します。

DEALLOCATE

```
DEALLOCATE = { * | keyword }
```

この機能は、指定されたキーワードに応じて、Adabas コンテナファイル、Adabas シーケンシャルファイル、または任意のデータベースの全エクステントの割り当てを解除（DEALLOCATE=*）します。キーワードは、ASSOx、DATAx、WORKx、TEMPx、SORTx、NUCTMPx、および NUCSRTx（x は『*Adabas Basics*』の「*ユーティリティの使用*」の説明にあるように、1～最大エクステント数の範囲の数字）、PLG、PLG.n、PLG*、CLG、CLG.n、CLG*、BCK、BCK00n、BCK*、BCKOUT、RECOUT、RECOUT.n、ERR、MUPLOG、MUPTMP、ORDEXP、DTA または DVT の値を取ります。Adabas シーケンシャルファイルのキーワードの後には、エクステントラベル（m）または（*）に続けることができます。場合によっては、DBID パラメータの設定が必要になります（詳細については ALLOCATE を参照）。

DBA は、このパラメータを使用して、コンテナファイルの割り当てを解除できます。例えば、データベースが必要でなくなった場合や、PLOG エクステントがテープにセーブされた場合です。割り当て解除されたスペースは空きスペースとして管理され、他のコンテナに割り当てられます。

注意: ADADBM REDUCE_CONTAINER または ADADBM REMOVE_CONTAINER を使用して、RAW セクションに保存されているデータベースに必要なスペースを削減すると、RAW セクション内の対応するスペースが自動的に割り当て解除されます。この目的のために ADADEV DEALLOCATE を実行する必要はありません。

注意: ADADEV DEALLOCATE を使用する際は注意してください。ADADEV DEALLOCATE は、コンテナがまだ必要かどうかをチェックしません。使用中のデータベースコンテナを指定した場合、対応するデータベースが破損します。

FREE_SPACE

```
FREE_SPACE
```

この機能は、現在のディスクセクションのどこに空きスペースがあるのかを表示します。この機能は、LAYOUT 機能のサブセットです。デフォルトのブロックサイズ単位での割り当てをより簡単にするために、この機能は、空き領域の大きさを 2 KB および 4 KB 単位でも表示します。

INITIALIZE

```
INITIALIZE
```

割り当て済みの領域と空き領域を管理するために、ディスクセクションの先頭にある数個のセクタが使用されます。この機能は、この管理部分を初期化します。Adabas によって使用される各ディスクセクションは、あらかじめ初期化されていなければなりません。Adabas を利用する目的でディスクセクションがアクセスされるときに、最初の段階で管理部分が検証されます。ディスクセクションは、この検証が失敗した場合にのみ初期化されます。

LAYOUT

```
LAYOUT
```

この機能は、ディスクセクション使用状況の要約情報を表示します。この機能を実行すると、コンテナエリア、Adabas シーケンシャルエリアだけでなく空きスペースもリストさます。また、コンテナエリアのステータス（割り当て済みまたは作成済み）の表示も行います。Adabas シーケンシャルファイルが作成中のステータスであることもあります。これは Adabas シーケンシャルファイルが割り当てられたスペースを使い尽くしても、フリーな状態のものがあり、増大しているからです。

電源障害が発生した場合には、TEMP、SORT コンテナは読み込みまたは書き込みがロックされることもあります（スタータスは rlocked または wlocked）。コンテナファイルのロック解除については、このセクションの UNLOCK 機能に関する説明を参照してください。

[NO]MOUNTCHECK

```
[NO]MOUNTCHECK
```

MOUNTCHECK を使用する場合、ADADEV は、ファイルシステムが、ADADEV SECTION パラメータに指定されたディスクにマウントされるかをチェックします。ファイルシステムがマウントされると、ADADEV は終了します。SECTION パラメータを使用する前に、NOMOUNTCHECK を指定すると、このチェックはスキップすることができます。

デフォルトは MOUNTCHECK です。

MOVE

```
MOVE = keyword, DESTINATION = string
```

MOVE 機能は基本的に COPY 機能と同じですが、違いはソースファイルを移動することです（詳細については **COPY** の説明箇所を参照）。

NEW_DBID

```
NEW_DBID = (container-name, new-dbid)
```

この機能は、RAW セクション内の Adabas ファイルの DBID を変更するために使用します。例えば、データベース番号を変更したり、新しい番号のデータベースに既存の PLOG を適用したりするときなどです。

コンテナ名に次を指定できます。

- 関連データベース ID を持つすべてのコンテナ／シーケンシャルファイル（ASSO、DATA、PLG など）
- セクション内のすべてのファイルを変更するための PLOG や CLOG とワイルドカード文字の組み合わせ（PLG*、CLG*）
- 任意の PLOG（PLG.175 など）を指定して、シーケンシャルファイルのエクステント（PLG.175(1) など）があるとき、これらのオカレンスはすべて変更されます。

次をコンテナ名に指定することはできません。

- DBID を持たないコンテナ（SORT、DTA など）
- 特定の PLOGファイルエクステント（PLG.175(1) など）

例

次の例は、DBID を 1 から 2 に変更する方法です。

```
adadev section=xxx dbid=1 new_dbid=(asso1,2)
adadev section=xxx dbid=1 new_dbid=(plg.175,2)
adadev section=xxx dbid=1 new_dbid=(plg\*,2)
```

REALLOCATE

```
REALLOCATE = { * | keyword }
```

この機能は、スペースを割り当て解除し、単一ステップで直接割り当てます。キーワードは、DEALLOCATE パラメータと同じ値を取ることができます。50 KB 以上の Adabas シーケンシャルファイルのエリアは常に割り当て解除されます。

コンテナファイルを割り当て解除した後、それを再び同じ位置に割り当てる場合に、この機能を使用すると短時間で実行できます。これは特にデータベースを作成中に ADAFRM が異常終了し、いくつかのコンテナファイルがすでに作成され、ある開始セクタから始まる領域がすでに確保されている場合に便利です。

ディスクセクション内のコンテナごとにフラグが存在します。このフラグは、コンテナまたは Adabas シーケンシャルファイルのどちらが実際に作成されているのか、またはスペースがそのコンテナだけに割り当てられているのかどうかを表します。セキュリティの理由から、既存のコ

ンテナまたはAdabasシーケンシャルファイルは再作成するだけでは上書きすることができません。まず、割り当て解除ないし再割り当てを行わなければいけません。

RESET

```
RESET
```

セクションが初期化される場合、現在のディスクセクションの管理部分はバイナリのゼロに設定されます。この機能は、初期化対象のディスクセクションが、初期化済みのディスクセクションと開始セクタが重複しているディスクセクションを処理化する場合に使用します。

RESIZE

```
RESIZE
```

新規にディスクセクションを作成する場合に、そのディスクセクションと現在のディスクセクションの開始セクタと重複しているのであれば、この機能を使用したときに、新規セクションのサイズに合わせてセクションのサイズが更新されます。新規セクションは現在のセクションよりも大きくても小さくてもかまいません。より大きくする場合、セクション末尾のフリースペースが増加されます。小さくする場合、セクション末尾の既存のフリースペースが縮小されます。

SECTION

```
SECTION = string
```

このパラメータは、使用対象となるディスクセクションを選択するためのものです。該当ディスクセクションの RAW I/O インターフェイスを表すデバイスファイルの名前（パス名）を指定する必要があります。

UNLOCK

```
UNLOCK = keyword
```

このパラメータは異常終了したユーティリティのコンテナをロック解除するために使います（kill-9 など）。キーワードは値 TEMP1、SORT1、または SORT2 を取ることができます。

電源障害が発生すると、コンテナファイルはロックされますが、ファイルを使用する必要がある場合には、UNLOCK 機能を使用して、ファイルのロックを解除する必要があります。Adabasシーケンシャルファイルが1つ以上の物理エクステントから構成されている場合、最後のエクステントのステータスフィールドに EOF を付けなければいけません。

12 ADAELA（Event Analytics 管理）

この章では ADAELA ユーティリティについて説明します。

機能概要

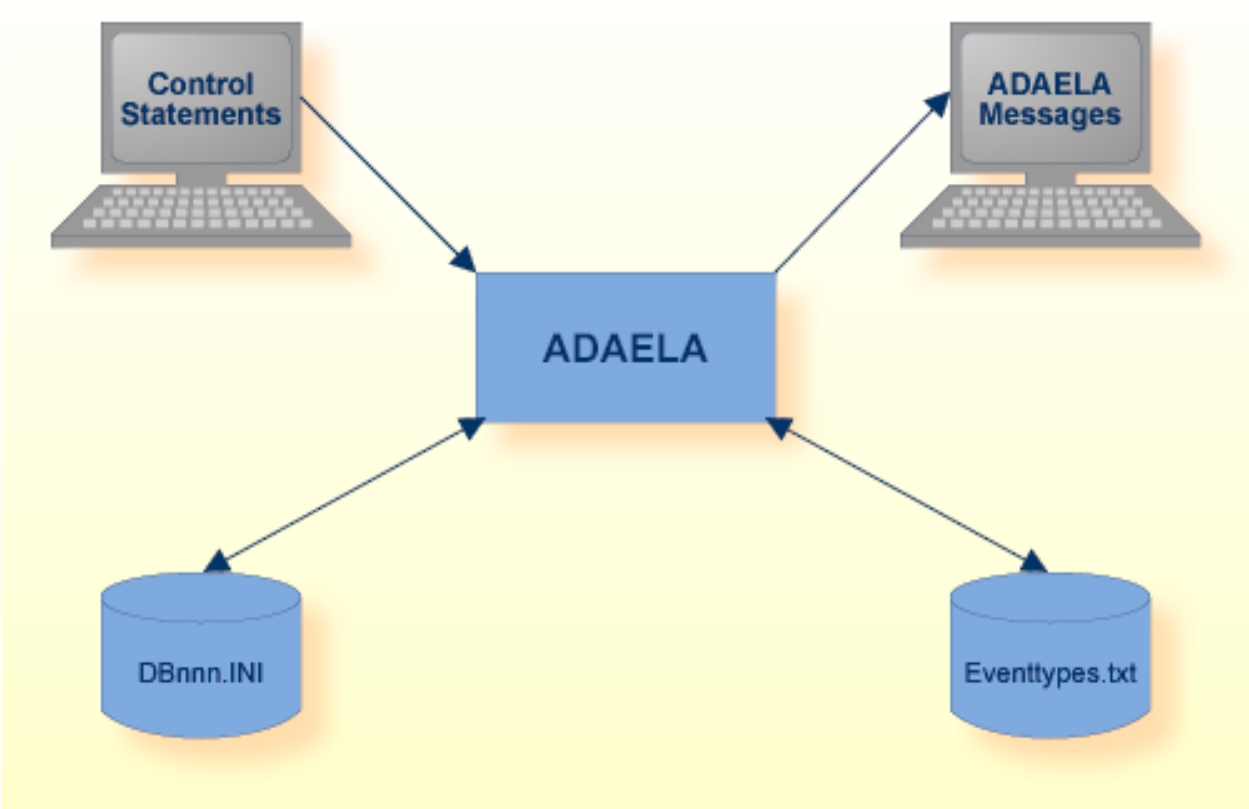
管理ユーティリティ ADAELA は、Event Analytics アドオンを構成します。Adabas ニュークリアスをアクティブにする必要はありませんが、コンフィグレーションを変更した場合は、それらのコンフィグレーションをアクティブにするために、ニュークリアスを再起動する必要があります。

ユーティリティの1回の実行中に、複数のイベントタイプの追加または削除が可能です。また、1 回の実行でイベントタイプの追加と削除の両方を実行することもできます。複数の「add=eventtype,…」や「remove=eventtype,…」ステートメントが発行される場合、各ステートメントはキーワード `END_OF_EVENTTYPE` で終了する必要があります。

注意: すでにコンフィグレーションが存在する場合は、上書きされます。

このユーティリティは多機能ユーティリティです。

処理フロー



データセット	環境 変数／ 論理名	記憶 媒体	追加情報
データベースコンフィグレーション		DBnnn.ini	
イベントタイプストア		eventtypes.txt	
コントロールステートメント	stdin/SYS$INPUT		「制御パラメータ」を参照してください。
ADAELA メッセージ	stdout/SYS$OUTPUT		メッセージおよびコード

チェックポイント

このユーティリティはチェックポイントを書き込みません。

制御パラメータ

次のコントロールパラメータを使用できます。

```
ADD = SERVER | NUCELG | EVENTTYPE | FILTER
           [ ,server_keywords |
             ,nucelg_keywords |
             ,eventtype_keywords |
             ,filter_keywords]

M    DBID = number

    DISABLE

    DISPLAY = CONFIGURATION | EVENTTYPES

    ENABLE

    REMOVE = SERVER | NUCELG | EVENTTYPE, NAME=string | FILTER
```

ADD

ADD = SERVER

```
ADD = SERVER
D            [,HOST=string]
D            [,PORT=number]
D            [,RECONNECT_TIMEOUT=number]
D            [,RETRY=number]
D            [,ON_ERROR=keyword]
```

このパラメータは、Analytics Server のコンフィグレーションを DBnnn.INI ファイルに追加します。ファイルへのイベントの書き込みと並行して Analytics Server を使用することはできません（「NUCELG」を参照）。NUCELG ファイルのコンフィグレーションがすでに存在する場合、それは削除されます。

HOST=string

Analytics Server が実行されているホスト名。デフォルトは *localhost* です。これは、最高のパフォーマンスを得るためにサーバーが Adabas ニュークリアスと同じホストで実行されることを前提としています。

PORT=number

Analytics Server の TCP/IP 待ち受けポート番号。デフォルト値は 6521 です。ポート番号は、Analytics Server で構成されたポート番号と一致する必要があります。

RECONNECT_TIMEOUT=number

このパラメータは、Adabas ニュークリアスと Analytics Server の接続が失われた場合の、Adabas ニュークリアスが Analytics Server への再接続の試行を待機する時間（秒単位）を指定します。デフォルトは 1 です。

注意: 再接続の試行には時間がかかり、Adabas のパフォーマンスに悪影響を及ぼします。

RETRY=number

このパラメータは、Adabas ニュークリアスと Analytics Server の接続が失われた場合の、再接続の試行回数を指定します。デフォルトは 0 で、「継続的に試行」を意味します。

注意: 試行回数の上限が 0 ではなく、再接続が成功することなく再接続試行回数の上限に達すると、イベントロギングは無効になります。

ON_ERROR=keyword

このパラメータは、Analytics Server でエラーが発生した場合の Adabas ニュークリアスの動作を定義します。デフォルトは *IGNORE* です。「keyword」には、次のいずれかを指定できます。

キーワード	説明
ABORT	Analytics Server でエラーが発生すると、Adabas ニュークリアスは異常終了します。
IGNORE	Adabas ニュークリアスは、Analytics Server への再接続を試行します。

ADD=NUCELG

```
ADD = NUCELG
D             [,TIME_SWITCH=number]
D             [,EVENT_SWITCH=number]
D             [,ELGFILE=path]
```

このパラメータは、イベントをファイルに書き込むためのコンフィグレーションを追加します。ファイルへのイベントの書き込みと並行して Analytics Server を使用することはできません。Analytics Server のコンフィグレーションがすでに存在する場合、それは削除されます。

TIME_SWITCH=number

新しいログファイルを開始するまでの経過時間（秒単位）。デフォルト値は 0 です。これは、一定時間でログファイルが切り替えられないことを意味します。

EVENT_SWITCH=number

このイベント数が発生した後で、新しいログファイルが開始されます。デフォルト値は0です。これは、一定数のイベントでログファイルが切り替えられないことを意味します。

ELGFILE=path

ログファイルの完全修飾パス名です。デフォルトは *$ADADATADIR/dbnnn/NUCELG* または *%ADADATADIR%\dbnnn\NUCELG* です。番号の接尾辞は自動的に追加されます。

ADD=EVENTTYPE

```
ADD = EVENTTYPE ,NAME=string
                ,AREA=keyword
                ,FIELDS=(keyword [,keyword] …)
D              [,FILE=path]
D              [,END_OF_EVENTTYPE]
```

このパラメータは、イベントタイプ定義をイベントタイプファイルに追加します。

NAME=string

イベントタイプの名前です。

FILE=PATH

イベントタイプを格納するファイルの完全修飾パス名です。

AREA=keyword

イベントが作成される領域。「keyword」には、次のいずれかを指定できます。

キーワード	説明
POST_COMMAND	このイベントは、Adabas コントロールブロックに基づいています。通常の Adabas コマンドが処理された後で、コマンドコードに依存するイベントが送信されます。 このエリアでは、イベントで使用可能なフィールドも定義します。

FIELDS = (keyword [,keyword] …)

イベントタイプのフィールドのリストです。次のフィールドを利用できます。

キーワード	エリア	タイプ、長さ	説明
event_timestamp	POST_COMMAND	整数値、8	イベントが生成されたときのタイムスタンプ
Dbid	POST_COMMAND	整数値、4	データベース ID
file_number	POST_COMMAND	整数値、4	Adabas ファイル番号
command_code	POST_COMMAND	文字列、2	Adabas コマンドコード
response_code	POST_COMMAND	整数、4	ニュークリアスレスポンスコード
Pid	POST_COMMAND	整数値、8	プロセス ID
Isn	POST_COMMAND	整数値、8	ISN
hostname	POST_COMMAND	文字列、8	ホスト名
user_id	POST_COMMAND	文字列、8	ユーザー ID
Tsid	POST_COMMAND	2 進数、8	ユニーク ID
natapplication	POST_COMMAND	文字列、8	Natural アプリケーション名
natprogram	POST_COMMAND	文字列、8	Natural プログラム名
natlevel	POST_COMMAND	整数値、4	Natural コールレベル
natcount	POST_COMMAND	整数値、8	最後の IO 以降の Adabas コールの数
natexec	POST_COMMAND	整数値、8	Natural オブジェクトが実行された回数
natuser	POST_COMMAND	文字列、8	Natural ユーザー ID
natstatement	POST_COMMAND	文字列、4	Natural ステートメント番号
Natlib	POST_COMMAND	文字列、8	Natural ライブラリ名
natrpcclientuid	POST_COMMAND	文字列、8	Natural RPC クライアントユーザー ID
natrpcid	POST_COMMAND	文字列、16	Natural RPC ID
natrpcconvid	POST_COMMAND	文字列、16	Natural RPC 会話 ID
natsecgroup	POST_COMMAND	文字列、8	Natural セキュリティグループ
additions1	POST_COMMAND	2 進数、8	アディション 1 フィールド
duration	POST_COMMAND	整数値、4	コマンド期間（マイクロ秒単位）
command_opt1	POST_COMMAND	文字、2	コマンドオプション 1 フィールド
command_opt2	POST_COMMAND	文字、2	コマンドオプション 2 フィールド

END_OF_EVENTTYPE

複数のイベントタイプを追加する場合、またはイベントタイプを追加して1つ以上のイベントタイプを削除する場合は、キーワード END_OF_EVENTTYPE を指定する必要があります。

DBID

```
DBID = number
```

このパラメータは、使用対象となるデータベースを選択するためのものです。

DISABLE

```
DISABLE
```

このパラメータは、イベントロギング機能を無効にします。

DISPLAY

```
DISPLAY = CONFIGURATION | EVENTTYPES
```

このパラメータは、イベントロギングに関する情報を表示します。次のキーワードを指定できます。

キーワード	説明
CONFIGURATION	イベントロギングのコンフィグレーションを表示します。情報は *DBnnn.INI* ファイルから取得されます。
EVENTTYPES	構成されているすべてのイベントタイプを表示します。

ENABLE

```
ENABLE
```

このパラメータは、イベントロギング機能を有効にします。

REMOVE

```
REMOVE = SERVER | NUCELG | EVENTTYPE, NAME=string | FILTER
```

REMOVE = SERVER

このパラメータは、サーバーコンフィグレーションを削除します。セクション *TARGET_SERVER* が *DBnnn.INI* ファイルから削除されます。

REMOVE = NUCELG

このパラメータは、NUCELG コンフィグレーションを削除します。

REMOVE = EVENTTYPE, NAME=string [,END_OF_EVENTTYPE]

このパラメータは、指定した名前のイベントタイプをイベントタイプファイルから削除します。

複数のイベントタイプを削除する場合、またはイベントタイプを削除して1つ以上のイベントタイプを追加する場合は、キーワード END_OF_EVENTTYPE を指定する必要があります。

13 ADAELP（イベントログレポート）

この章では「ADAELP」ユーティリティについて説明します。

機能概要

ADAELP ユーティリティは、Adabas Analytics が作成したイベントログからのイベントを出力します。

注意: イベントログを書き込むには、イベントロギングを有効にし、レプリケーションユーザー出口をロードする必要があります。詳細については、『ユーティリティ』ドキュメントの「*ADAELA（Event Analytics 管理）*」を参照してください。

ADAELP パラメータ USER_ID、HOSTNAME、および EVENT_TIMESTAMP は、イベントログ内のイベントのサブセットを選択します。

対話モードでは、キーワード LIST が入力されたときに、選択したイベントが ADAELP によって表示されます。パラメータを指定して ADAELP を呼び出すと、選択したイベントがすぐに表示されます。

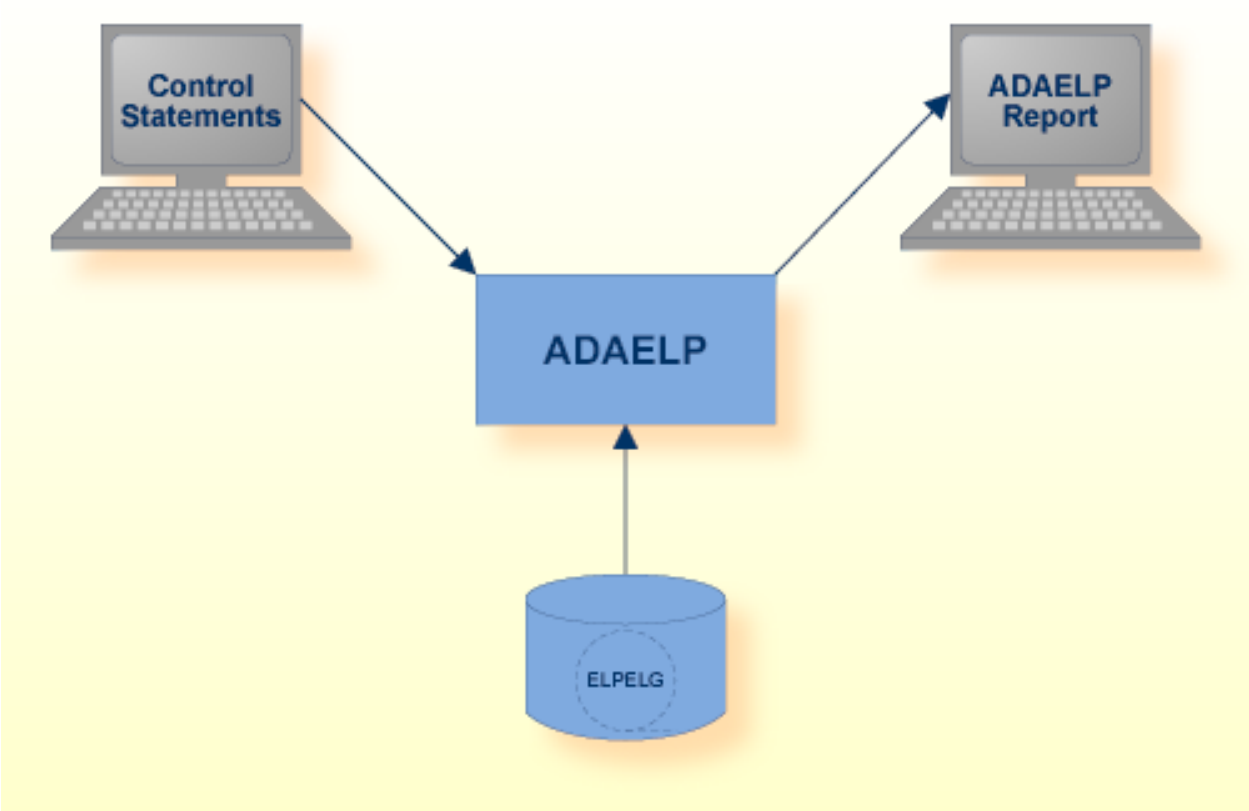
イベントは次のように表示されます。イベントタイプの1行目の後に、対象のイベントのフィールドデータを含む行が続きます。イベントの表示は、最後の行でイベントタイプが繰り返されて終了します。

出力例

```
start read event
  event_timestamp=16-JUL-2015 11:51:01.977020
  dbid=163
  file_number=1
  command_code=L5
  response_code=0
  isn=993
  pid=6772
  hostname=PCST01
  user_id=st
  tsid=68 ba 56 02 fb 1a 05 00
end read event
```

このユーティリティは単一機能ユーティリティです。

処理フロー



データセット	環境変数／論理名	記憶媒体	追加情報
イベントログ	ELPELG	ディスク	
コントロールステートメント	stdin		「制御パラメータ」を参照してください。
ADAELP レポート	stdout		

チェックポイント

このユーティリティはチェックポイントを書き込みません。

制御パラメータ

次のコントロールパラメータを使用できます。

```
D    DBID = number

    EVENT_TIMESTAMP = ([absolute-date][,[absolute-date]])

    HOSTNAME = string

    LIST

    USER_ID = string
```

DBID

```
DBID = number
```

このパラメータは、イベントログが書き込まれたデータベースのデータベース ID を指定します。

EVENT_TIMESTAMP

```
EVENT_TIMESTAMP = ([absolute-date][,[absolute-date]])
```

このパラメータは、任意の日付文字列によって指定された範囲にあるログレコードを選択します。指定する日付文字列は、次の絶対日付および時刻の形式をとらなければいけません。

```
dd-mmm-yyyy[:hh:mm:ss[.mmmmmm]]
```

日付と時刻の仕様の上位桁のゼロは、省略される可能性があります。指定されていない数字は、0 に設定されます（例：28-jul-2015 は 28-jul-2015:00:00:00.000000 と同じです）。

デフォルトでは、すべてのログレコードが選択されます。

例：

```
adaelp: event_timestamp = 8-aug-2015
```

event_timestamp 8-AUG-2015 00:00:00 のイベントが選択されます。

```
adaelp: event_timestamp = (8-aug-2015:12,)
```

8-AUG-2015 12:00:00 以後の time_stamp を持つ全イベントが選択されます。

```
adaelp: event_timestamp = (,8-aug-2012:12:34)
```

8-AUG-2015 12:34:00 より前の time_stamp を持つ全イベントが選択されます。

```
adaelp: event_timestamp = (16-JUL-2015 11:51:01.977020, 16-JUL-2015 11:51:02.177000)
```

16-JUL-2015 11:51:01.977020 から 16-JUL-2015 11:51:02.177000 の範囲の event_timestamp を持つ全イベントが選択されます。

HOSTNAME

```
HOSTNAME = string
```

このパラメータは、「string」で指定されたホスト名を持つレコードをすべて選択します。パラメータ値の長さは、8 文字に制限されています。

LIST

```
LIST
```

このパラメータは、DBID、EVENT_TIMESTAMP、HOSTNAME、および USER_ID パラメータで選択されたイベントの一覧を表示します。

USER_ID

```
USER_ID = string
```

このパラメータは、「string」で指定されたユーザー ID を持つ全イベントを選択します。パラメータ値の長さは、8 文字に制限されています。

複数選択条件の指定

複数の選択条件を指定する場合、それらは論理 AND で結合しなければなりません。

```
event_timestamp=(8-aug-2015:12:34,), user_id = guest, hostname  = machine3
```

8-aug-2015:12:34 より後で、user_id = guest かつ hostname = machine3 の全イベントを選択します。

14 ADAERR（エラーファイルレポート）

この章では ADAERR ユーティリティについて説明します。

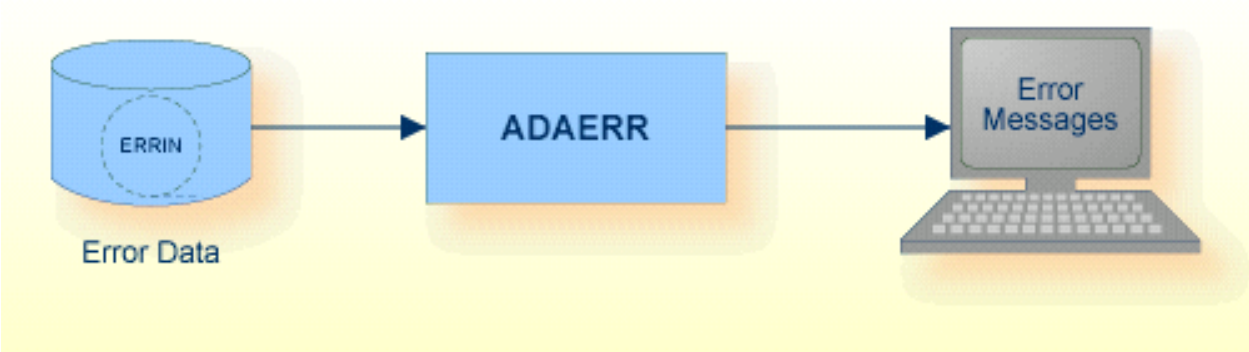
機能概要

ADAERR ユーティリティは、次のユーティリティによって生成されたエラーファイルの内容を表示するものです。

- ADACMP
- ADADCU
- ADAINV
- ADAMUP
- ADAREC

このユーティリティは単一機能ユーティリティです。

処理フロー



データセット	環境 変数／ 論理名	記憶媒体	追加情報
エラーデータ	ERRIN	ディスク、テープ（* 注参照）	
エラーメッセージ	stdout/ SYS$OUTPUT		

注意: （*）このシーケンシャルファイルには、名前付きパイプを使用できます（OpenVMS では使用できません。詳細については、『*Adabas Basics*』の「*ユーティリティの使用*」を参照してください）。

シーケンシャルファイル ERRIN には複数エクステントを持つことができます。複数のエクステントを持つシーケンシャルファイルの詳細については、『*Adabas Basics*』の「*ユーティリティの使用*」を参照してください。

チェックポイント

このユーティリティはチェックポイントを書き込みません。

制御パラメータ

次にあげる制御パラメータが使用可能です。

```
D    [NO]DUMP
```

[NO]DUMP

```
[NO]DUMP
```

NODUMP を指定した場合、拒否されたレコードの記述（レコード長、レコードの ISN など）だけが出力され、実際のレコードの内容は出力されません。拒否されたレコードの内容の詳細については、このセクションの「拒否されたデータレコード」セクションを参照してください。

DUMP を指定すると、レコード記述に加えてレコードの内容がダンプされます。ADACMP の場合、圧縮解除されたレコードがダンプされますが、ADADCUの場合は、圧縮されたレコードがダンプされます。

デフォルトは NODUMP です。

例

```
$ adaerr
```

```
%ADAERR-I-STARTED,      11-OCT-2006 18:59:20, Version 6.1.1
%ADAERR-I-RECNOTF, Record NOT found for ISN 317 in file 49
%ADAERR-I-PLOGRB,  from record 1 in block 6 on PLOG 1
%ADAERR-I-IOCNT,        1 IO  on dataset ERRIN
%ADAERR-I-TERMINATED,   11-OCT-2006 18:59:20, elapsed time: 00:00:01
```

拒否されたデータレコード

次のユーティリティによって拒否されたレコードは、可変長フォーマットのエラーファイルに書き込まれます。

- ADACMP
- ADADCU
- ADAINV
- ADAMUP

ADAREC

エラーレコードの構造は、Windows と UNIX の両方のインストールディレクトリのサブディレクトリ 「Adabas/inc」 にヘッダーファイル `iodesam.h` として格納されています。

15 ADAFDU（ファイル定義）

この章では ADAFDU ユーティリティについて説明します。

機能概要

ファイル定義ユーティリティ ADAFDU では、データベースの新しい基本ファイルや LOB ファイルを定義します。Adabas ニュークリアスがアクティブである必要はありません。

外部キーの特殊ディスクリプタ定義および参照整合性定義をはじめとする基本ファイルのフィールド定義は、シーケンシャルファイル FDUFDT から読み取られます。LOB ファイルのフィールド定義は、あらかじめ定義されています。ADAFDU の追加入力はパラメータで指定します。

注意: 新しいファイルに照合ディスクリプタが含まれる場合は、常に ICU バージョン 5.4 で作成されます。

データベースのファイルの論理的な構造を定義するためのデータ定義の構文および使用方法については、『*管理マニュアル*』の「*FDT のレコード構造*」を参照してください。

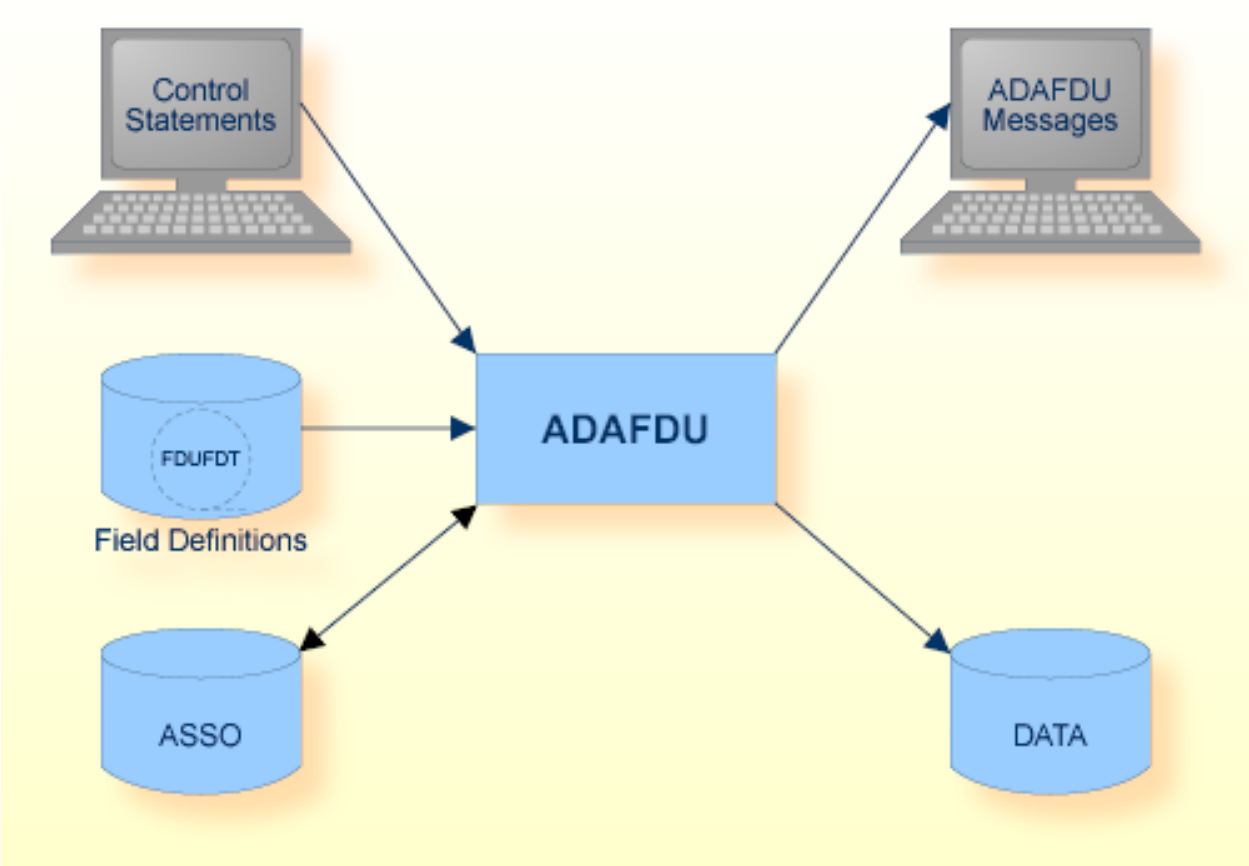
ファイルに必要なアソシエータおよびデータストレージのスペースを計算するための公式については、『*管理マニュアル*』の「*データのロードとアンロード*」の「*ファイルのスペースの見積り*」を参照してください。

データベースでファイルの論理構造を定義するデータ定義の構文と使用方法については、『*管理マニュアル*』の「*FDT のレコード構造*」を参照してください。

ファイルに必要なアソシエータとデータストレージのスペースを計算する公式については、『*管理マニュアル*』の「データのロードとアンロード」の「*ファイルのスペースの見積り*」を参照してください。

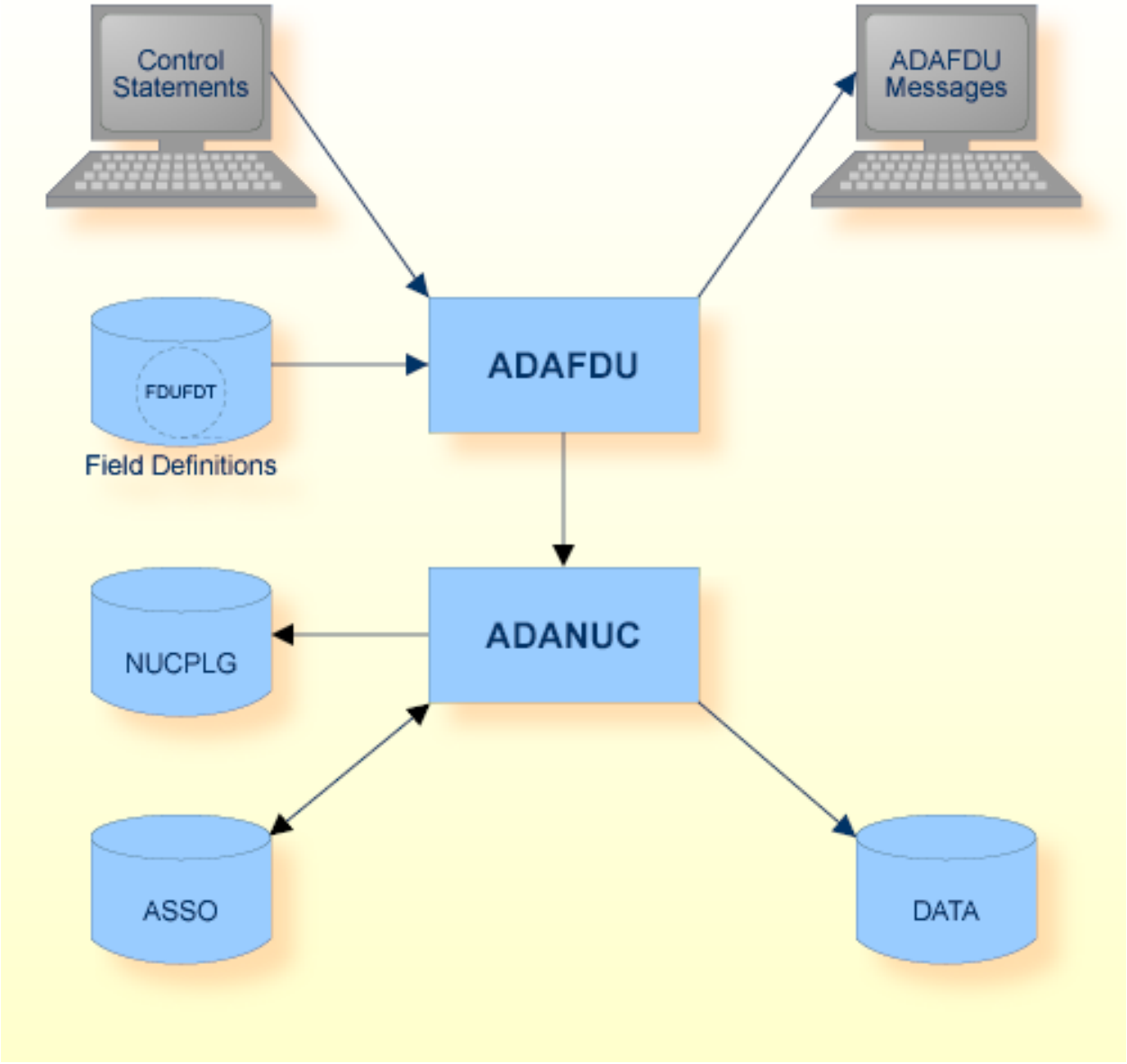
このユーティリティは単一機能ユーティリティです。

処理フロー



Offline Mode

ニュークリアスがアクティブではない場合、ADAFDU が ASSO と DATA に新しいファイルを作成します。



Online Mode

Adabas ニュークリアスがアクティブな場合、ADAFDU は ASSO と DATA に新しいファイルを作成するためにニュークリアスを呼び出します。この場合、チェックポイントは書き込まれませんが、ファイルの作成に関する情報、データベースログに書き込まれます。リカバリの場合は、ファイルが自動的に作成されます。

データセット	環境 変数／ 論理名	記憶 媒体	追加情報
アソシエータ	ASSOx	ディスク	
データストレージ	DATAx	ディスク	
コントロールステートメント	stdin/ SYS$INPUT		ユーティリティマニュアル
ADAFDU メッセージ	stdout/ SYS$OUTPUT		メッセージおよびコード
FDT 情報	FDUFDT	ディスク、テープ（* 注意を参照）	
プロテクションログ	NUCPLG	ディスク	ユーティリティマニュアル ADAPLP

注意: （*）このシーケンシャルファイルには、名前付きパイプを使用できます（OpenVMS にはない。詳細については、『*Adabas Basics*』の「*ユーティリティの使用*」を参照）。

チェックポイント

ユーティリティは、オフラインで実行されると、SYNP チェックポイントを書き込みます。ユーティリティをオンラインで実行すると、ファイル定義が PLOG に書き込まれ、SYNX チェックポイントが書き込まれます。

制御パラメータ

次のコントロールパラメータを使用できます。

```
     ACBLOCKSIZE = numberK

     ACRABN = number

     ADAM_KEY = key

D    ADAM_OVERFLOW = number

D    ADAM_PARAMETER = number

     ADD_LOBFILE = (number,number)

D    ASSOPFAC = number

D    [NO]BT
```

```
D    [NO]CIPHER
D    CONTIGUOUS = ([AC], [,DS] [,NI] [,UI])
D    DATAPFAC = number
M    DBID = number
     DSBLOCKSIZE = numberK
     DSRABN = number
D    DSSIZE = number[B|M]
     FDT
    FILE = number
D    [NO]FORMAT
     LOBFILE = number [,LOBSIZE = number[B|M]]
D    [NO]LOWER_CASE_FIELD_NAMES
D    MAXISN = number
D    NAME{=|:} string
     NIBLOCKSIZE = numberK|(numberK,numberK)
     NIRABN = number|(number,number)
D    NISIZE = number[B|M]|(number[B|M],number[B|M])
     [NO]PGM_REFRESH
D    REUSE = (keyword [,keyword])
    SYFMAX = number
     UIBLOCKSIZE = numberK|(numberK,numberK)
     UIRABN = number|(number,number)
D    UISIZE = number[B|M]|(number[B|M],number[B|M])
```

ACBLOCKSIZE

```
ACBLOCKSIZE = numberK
```

このパラメータで、アドレスコンバータエクステントの割り当てに使用するコンテナのブロックサイズを指定することができます。

例：

```
acblocksize = 6k
```

アドレスコンバータは 6 キロバイトのブロックサイズで割り当てられます。

このブロックサイズのスペースがデータベースに不足している場合、ADAFDUは終了します。

ACRABN

```
ACRABN = number
```

このパラメータは、アドレスコンバータに対するスペース割り当てを開始する RABN を指定するものです。

このパラメータを使用して、指定したコンテナファイルのエクステントにアドレスコンバータを割り当てることができます。

このパラメータを省略すると、ADAFDU が開始 RABN を割り当てます。

ADAM_KEY

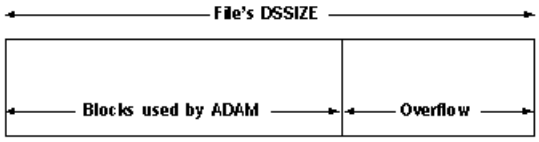
```
ADAM_KEY = key
```

このパラメータを指定すると、ファイルは ADAM ファイルとして定義されます。キーは、ディスクリプタ名とキーワード ISN のどちらでもかまいません。ADAM キーを使用するには、UQ オプションとともに FDT に定義しておく必要があります。キーは、サブディスクリプタ、スーパーディスクリプタ、フォネティックディスクリプタ、ハイパーディスクリプタ以外でなければいけません。マルチプルバリューフィールドやピリオディックグループ内のフィールドに ADAM キーを定義することはできません。また、NU/NC オプションを持つフィールドには指定できません。

ADAM_OVERFLOW

```
ADAM_OVERFLOW = number
```

このパラメータは、ファイルの DATA オーバーフローブロック数を指定するものです。ADAM の計算上のブロック数が多すぎて割り当てできない場合に、オーバーフローブロックが必要になります。オーバーフローブロックはファイルの DATA ブロックの最後から取られます。



オーバーフローブロックは、少なくとも 1 ブロックを割り当てる必要があります。

最大値は（DSSIZE-1）です。

注意: 最大値のチェック時に、DSSIZE がメガバイト単位で指定されている場合は、データストレージのブロックサイズが、実際の値と関係なく 32 であると想定されます。データストレージのブロックサイズが小さければ、より大きい値を ADAM_OVERFLOW に指定することが可能ですが、その場合には、DSSIZE をブロック単位で指定する必要があります。

デフォルトは 1 です。

ADAM_PARAMETER

```
ADAM_PARAMETER = number
```

このパラメータは、ADAM_KEY パラメータにキーワード ISN が指定されている場合の、1 ブロックに格納される連続する ISN の数を指定するものです。

ADAM キーが固定小数点形式のディスクリプタの場合は、パラメータは 1 ブロックに格納される連続する値の数を指定します。その他のキーフォーマットの場合は、値のオフセットを指定します。

ADAM キーが固定小数点形式のディスクリプタの場合は、パラメータは 1 ブロックに格納される連続する値の数を指定します。その他のキーフォーマットの場合は、値のオフセットを指定します。詳細については、『管理マニュアル』を参照してください。

値としては、1 から 10000 まで指定できます。

デフォルト値は 8 です。

ADD_LOBFILE

```
ADD_LOBFILE = (number, number)
```

パラメータ ADD_LOBFILE を使用して LOB ファイルを作成し、最初の番号で指定した既存の基本ファイルに割り当てます。その基本ファイルには、LOB ファイルがまだ割り当てられていない必要があります。第 2 の番号で指定したファイル番号の LOB ファイルが生成され、基本ファイルに割り当てられます。基本ファイルでは LOB 処理が可能になります。指定したファイル番号のファイルは、存在していない必要があります。指定できる最大数値は 32000 です。LOB ファイルのデータストレージ、アドレスコンバータ、ノーマルインデックスとアッパーインデックスを記述するパラメータを指定できますが、次を考慮する必要があります。

- LOB ファイルのデータブロックのブロックサイズは 32 KB にする必要があります。
- LOB の NI ブロックと UI ブロックのブロックサイズは 16 KB より小さくする必要があります。

ADD_LOBFILE と FILE を同時に指定することはできません。

LOB ファイルには要件があらかじめ定義されているので、ADAM_* パラメータなどのように、すべての ADAFDU パラメータが ADD_LOBFILE に関係があるというわけではありません。このようなパラメータは、LOB ファイルが追加されたとき、ADAFDU によって無視されます。

ASSOPFAC

```
ASSOPFAC = number
```

このパラメータは、ファイルのインデックスに使用されるパディングファクタを指定するものです。この指定数値は、後続の一括更新ユーティリティ ADAMUP の実行時に、使用されない各インデックスブロックのパーセンテージになります。このパディングエリアは、Adabas ニュークリアスによって、将来ブロックにエントリを追加する必要が発生した場合に使用できるように予約されています。これにより、オーバーフローエントリを他のブロックに再配置する必要がなくなります。

指定可能な値の範囲は 0～95 です。

ディスクリプタの更新頻度が少ないかまったくない場合には、小さいパディングファクタ（0～10）を指定します。新規ディスクリプタ値が作成されるような大量のディスクリプタ更新がある場合は、大きなパディングファクタ（10～50）を指定します。

パディングファクタは、ユーティリティ ADAORD を使用して後から変更できます。

デフォルトのパディングファクタは 5 です。

[NO]BT

```
[NO]BT
```

NOBT を指定すると、このファイルは BT 以外のファイルになります。つまり、このファイルの変更は通常のトランザクションロジックでは行われず、トランザクションがバックアウトされても、すべての変更はデータベースで維持されます。

BT がデフォルトです。

注意: ニュークリアスがクラッシュした場合は、次の点を考慮してください。

- 前回の ET コマンドの前に発行された、BT 以外のファイルのすべてのデータベース更新は、データベースに適用されます。
- 前回の ET コマンドの後に発行された、BT 以外のファイルのデータベース更新がデータベースに適用されるかどうかははっきりしません。

[NO]CIPHER

```
[NO]CIPHER
```

このオプションは、データレコードの暗号化を有効または無効にするのに使用できます。

暗号化しておけば、Adabas コンテナファイルの内容を権限のないユーザーに見られる心配がなくなります。暗号化が有効であれば、データレコードは Adabas ニュークリアスまたは一括更新ユーティリティ ADAMUP のいずれかを使用して、データベースに格納されるときに暗号化されます。データレコードは、ユーザーとアプリケーションのいずれかから要求があれば復号化されます。したがって、暗号化はどちらからも完全に透過的です。暗号化の詳細については、『*管理マニュアル*』の「*Adabas のセキュリティ機能*」を参照してください。

暗号化しておけば、Adabas コンテナファイルの内容を権限のないユーザーに見られる心配がなくなります。暗号化が有効であれば、データレコードは Adabas ニュークリアスまたは一括更新ユーティリティ ADAMUP のいずれかを使用して、データベースに格納されるときに暗号化されます。データレコードは、ユーザーとアプリケーションのいずれかから要求があれば復号化されます。したがって、暗号化はどちらからも完全に透過的です。暗号化の詳細については、『*管理マニュアル*』の「*Adabas のセキュリティ機能*」を参照してください。

デフォルトは NOCIPHER です。

CONTIGUOUS

```
CONTIGUOUS = ( [AC] [,DS] [,NI] [,UI] )
```

このパラメータは、ADAFDUのスペース割り当ての制御に使用されます。指定すると、指定したタイプの最初の論理エクステントのみが使用されるようになります。

デフォルトでは、ADAFDU はできる限り連続した領域に割り当てようとします。

DATAPFAC

```
DATAPFAC = number
```

このパラメータは、ファイルのデータストレージに使用されるパディングファクタを指定するものです。指定される値は、後で一括更新ユーティリティ ADAMUP または Adabas ニュークリアスを使用してファイルに新規レコードを追加するときに、使用してはならない各データブロックのパーセンテージです。このパディングエリアは、Adabas ニュークリアスによるレコード更新の結果として、ブロック内のレコードが追加スペースを必要とする場合に将来的に使用するために確保されます。これによって、レコードを別のブロックに割り当て直す必要性を回避することができます。

値としては、0 から 95 までが指定可能です。

レコード拡張の頻度が少ないかまったくない場合は、小さいパディングファクタ（0〜10）を指定します。拡張が発生するような大量のレコード更新がある場合は、大きなパディングファクタ（10〜50）を指定します。

パディングファクタは、ユーティリティ ADAORD を使用して後から変更できます。

デフォルトのパディングファクタは 5 です。

DBID

```
DBID = number
```

このパラメータは、使用対象となるデータベースを選択するためのものです。

DSBLOCKSIZE

```
DSBLOCKSIZE = numberK
```

このパラメータで、データストレージエクステントの割り当てに使用するブロックサイズを指定することができます。

例：

```
dsblocksize = 6k
```

データストレージは 6 キロバイトのブロックサイズで割り当てられます。

このブロックサイズのスペースがデータベースに不足している場合、ADAFDUは終了します。

DSRABN

```
DSRABN = number
```

このパラメータは、データストレージのスペース割り当てを開始する RABN を指定するためのものです。

このパラメータは、データストレージを任意のコンテナファイルエクステントに割り当てるのに使用できます。

このパラメータを省略すると、ADAFDU が開始 RABN を割り当てます。

DSSIZE

```
DSSIZE = number [B|M]
```

このパラメータは、データストレージに割り当てるスペースをブロック数またはメガバイト数で指定するものです。デフォルトでは、サイズはメガバイト単位で認識されます。

DSSIZEに指定した値は、該当ファイルのデータストレージに対して割り当てられる論理エクステントのサイズを決定します。

CONTIGUOUS=DSが指定されない限り、できるだけ物理的に連続した領域に割り当てようとする方式が適用されます。

このパラメータは ADAM ファイルの場合には必須です。パラメータ ADAM_OVERFLOW と DSSIZE の間の依存関係については、パラメータ ADAM_OVERFLOW の部分で説明しています。

ADAM 以外のファイルでは、このパラメータを省略できます。この場合、Adabas がデータストレージに使用する合理的なブロック数を計算します。実際に必要なサイズの方が大きい場合、ファイルのサイズが自動的に増やされます。

FDT

```
FDT
```

このパラメータを指定すると、シーケンシャルファイルFDUFDTに含まれたFDTが表示されます。

FILE

```
FILE = number
```

このパラメータは、基本ファイルを作成するときに必要となります。ファイルに割り当てるファイル番号を指定します。

"number"に指定する番号は、現時点でデータベース内の他のファイルに割り当てられておらず、データベースに対して定義可能な最大ファイル番号を超えないものでなければなりません。指定できる最大数値は 32000 です。

ファイル番号の割り当てはどのシーケンスでも可能です。

ADD_LOBFILE と FILE を同時に指定することはできません。

[NO]FORMAT

```
[NO]FORMAT
```

このオプションは、ファイルのインデックスおよびデータストレージに対して割り当てられた RABN をフォーマット化するかどうかの制御に使用されます。ファイルのアドレスコンバータの RABN は、常にフォーマット化されます。

デフォルトは NOFORMAT です。

LOBFILE

```
LOBFILE = number [, LOBSIZE = number[B|M]]
```

LOBFILEを指定すると、指定した番号のLOBファイルが生成され、作成される基本ファイルに割り当てられます。この基本ファイルでは LOB 処理が可能になります。指定したファイル番号の LOB ファイルは、存在していない必要があります。指定できる最大数値は 32000 です。次のことを考慮してください。

- LOB ファイルのデータブロックのブロックサイズは 32 KB にします。
- LOB の NI ブロックと UI ブロックのブロックサイズは 16 KB より小さくします。
- LOBSIZE では、パラメータ DSSIZE と同じように、LOB ファイルのデータストレージのサイズを指定します。
- Adabasは、アドレスコンバータの合理的なサイズ、LOBファイルのノーマルインデックスとアッパーインデックスのサイズを計算します。この値を自分で指定する場合は、LOBFILE を

指定せずに基本ファイルを最初に作成してから、ADAFDUを再び呼び出して、ADD_LOBFILE パラメータで LOB ファイルを追加する必要があります。

[NO]LOWER_CASE_FIELD_NAMES

```
[NO]LOWER_CASE_FIELD_NAMES
```

LOWER_CASE_FIELD_NAMES が指定されている場合、Adabas フィールド名は、大文字に変換されません。NOLOWER_CASE_FIELD_NAMESが指定されている場合、Adabasフィールド名は、大文字に変換されます。デフォルトは、NOLOWER_CASE_FIELD_NAMES です。

MAXISN

```
MAXISN = number
```

このパラメータは、ファイルで見積られる最大の ISN を指定するものです。ファイル定義ユーティリティ ADAFDU は、このパラメータを使用してファイルのアドレスコンバータ（AC）に割り当てるスペース量を決定します。MAXISN のデフォルト値は 5000 です。

CONTIGUOUS=AC が指定されていない限り、できるだけ物理的に連続した領域に割り当てようとする方式が適用されます。

注意: 値は、アドレスコンバータでの MAXISN ISN の保存に必要なアドレスコンバータブロックに収まる ISN 数に切り上げられます。ファイルの MAXISN として使用される正確な値は次の通りです。
(指定された MAXISN / (アドレスコンバータブロックサイズ / 4) + 1) * (アドレスコンバータブロックサイズ / 4) -1例えば、ブロックサイズが 4 KB のアドレスコンバータを使用する場合、デフォルト値 5000 は、(5000 / (4096 / 4) + 1) * (4096 / 4) -1 = 5119 に増えます。

NAME

```
NAME {=|:} string
```

このパラメータは、ファイルに割り当てる名前を指定するものです。この名前は、このファイルに関するデータとともにレポートユーティリティ ADAREP が作成するデータベースステータスレポート内に表示されます。最大 16 文字までが指定できます。等号を指定すると、「文字列」に指定された値が大文字に変換されます。コロンを指定した場合は、大文字への変換は実行されません。

デフォルト値は FILE-n です（n はファイル番号）。

NIBLOCKSIZE

```
NIBLOCKSIZE = numberK|(numberK,numberK)
```

このパラメータで、ノーマルインデックスの割り当てに対するブロックサイズを指定することができます。大きい方のインデックス値のサイズが 253 バイトを超える場合には、ノーマルインデックスに 16 KB 以上のブロックサイズが必要になりますが、ディスクリプタの値が小さい場合には、そのディスクリプタには小さなブロックが割り当てられます。次の点を考慮してください。

- ブロックサイズを1つだけ指定した場合、ファイルはそのブロックサイズのノーマルインデックスブロックだけで作成されます。
- ブロックサイズを 2 つ指定する場合、一方の値は 16K 未満、もう一方の値は 16K 以上にする必要があります。NISIZEにも 2 つの値を指定する必要があります。NIBLOCKSIZE の最初の値は NISIZE の最初の値に対応し、NIBLOCKSIZE の 2 番目の値は NISIZE の 2 番目の値に対応します。

例：

```
niblocksize = 6k
```

ノーマルインデックスは 6 キロバイトのブロックサイズで割り当てられます。

```
niblocksize = (8k,32k)
nisize = (1000b,10m)
```

ノーマルインデックスは、ブロックサイズ 8 KB のブロックが 1000 ブロックとブロックサイズ 32 KBのブロックが 10 MB のブロックで割り当てられます。

このブロックサイズのスペースがデータベースに不足している場合、ADAFDUは終了します。

NIRABN

```
NIRABN = number|(number,number)
```

このパラメータは、ノーマルインデックスのスペース割り当てを開始する RABN を指定するためのものです。このパラメータは、任意のコンテナファイルのエクステントにノーマルインデックスを割り当てるのに使用されます。

2 つの RABN が指定されている場合、一方のブロックサイズは 16 KB 未満、もう一方のブロックサイズは 16 KB 以上にする必要があります。

このパラメータを省略すると、ADAFDU が開始 RABN を割り当てます。

NIBLOCKSIZE と NIRABN の両方が指定される場合、NIRABN として指定された RABN のブロックサイズは NIBLOCKSIZE として指定された値と等しくなければなりません。

NISIZE

```
NISIZE = number [B|M]|(number [B|M],number [B|M])
```

このパラメータは、ノーマルインデックスに対して割り当てられるスペースをブロック数またはメガバイト数で指定するものです。デフォルトでは、サイズはメガバイト単位です。

ブロックサイズを NIBLOCKSIZE または NIRABN パラメータから引き出すことができない場合、NISIZE の最初の値は 16 KB 未満のブロックに使用され、2 番目の値は 16 KB 以上のブロックに使用されます。

CONTIGUOUS=NI が指定されていない限り、できるだけ物理的に連続した領域に割り当てようとする方式が適用されます。

このパラメータを省略すると、Adabas はノーマルインデックスに使用する適切なブロック数を計算します。

例：

```
adafdu: nisize = 100b
```

ブロックサイズを NIBLOCKSIZE または NIRABN パラメータから引き出すことができない場合、ブロックサイズ 16 KB 未満の 100 個のブロックがノーマルインデックスに割り当てられます。

```
adafdu: nisize = (10m,1000b)
```

ブロックサイズを NIBLOCKSIZE または NIRABN パラメータから引き出すことができない場合、ブロックサイズ 16 KB 未満の 10 MB ブロックと、ブロックサイズ 16 KB以上の 1000 個のブロックのノーマルインデックスに割り当てられます。

[NO]PGM_REFRESH

```
[NO]PGM_REFRESH
```

PGM_REFRESHを指定すると、ファイルをロードするときに、E1コマンドによってファイルのリフレッシュが可能です（0 件のレコードをロードした状態にリセット）。

デフォルトは、NOPGM_REFRESH です。

REUSE

```
REUSE = ( keyword [,keyword] )
```

REUSE パラメータはデータストレージスペースまたは ISN の Adabas による再使用を制御します。

REUSE = [NO]DS

NODS を指定する場合、新しく追加されたレコードと、他のブロックに移動する必要があるレコード（更新によてレコードが拡張された結果）は、ファイルに割り当てられたデータストレージエクステントの中にある、最終使用ブロック内に移されます。このブロックに十分なスペースがない場合、次のブロックが使用されます。

DS キーワードを指定する場合、十分なスペースを持つブロックを探すためのデータストレージスペーステーブル（DSST）の検索が Adabas によって行われます。この場合、十分なスペースを持つものとして最初に検出されたブロックが使用されます。

指定ファイルのファイルコントロールブロックは、新規レコードの追加時または更新レコードの移動時に使用される割り当てのタイプを示すように修正されます。

デフォルト値は DS です。

REUSE = [NO]ISN

キーワードの NOISN を指定する場合、削除レコードの ISN は新規レコードに再利用されません。新規レコードにはそれぞれ次に大きい未使用 ISN が割り当てられます。

ISN に REUSE が設定されている場合は、削除されたレコードの ISN を Adabas で再利用できます。ISN 再利用は次のように行われます。新しいレコードがデータベースに保存されると、アドレスコンバータ（AC）ブロックが読み取られ、空き ISN がチェックされます。ISN 再利用のオーバーヘッドを小さく抑えるために、AC ブロックは 1 つだけ読み込まれます。AC ブロックに空いている ISN がない場合は、ISN 再利用がオフの場合と同様に処理されます。

注意: ISN に REUSE を設定しても、必ずしも ISN 再利用が実際に行われるとは限りません。ISN 再利用の失敗は、追加の AC ブロックを読み取るオーバーヘッドにつながるので、再利用できる ISN が見つかる可能性が低い場合、Adabas は ISN 再利用を非アクティブにします。十分な数のレコードを削除した後で、ISN 再利用が再びアクティブになります。

デフォルト値は NOISN です。

例

```
adafdu: reuse = (isn, ds)
```

削除レコードの ISN は、新規レコードに再割り当てされます。データベースにレコードを追加するか、データベース内で更新レコードを移動するときに、DSST に空きスペースがあるかどうかチェックされます。

```
adafdu: reuse = isn
```

データストレージおよび ISN の再利用が可能です。

```
adafdu: reuse = <cr>
```

データストレージを再使用し、ISN を再使用しないという指定です。これはデフォルト設定です。

SYFMAX

```
SYFMAX = number
```

このパラメータは、システム生成マルチプルバリューフィールドに対して、生成される値の最大数を指定します。明示的な最大値はありませんが、値が高く定義されると、レコードオーバーフローが発生することに注意してください。圧縮データレコードは 1 つの DATA ブロック内に収まる必要もあり、SYFMAX 値がシステム生成マルチプルバリューフィールドに対して定義されています。

デフォルト値は 1 です。

UIBLOCKSIZE

```
UIBLOCKSIZE = numberK|(numberK,numberK)
```

このパラメータで、アッパーインデックスの割り当てに対するブロックサイズを指定することができます。大きい方のインデックス値のサイズが 253 バイトを超える場合には、アッパーインデックスに 16 KB 以上のブロックサイズが必要になりますが、ディスクリプタの値が小さい場合には、そのディスクリプタには小さなブロックが割り当てられます。次の点を考慮してください。

- ブロックサイズを1つだけ指定した場合、ファイルはそのブロックサイズのノーマルインデックスブロックだけで作成されます。
- ブロックサイズを 2 つ指定する場合、一方の値は 16K 未満、もう一方の値は 16K 以上にする必要があります。UISIZE にも 2 つの値を指定する必要があります。UIBLOCKSIZE の最初の値は UISIZE の最初の値に対応し、UIBLOCKSIZE の 2 番目の値は UISIZE の 2 番目の値に対応します。

例：

```
uiblocksize = 6k
```

アッパーインデックスは 6 キロバイトのブロックサイズで割り当てられます。

```
uiblocksize = (8k,32k)
uisize = (1000b,10m)
```

アッパーインデックスは、ブロックサイズ 8 KB のブロックが 1000 ブロックとブロックサイズ 32 KBのブロックが 10 MB のブロックで割り当てられます。

このブロックサイズのスペースがデータベースに不足している場合、ADAFDUは終了します。

UIRABN

```
UIRABN = number|(number,number)
```

このパラメータは、アッパーインデックスのスペース割り当てを開始する RABN を指定するためのものです。このパラメータは、コンテナファイルエクステントにアッパーインデックスを割り当てるのに使用されます。

2 つの RABN が指定されている場合、一方のブロックサイズは 16 KB 未満、もう一方のブロックサイズは 16 KB 以上にする必要があります。

UIBLOCKSIZE と UIRABN の両方が指定される場合、UIRABN として指定された RABN のブロックサイズは UIBLOCKSIZE として指定された値と等しくなければなりません。

このパラメータを省略すると、ADAFDU が開始 RABN を割り当てます。

UISIZE

```
UISIZE = number [B | M]
```

このパラメータは、アッパーインデックスに対して割り当てられるスペースをブロック数またはメガバイト数で指定するものです。デフォルトでは、サイズはメガバイト単位です。

ブロックサイズを UIBLOCKSIZE または UIRABN パラメータから引き出すことができない場合、UISIZE の最初の値は 16 KB 未満のブロックに使用され、2 番目の値は 16 KB 以上のブロックに使用されます。

CONTIGUOUS=UI が指定されない限り、できるだけ物理的に連続した領域に割り当てようとする方式が適用されます。

このパラメータを省略すると、Adabas はアッパーインデックスに使用する適切なブロック数を計算します。

例

例：

```
adafdu: dbid = 1, file = 6, maxisn = 20000, dssize = 100B,
adafdu: assopfac = 10, datapfac = 10,
adafdu: uisize = 20b, nisize = 5
```

ファイル6をロードします。ファイルに対して最大見積りレコード数20000がプリセットされます。データストレージに対しては、100ブロックが割り当てられます。アソシエータおよびデータストレージのパディングファクタは両方とも 10 %です。アッパーインデックスには、20 ブロックが割り当てられ、ノーマルインデックスには、5メガバイトが割り当てられます。ノーマルインデックスの ISN サイズには、暗黙的に 2 が設定されます。

例：

```
adafdu: dbid = 1, file = 7, maxisn = 350000,
adafdu: assopfac = 5, datapfac = 15,
adafdu: dssize = 100,
adafdu: uisize = 2, nisize = 30
```

ファイル 7 をロードします。ファイルに対しては最大見積りレコード数 350000 がプリセットされます。アソシエータパディングファクタは5 %です。データストレージパディングファクタは 15 %です。データストレージに対しては、100 メガバイトが割り当てられます。ノーマルインデックスの ISN サイズには、暗黙的に 4 が設定されます。

例：

```
adafdu: dbid = 1, file = 8,
adafdu: maxisn = 10000, dssize = 20,
adafdu: uisize = 10b, nisize = 50b
```

ファイル8をロードします。ファイルに対しては最大見積りレコード数10000がプリセットされます。データストレージに対しては、20 メガバイトが割り当てられます。アソシエータとデータストレージのパディングファクタは両方とも 5 %（デフォルト）です。

例：

```
adafdu: dbid = 1, file = 9, maxisn = 55000, dssize = 2000b, dsrabn = 30629,
adafdu: uisize = 50b, nisize = 300b,
adafdu: assopfac = 20, datapfac = 10
```

ファイル9をロードします。ファイルに対しては最大見積りレコード数55000がプリセットされます。データストレージに対しては、2000 ブロックが割り当てられます。データストレージの割り当ては RABN 30629 から開始されます。アッパーインデックスに対しては、50 ブロックが割り当てられます。ノーマルインデックスに対しては、300ブロックが割り当てられます。アソシエータのパディングファクタは20％です。データストレージのパディングファクタは10％です。

例：

```
adafdu: dbid = 1, file = 10, maxisn = 20000
```

ファイル 10 をロードします。ファイルに対して最大見積りレコード数 20000 がプリセットされます。すべてのスペース割り当ては Adabas によって計算されます。

16 ADAFIN（ファイル情報レポート）

この章では ADAFIN ユーティリティについて説明します。

機能概要

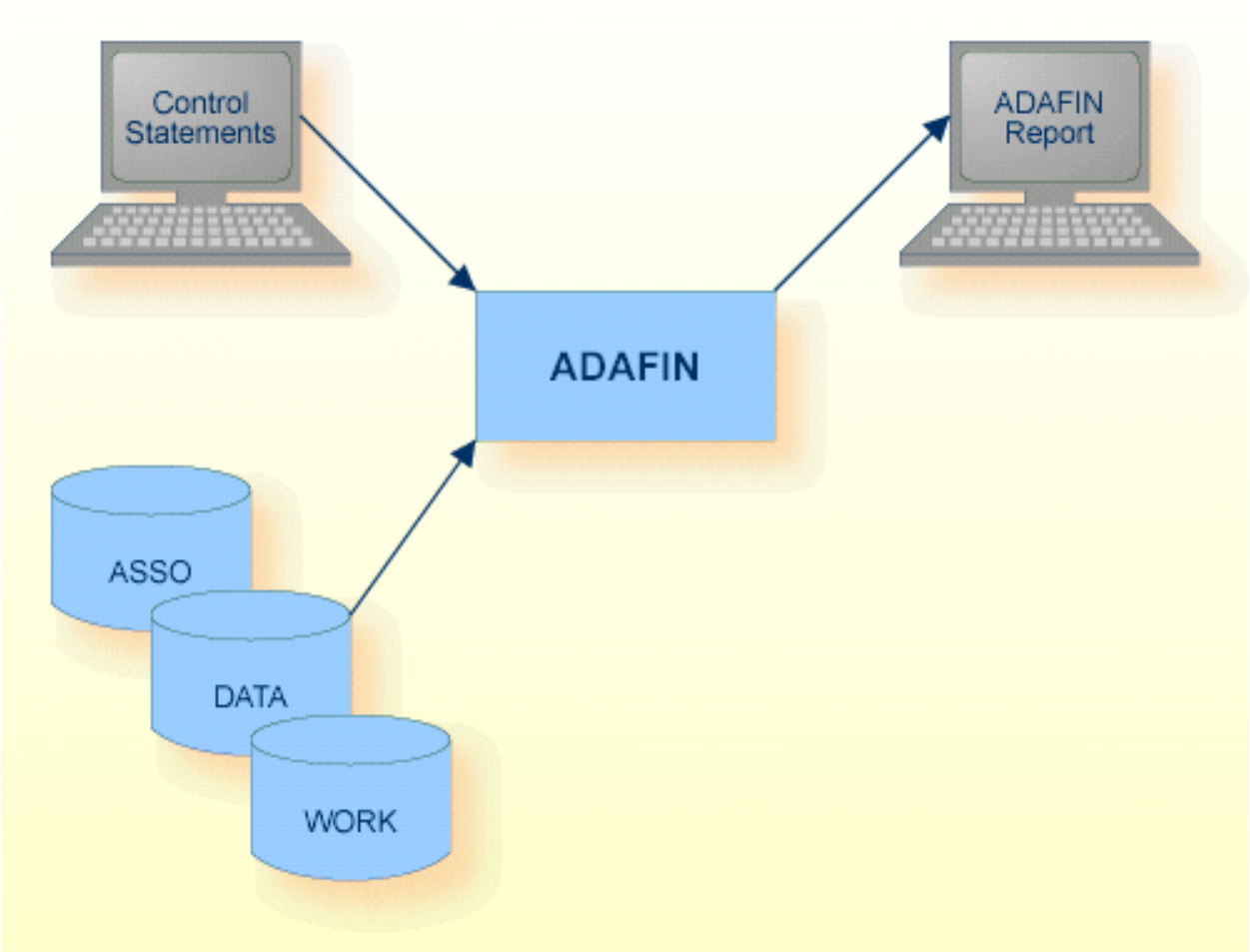
ファイル情報ユーティリティ ADAFIN を使用すると、選択した 1 つ以上のファイルについて、次の情報を表示できます。

- FDT
- ディスクリプタ情報
- データストレージ、ノーマルインデックス、またはアッパーインデックスのブロック数とその用途

（選択された 1 つ以上のファイル）。

このユーティリティは多機能ユーティリティです。

処理フロー



データセット	環境変数／論理名	記憶媒体	追加情報
アソシエータ	ASSOx	ディスク	
データストレージ	DATAx	ディスク	
コントロールステートメント	stdin/ SYS$INPUT		ユーティリティマニュアル
ADAFIN メッセージ	stdout/ SYS$OUTPUT		メッセージおよびコード
WORK	WORK1	ディスク	

チェックポイント

このユーティリティはチェックポイントを書き込みません。

制御パラメータ

次のコントロールパラメータを使用できます。

```
     ADAM_DS = keyword

M    DBID = number

     DESCRIPTOR = { = | : }{ * | (string [,string]...)  }

     FDT

M    FILE = { * | (number [-number] [,number [-number]]...)  }

D    [NO]HISTOGRAM

     USAGE = (keyword [,keyword [,keyword]])
```

ADAM_DS

```
ADAM_DS = keyword
```

このパラメータは、ADAMファイルのUSAGE=DSと組み合わせて使用できます。これにより、情報を表示するためのADAMファイルのデータセクションを選択します。次のキーワードを使用できます。

キーワード	説明
FULL	DS スペースすべてを選びます。
ADAM	ADAM エリアだけを選びます。
OVERFLOW	ADAM オーバーフローエリアだけを選びます。

DBID

```
DBID = number
```

このパラメータは、使用対象となるデータベースを選択するためのものです。

DESCRIPTOR

```
DESCRIPTOR = { = | : }{ * | (string [,string]...) }
```

この機能は、情報を表示するためのディスクリプタのリストを定義するものです。複数のファイルを選択する場合、全ディスクリプタに対する場合（DESCRIPTOR=*）にのみ情報は要求できます。

DESCRIPTOR 機能は、ニュークリアスが稼動している状態で、選択ファイルが更新用にオープンされていない場合にだけ実行できます。この機能は、FILE パラメータとともに選択する必要があります。

DESCRIPTOR 機能は並列更新（例えば ADAINV REINVERT）に対して同期を取りません。

例

```
adafin: file=13, descriptor=ca

Database 76, File    13  (MISCELLANEOUS   )          27-OCT-2006 08:08:17


Descriptor CA , Format: A , Options: NU

                              min         max          ave
-------------------------------------------------------------
Length                          1         233        20.59
ISNs per value                  1           2         1.08

Values:       different:          86     total:          93
ASSO-Blocks:          NI:           2        UI:           1
```

```
adafin: file=(11,12), descriptor=*
Database 1, File    11  (EMPLOYEES-NAT   )          27-OCT-2006 08:09:39


Descriptor AA , Format: A , Options: UQ

                                   min              max                ave
-------------------------------------------------------------------------
Length                               8                8               8.00
ISNs per value                       1                1               1.00

Values:       different:               1,107     total:              1,107
```

```
ASSO-Blocks:         NI:                  5        UI:                  1


Descriptor AE , Format: A , Options: None

                                  min              max                ave
-------------------------------------------------------------------------
Length                              3               17               6.78
ISNs per value                      1               19               1.37

Values:       different:                804     total:             1,107
ASSO-Blocks:         NI:                  4        UI:                  1


Descriptor AH , Format: P , Options: NC

                                  min              max                ave
-------------------------------------------------------------------------
Length                              4                4               4.00
ISNs per value                      1               43               1.20

Values:       different:                921     total:             1,107
ASSO-Blocks:         NI:                  4        UI:                  1


Descriptor AJ , Format: A , Options: NU

                                  min              max                ave
-------------------------------------------------------------------------
Length                              3               20               8.52
ISNs per value                      1              141               3.60

Values:       different:                307     total:             1,107
ASSO-Blocks:         NI:                  3        UI:                  1


Descriptor AO , Format: A , Options: None

                                  min              max                ave
-------------------------------------------------------------------------
Length                              6                6               6.00
ISNs per value                      1               99               6.62

Values:       different:                167     total:             1,107
ASSO-Blocks:         NI:                  2        UI:                  1


Descriptor AP , Format: A , Options: NU

                                  min              max                ave
-------------------------------------------------------------------------
Length                              2               25              12.56
```

```
ISNs per value                      1              75               4.67

Values:       different:              237     total:             1,107
ASSO-Blocks:         NI:                3        UI:                 1


Descriptor AZ , Format: A , Options: NU,MU

                                   min             max               ave
-------------------------------------------------------------------------
Length                              3               3              3.00
ISNs per value                      1             843             86.28

Values:       different:               21     total:             1,812
ASSO-Blocks:         NI:                2        UI:                 1


Super-Descriptor H1 , Format: B , Options: NU
Parent field(s):            AU  (  1 -   2) U
                            AV  (  1 -   2) U

                                   min             max               ave
-------------------------------------------------------------------------
Length                              4               4              4.00
ISNs per value                      1              93              4.17

Values:       different:              259     total:             1,081
ASSO-Blocks:         NI:                2        UI:                 1


Phonetic-Descriptor PH , Format: A , Options: None
Parent field(s):            AE              A

                                   min             max               ave
-------------------------------------------------------------------------
Length                              3               3              3.00
ISNs per value                      1              33              1.82

Values:       different:              608     total:             1,107
ASSO-Blocks:         NI:                3        UI:                 1


Sub-Descriptor S1 , Format: A , Options: None
Parent field(s):            AO  (  1 -   4) A

                                   min             max               ave
-------------------------------------------------------------------------
Length                              4               4              4.00
ISNs per value                      1             208             85.15

Values:       different:               13     total:             1,107
ASSO-Blocks:         NI:                2        UI:                 1
```

```
Super-Descriptor S2 , Format: A , Options: None
Parent field(s):            AO  (  1 -   6) A
                            AE  (  1 -  20) A

                                    min              max                ave
-------------------------------------------------------------------------------
Length                               9               23              12.78
ISNs per value                       1                5               1.05

Values:       different:             1,052     total:               1,107
ASSO-Blocks:         NI:                 6        UI:                   1


Super-Descriptor S3 , Format: A , Options: NU,PE
Parent field(s):            AR  (  1 -   3) A
                            AS  (  1 -   9) P

                                    min              max                ave
-------------------------------------------------------------------------------
Length                              12               12              12.00
ISNs per value                       1               25               2.15

Values:       different:             1,567     total:               3,383
ASSO-Blocks:         NI:                10        UI:                   1

Highest PE-occurrence:          5



Database 1, File    12  (VEHICLES        )          10-OCT-2006 14:30:39


Descriptor AA , Format: A , Options: UQ,NU

                                    min              max                ave
-------------------------------------------------------------------------------
Length                               6               10               7.91
ISNs per value                       1                1               1.00

Values:       different:               772     total:                 772
ASSO-Blocks:         NI:                 4        UI:                   1


Descriptor AC , Format: A , Options: None

                                    min              max                ave
-------------------------------------------------------------------------------
Length                               1                8               7.74
ISNs per value                       1               24               1.16
```

```
Values:       different:                662     total:                 773
ASSO-Blocks:         NI:                  3        UI:                   1


Descriptor AD , Format: A , Options: NU

                                  min              max                ave
-------------------------------------------------------------------------
Length                              2               14               6.63
ISNs per value                      1              179              17.17

Values:       different:                45     total:                 773
ASSO-Blocks:         NI:                  1        UI:                   1


Descriptor AF , Format: A , Options: NU

                                  min              max                ave
-------------------------------------------------------------------------
Length                              3               10               4.95
ISNs per value                      1              135              11.36

Values:       different:                68     total:                 773
ASSO-Blocks:         NI:                  1        UI:                   1


Descriptor AH , Format: A , Options: FI

                                  min              max                ave
-------------------------------------------------------------------------
Length                              1                1               1.00
ISNs per value                    169              329             257.66

Values:       different:                 3     total:                 773
ASSO-Blocks:         NI:                  1        UI:                   1


Super-Descriptor AO , Format: A , Options: NU
Parent field(s):           AG  (  1 -   2) U
                           AD  (  1 -  20) A

                                  min              max                ave
-------------------------------------------------------------------------
Length                              4               16               8.63
ISNs per value                      1               45               4.29

Values:       different:               180     total:                 773
ASSO-Blocks:         NI:                  2        UI:                   1

Total of 18 descriptors
```

指定されたファイルのすべてのディスクリプタについての情報が表示されます。

FDT

```
FDT
```

このパラメータは、FILE パラメータで選択されたファイルのフィールド定義テーブル（FDT）を表示するためのものです。この機能は、FILE パラメータとともに選択する必要があります。

例

```
adafin: file=9, fdt
Database 1, File     9  (EMPLOYEES       )          27-OCT-2006 08:11:42


Field Definition Table:

   Level  I Name I Length I Format I   Options          I Flags   I Encoding
-------------------------------------------------------------------------------
  1       I  AA  I    8   I    A   I DE,UQ             I         I
  1       I  AB  I        I        I                   I         I
   2      I  AC  I   20   I    W   I NU                I         I
   2      I  AE  I   20   I    W   I NU                I SP      I
   2      I  AD  I   20   I    W   I NU                I         I
  1       I  AF  I    1   I    A   I FI                I         I
  1       I  AG  I    1   I    A   I FI                I         I
  1       I  AH  I    8   I    U   I DE                I         I
  1       I  A1  I        I        I                   I         I
   2      I  AI  I   20   I    W   I NU,MU             I         I
   2      I  AJ  I   20   I    W   I DE,NU             I         I
   2      I  AK  I   10   I    A   I NU                I         I
   2      I  AL  I    3   I    A   I NU                I         I
  1       I  A2  I        I        I                   I         I
   2      I  AN  I    6   I    A   I NU                I         I
   2      I  AM  I   15   I    A   I NU                I         I
  1       I  AO  I    6   I    A   I DE                I SB,SP   I
  1       I  AP  I   25   I    W   I DE,NU             I         I
  1       I  AQ  I        I        I PE                I         I
   2      I  AR  I    3   I    A   I NU                I SP      I
   2      I  AS  I    5   I    P   I NU                I SP      I
   2      I  AT  I    5   I    P   I NU,MU             I         I
  1       I  A3  I        I        I                   I         I
   2      I  AU  I    2   I    U   I                   I SP      I
   2      I  AV  I    2   I    U   I NU                I SP      I
  1       I  AW  I        I        I PE                I         I
   2      I  AX  I    8   I    U   I NU                I         I
   2      I  AY  I    8   I    U   I NU                I         I
  1       I  AZ  I    3   I    A   I DE,NU,MU          I         I
-------------------------------------------------------------------------------
   Type   I Name I Length I Format I   Options          I Parent field(s)   Fmt
-------------------------------------------------------------------------------
 COLL     I  CN  I1,144   I        I NU,HE             I AE  de__PHONEBOOK
          I      I        I        I                   I     PRIMARY
```

```
--------------------------------------------------------------------------------
 SUPER    I  H1  I    4   I    B   I NU               I AU (   1 -   2 )   U
          I      I        I        I                  I AV (   1 -   2 )   U
--------------------------------------------------------------------------------
 SUB      I  S1  I    4   I    A   I                  I AO (   1 -   4 )   A
--------------------------------------------------------------------------------
 SUPER    I  S2  I   26   I    A   I NU               I AO (   1 -   6 )   A
          I      I        I        I                  I AE (   1 -  20 )   W
--------------------------------------------------------------------------------
 SUPER    I  S3  I   12   I    A   I NU,PE            I AR (   1 -   3 )   A
          I      I        I        I                  I AS (   1 -   9 )   P
--------------------------------------------------------------------------------
```

FILE

```
FILE = { * | (number [-number] [,number [-number]]...) }
```

このパラメータは、1つのデータベースから1つ以上のファイルを選択し、後続のパラメータに基づいてそれらのファイルの情報を表示させるものです。FILE=* を指定すると、すべてのファイルが選択されます。

[NO]HISTOGRAM

```
[NO]HISTOGRAM
```

HISTOGRAMオプションを選択すると、DESCRIPTOR機能によって続けて表示されるすべての情報に、ディスクリプタ値の長さの配分が全体的にどのようになっているかが図示されます。

HISTOGRAMを使用する場合には、DESCRIPTORパラメータの前に指定する必要があります。

HISTOGRAM オプションを使用しても、データセットに対する I/O は発生しません。

デフォルトは NOHISTOGRAM です。

例（HISTOGRAM を指定するとき）

```
adafin: file=9, histogram, descriptor=ap
Database 1, File     9  (EMPLOYEES       )          27-OCT-2006 08:12:44


Descriptor AP , Format: W , Options: NU

                                   min              max                ave
--------------------------------------------------------------------------
Length                               2               26              12.71
ISNs per value                       1               75               4.61

Values:       different:                240     total:              1,107
ASSO-Blocks:         NI:                  3        UI:                  1
```

```
Histogram of descriptor value length for descriptor AP


Length |              25%             50%             75%           100%| Frequency
-------+--------------+--------------+--------------+---------------+----------
   2   |*                                                               |         1
   3   |*                                                               |        22
   5   |*                                                               |         7
   6   |*                                                               |        26
   7   |******                                                          |       124
   8   |****                                                            |        83
   9   |******                                                          |       117
  10   |******                                                          |       119
  11   |***                                                             |        67
  12   |****                                                            |        83
  13   |*                                                               |        23
  14   |**                                                              |        47
  15   |**                                                              |        46
  16   |**                                                              |        46
  17   |*                                                               |        29
  18   |*****                                                           |       101
  19   |*                                                               |        27
  20   |*                                                               |        29
  21   |*                                                               |        33
  22   |*                                                               |        17
  23   |*                                                               |        20
  24   |*                                                               |        21
  25   |*                                                               |         5
  26   |*                                                               |        14
-------------------------------------------------------------------------------


adafin:
```

表示される情報は、次のように意味を持っています。

キーワード	説明
Length	この列に表示されるそれぞれの値 n は、n バイトの長さのディスクリプタ値がファイルに存在することを示しています。 この列の値の範囲は、ヒストグラムの前の表に示されている最小値（"min" 列）と最大値（"max" 列）の間になります。
Frequency	この列に表示される値は、それぞれの長さに該当するディスクリプタの値の数を示します。 この列の値の合計は、対象のディスクリプタの値の総数に等しくなります。

全ディスクリプタ値が同じ長さの値を持つ場合、例えば次の例のように、その概要レポートは通常と異なるものとなります。

```
adafin: file=9, histogram, descriptor=aa
Database 1, File     9  (EMPLOYEES       )          27-OCT-2006 08:15:16


Descriptor AA , Format: A , Options: UQ

                                  min              max                ave
-------------------------------------------------------------------------
Length                              8                8               8.00
ISNs per value                      1                1               1.00

Values:       different:              1,107     total:              1,107
ASSO-Blocks:          NI:                  5        UI:                  1



Histogram of descriptor value length for descriptor AA


Length |             25%            50%            75%          100%| Frequency
-------+-------------+-------------+-------------+---------------+----------
   8   |************************************************************|     1,107
-------------------------------------------------------------------------------
```

このヒストグラムは、8バイトの長さを持つディスクリプタ値だけしか該当ファイル内に含まれていないことを示します。そのファイルには、該当ディスクリプタ AA に対して合計で 1107 個の値が含まれていることがわかります。

例（NOHISTOGRAM を指定するとき）

```
adafin: file=9, histogram, descriptor=ap
Database 1, File     9  (EMPLOYEES       )          27-OCT-2006 08:14:24


Descriptor AP , Format: W , Options: NU

                                  min              max                ave
-------------------------------------------------------------------------
Length                              2               26              12.71
ISNs per value                      1               75               4.61

Values:       different:                240     total:              1,107
ASSO-Blocks:          NI:                  3        UI:                  1
```

USAGE

```
USAGE = (keyword [,keyword [,keyword]])
```

指定されたキーワードに応じて、このパラメータはそのファイル内での使用ブロックのパーセンテージを表示します。

キーワード	説明
DS	データストレージ内のブロック使用状況
NI	ノーマルインデックス内のブロック使用状況
UI	メイン／アッパーインデックス内のブロック使用状況

例

```
adafin: file=13, usage=ds

Database 76, File    13  (MISCELLANEOUS   )          27-OCT-2006 08:16:18


DS - Blocks allocated =         50 , used =         49 , unused =          1

Records:  Number      =        179
          Length: max =      1,991 , min  =        260 , avg    =     997.47

  0%:                                                                0 blocks
  5%:                                                                0 blocks
 10%:                                                                0 blocks
 15%:                                                                0 blocks
 20%:                                                                0 blocks
 25%:                                                                0 blocks
 30%:                                                                0 blocks
 35%:                                                                0 blocks
 40%:                                                                0 blocks
 45%:                                                                0 blocks
 50%:                                                                0 blocks
 55%:                                                                0 blocks
 60%:                                                                0 blocks
 65%:                                                                0 blocks
 70%:****                                                           2 blocks
 75%:                                                                0 blocks
 80%:**                                                             1 block
 85%:*************                                                  6 blocks
 90%:*****************************                                 13 blocks
 95%:************************************************************** 27 blocks
100%:                                                               0 blocks
```

データベース76内のファイル13の使用データブロックについての情報が表示されます。データストレージでは、割り当てられた50ブロックのうち、49ブロックが使用済みであり、1ブロッ

クが未使用です。レコード数の合計は179であり、レコード長の範囲は260～1991です。平均のレコード長は、997.47です。後続の各行には、それぞれのパーセント範囲に応じた使用ブロック数が表示され、全体的な使用状況を把握できるようになっています。ブロック（27）の大多数は90~95 %まで使用されていることがわかります。

例（ADAM ファイルの場合）

```
adafin: file = 8
adafin: adam_ds = full
adafin: usage = ds

Database  30, File   8 (ADAM_FILE       )      11-OCT-2006 12:08:57


ADAM key = FF     ADAM parameter =     5               ADAM_DS = FULL

DS - Blocks used for ADAM =          94
Total overflow blocks     =           1,  used = 1

Records:  Number      =    3863
          In ADAM area=    3860 , ovfl =        3
          Length: max =       9 , min  =        9 , avg    =      9.00

  0%: ********                                                 10 blocks
  5%: ***                                                       4 blocks
 10%:                                                           0 blocks
 15%:                                                           0 blocks
 20%:                                                           0 blocks
 25%:                                                           0 blocks
 30%: *                                                         1 block
 35%:                                                           0 blocks
 40%:                                                           0 blocks
 45%:                                                           0 blocks
 50%: *                                                         1 block
 55%:                                                           0 blocks
 60%:                                                           0 blocks
 65%: ************************************************     74 blocks
 70%:                                                           0 blocks
 75%:                                                           0 blocks
 80%:                                                           0 blocks
 85%:                                                           0 blocks
 90%:                                                           0 blocks
 95%: ***                                                       3 blocks
100%: *                                                         2 blocks
```

ファイル8のデータブロックすべてについての情報、つまりADAMファイルが表示されます。ADAMパラメータは5に設定されています。94ブロックがADAMエリアに使用され、1ブロックがオーバーフローのために確保されています。ADAMエリアのレコード数は3860、オーバーフローエリアのレコード数は3です。

17 ADAFRM（新規データベースのフォーマットおよび作成）

この章では ADAFRM ユーティリティについて説明します。

機能概要

ユーティリティ ADAFRM はデータベースに割り当てるコンテナファイル（ASSO、DATA、WORK）を作成し、データベースシステムファイルを含むデータベースを設定します。さらに、TEMP および SORT ファイルをフォーマットするために、使用することもできます。

データベースは、ADABAS.INI ファイルに含まれます。

ファイル DBnnn.INI がまだ存在しない場合、ADAFRM は、ADABAS.INI から派生したデフォルトパラメータを含む DBnnn.INI ファイルを作成し、それを適切なデータベースディレクトリに保存します（DBnnn.INI ファイルの詳細については、『*Adabas 拡張オペレーション*』を参照してください）。

次のルールは、作成するコンテナエクステントの場所を決定する際に適用されます。

1. コンテナエクステントの環境変数が存在する場合、その環境変数を使用します。
2. ADAFRM の開始前に DBnnn.INI ファイルがすでに存在し、コンテナエクステントのエントリが含まれている場合は、DBnnn.INI ファイルのエントリが使用されます。
3. そうでない場合は、データベースディレクトリ（UNIX では $ADADATADIR/dbnnn、Windows では %ADADATADIR%\dbnnn）内に *xxxxx.nnn* という名前のコンテナエクステントを作成します。ここで、*xxxx* はコンテナタイプ、*x* はコンテナエクステント、および *nnn* はデータベース番号です。

同時にデータベースを作成せずに SORT コンテナと TEMP コンテナを作成した場合は例外となります。ここでは、次のルールが適用されます。

1. コンテナエクステントの環境変数が存在する場合、その環境変数を使用します。
2. それ以外の場合、*xxxxx* という名前で、現在のディレクトリ内にコンテナを作成します。ここで、*xxxx* は、コンテナタイプ、*x* はコンテナエクステントです。

SORT コンテナまたは TEMP コンテナだけではなく、データベースも作成した場合、作成されたコンテナエクステントは DBnnn.INI ファイルに保存されます。ADAFRM の開始前に DBnnn.INI ファイルがすでに作成されている場合、ファイル内の他のすべての値は変更されません。ファイルが存在しない場合は、デフォルト値を使用して作成されます。

データベースを作成せずに SORT コンテナまたは TEMP コンテナのみを作成する場合は、DBnnn.INI ファイルで更新は実行されません。

RAW デバイスインターフェイスの場合、ブロックの配置を意識することが必要であれば、ユーティリティ ADADEV が適しています。ファイルシステムの RAW デバイスおよびファイルは、データベースコンテナファイルに使用することができます。

ADAFRM によって 255 までの ASSO および 255 までの DATA コンテナファイルを作成することができます。ADADBM の ADD_CONTAINER 機能を使用することにより、さらに多くのコンテナを追加することができます。

ADAFRM がコンテナファイルを作成した後、グローバルな Adabas ブロックを初期化し、3 つの Adabas システムファイル（チェックポイントファイル、ET データファイル、セキュリティファイル）を挿入し、それらに対するスペースを割り当てます。チェックポイントファイルには 3000 レコード、ET データファイルには 3000 レコード、セキュリティファイルには 200 レコードが割り当てられます。

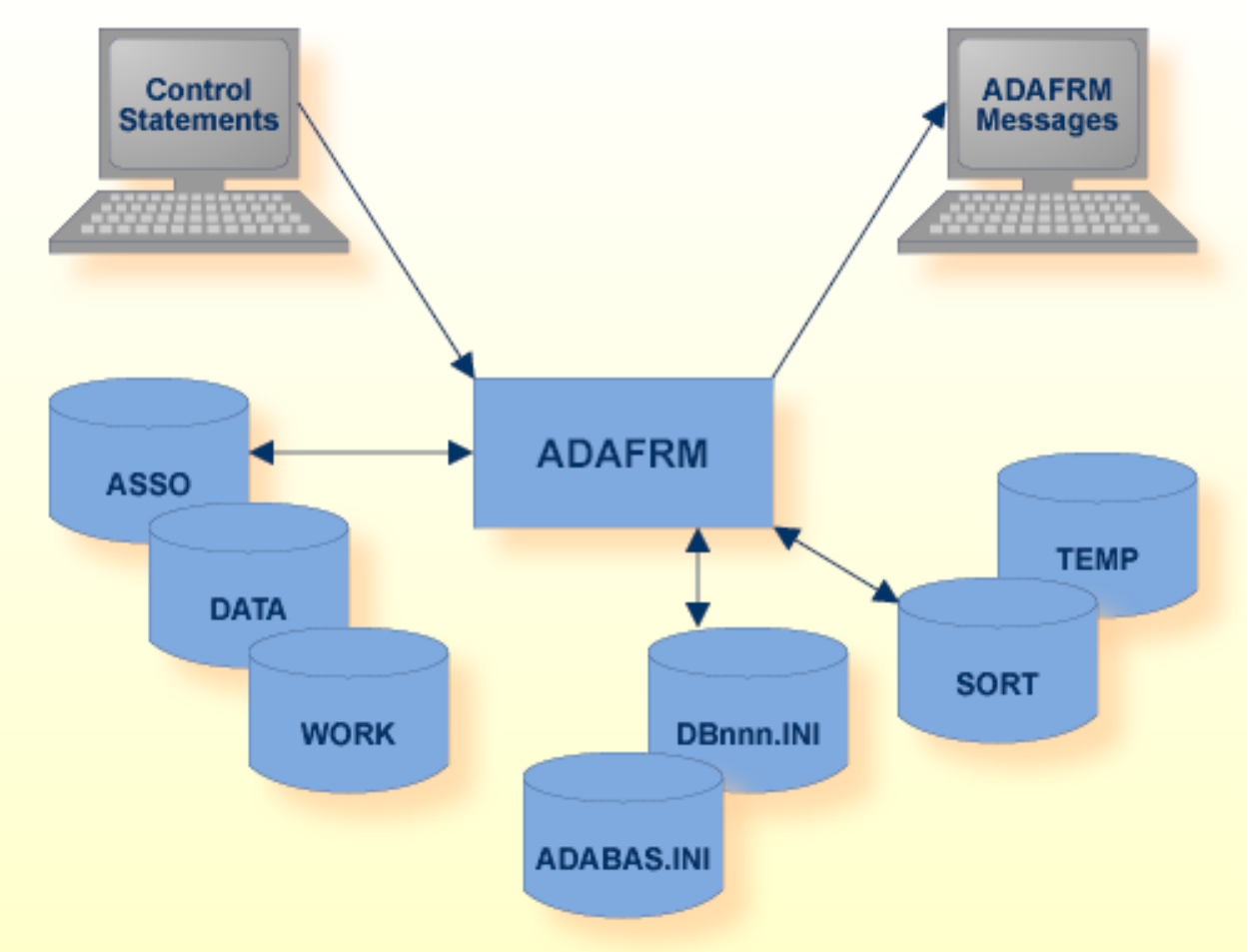
1 キロバイトから 32 キロバイトまでのブロックサイズをデータベースコンテナファイルに使用することができます。

コンテナファイルを再フォーマットようとすると、ユーティリティはエラーメッセージを発行して終了します。これは、データベースが誤って上書きされるのを防止するためです。

注意: OpenVMS では、できるだけ物理的に連続した領域にコンテナファイルを割り当てる方式が適用されます。この理由から、パフォーマンスの低下を避けるために、ディスクスペースをできる限り最適化する必要があります。

このユーティリティは、単一機能ユーティリティです。

処理フロー



データベースをフォーマットする場合は、次のような設定項目になります。

データセット	環境変数／論理名	記憶媒体	追加情報
ADABAS.INI		ディスク	Adabas 拡張オペレーションマニュアル
アソシエータ	ASSOx	ディスク	
データストレージ	DATAx	ディスク	
DBnnn.INI		ディスク	Adabas 拡張オペレーションマニュアル
コントロールステートメント	stdin/ SYS$INPUT		ユーティリティマニュアル

データセット	環境変数／論理名	記憶媒体	追加情報	
ADAFRM メッセージ	stdout/ SYS$OUTPUT		メッセージおよびコード	
WORK	WORK1	ディスク		

TEMP または SORT をフォーマットする場合は、次のような設定項目になります。

データセット	環境変数／論理名	記憶媒体	追加情報
ソートストレージ	SORTx	ディスク	
コントロールステートメント	stdin/ SYS$INPUT		ユーティリティマニュアル
ADAFRM メッセージ	stdout/ SYS$OUTPUT		メッセージおよびコード
一時ストレージ	TEMPx	ディスク	

チェックポイント

このユーティリティはチェックポイントを書き込みません。

制御パラメータ

新規データベース構築時、次にあげる制御パラメータが使用可能です。

```
D    ASSOBLOCKSIZE = (number[K] [,number[K]] ... )

M    ASSOSIZE = (number[B|M] [,number[B|M]]...)

D    DATABLOCKSIZE = (number[K] [,number[K]] ... )

M    DATASIZE = (number[B|M] [,number[B|M]]...)

     DBID = number

D    NAME {=|:} string

M    SORTSIZE = (number[M] [,number[M]] ... )

D    SYSFILES = (number, number, number)
```

```
M    TEMPSIZE = (number[M] [,number[M]] ... )

D    WORKBLOCKSIZE = number[K]

M    WORKSIZE = number[M | B]
```

ASSOBLOCKSIZE

```
ASSOBLOCKSIZE = (number[K] [,number[K]] ... )
```

このパラメータは、アソシエータコンテナファイルに対して使用されるブロックサイズを指定するものです。最初のブロックサイズはASSO1に、2番目のブロックサイズがASSO2にというように順次対応していきます。

ブロックサイズが指定されない場合には、デフォルトの4Kが使用されます。

ASSO1の場合は、2Kから8Kのブロックサイズだけを指定できます。ASSO2からASSOnの場合は、1Kから32Kのブロックサイズを指定できます。

注意: ASSOBLOCKSIZEパラメータは、指定したASSOSIZEごとに1回ずつ指定する必要があります。このパラメータはペアで指定する必要があります。ASSOSIZEの指定回数がASSOBLOCKSIZEの指定回数よりも多い場合、ブロックサイズが指定のものと違うコンテナに、最後に指定したブロックサイズが適用されます。ASSOBLOCKSIZEを指定しない場合は、デフォルト値が使用されます。

ASSOSIZE

このパラメータは、アソシエータに対して割り当てられるブロック数またはメガバイト数を指定するものです。

アソシエータが複数の物理ファイル内に含まれる場合、各ファイルのサイズを指定する必要があります。

数値に"B"が付けられている場合、サイズはブロック単位になります。それ以外の場合はメガバイト単位になります。

DATABLOCKSIZE

```
DATABLOCKSIZE = (number[K] [,number[K]] ... )
```

このパラメータは、データストレージコンテナファイルに対して使用されるブロックサイズを指定するものです。最初のブロックサイズはDATA1に、そして2番目のサイズがDATA2にというように順次対応します。

ブロックサイズを指定しない場合には、デフォルトの32Kが使用されます。

注意: DATABLOCKSIZEパラメータは、指定したDATASIZEごとに1回ずつ指定する必要があります。このパラメータはペアで指定する必要があります。DATASIZEの指定回

数がDATABLOCKSIZEの指定回数よりも多い場合、ブロックサイズが指定のものと違うコンテナに、最後に指定したブロックサイズが適用されます。DATABLOCKSIZE を指定しない場合は、デフォルト値が使用されます。

DATASIZE

```
DATASIZE = (number[B|M] [,number[B|M]]...) ↩
```

このパラメータは、データストレージに対して割り当てられるブロック数またはメガバイト数を指定するものです。

データストレージが複数の物理ファイル内に含まれる場合、各ファイルのサイズを指定する必要があります。

数値に"B"が付けられている場合、サイズはブロック単位になります。それ以外の場合はメガバイト単位になります。

DBID

```
DBID = number
```

このパラメータは、使用対象データベースを選択するためのものです。

最小値は 1 で、最大値は 255 です。

注意: このパラメータは、ASSO、DATA および WORK をフォーマットする場合にのみ必要とされるものです。SORT または TEMP のフォーマットする場合には使用しないでください。

NAME

```
NAME {=|:} string
```

このパラメータには、データベースに対して割り当てられる名前を指定します。この名前は、レポートユーティリティ ADAREP によって作成されるデータベースステータスレポートのタイトル内に表示されます。等号を指定する場合、「文字列」に指定された値が大文字に変換されます。コロンを指定した場合、大文字の変換は実行されません。

最大 16 文字まで指定できます。

このパラメータが省略された場合、GENERAL-DATABASEがデフォルト値として割り当てられます。

SORTSIZE

```
SORTSIZE = (number[M] [,number[M]] ... )
```

このパラメータには、SORTデータセットに対して割り当てられるメガバイト数を指定します。

SORTが複数のエクステントを持っている場合は、各エクステントのサイズを指定する必要があります。エクステントは50まで指定できます。SORTデータセットは単独でフォーマットされます。

SYSFILES

```
SYSFILES = (number, number, number)
```

このパラメータは、Adabasシステムファイルに対して確保されるファイル番号を指定するものです。ここで指定したファイル番号は、ユーザーファイルに使用することはできません。

最初の番号は、チェックポイントファイルのファイル番号を指定します。

2番目の番号はセキュリティファイルのファイル番号を指定します。

3番目の番号はユーザーデータファイルのファイル番号を指定します。

デフォルトは、SYSFILES=(1,2,3)です。

TEMPSIZE

```
TEMPSIZE = (number [M] [,number[M]] ... )
```

このパラメータには、TEMPxに対して割り当てられるメガバイト数を指定します。

TEMPが複数の物理ファイルから構成される場合、各々のファイルのサイズを指定する必要があります。

この構成要素は、単独でフォーマットされます。

WORKBLOCKSIZE

```
WORKBLOCKSIZE = number[K]
```

このパラメータは、WORKファイルに対して使用されるブロックサイズを指定するものです。

ブロックサイズを指定しない場合には、デフォルトの16Kが使用されます。

WORKSIZE

```
WORKSIZE = number [B|M]
```

このパラメータには、WORK1に対して割り当てられるブロック数またはメガバイト数を指定します。

数値に"B"が付けられている場合、サイズはブロック単位になります。それ以外の場合はメガバイト単位になります。

再スタートに関する考慮事項

ADAFRM は、再スタート機能を備えていません。ADAFRM の処理が中断した場合は、再び最初から始めなければいけません。アソシエータ、データストレージ、および WORK は必ず一緒にフォーマットしなければなりません。

中断した処理で作成されたファイルをまず削除する必要があります。

コントロールステートメントの例

例：データベースのフォーマット

```
adafrm: dbid = 1, name = DATABASE_1
adafrm: assosize = (200M, 100M), assoblocksize = (2K, 4K)
adafrm: datasize = (500M, 500M, 2M), datablocksize = (4K, 16k)
adafrm: worksize = 50M, workblocksize = 16K
```

新規データベースの識別番号は 1 で、実際の名前は DATABASE_1 です。2 つの ASSO コンテナファイルが作られます。ASSO1 は、ブロックサイズが 2 キロバイトで、大きさは 200 メガバイトです。ASSO2 は、ブロックサイズが 4 キロバイトで、大きさは 100 メガバイトです。3つの DATA コンテナファイルがあり、DATA1 と DATA3 は 4 キロバイトのブロックサイズ、DATA2 は 16 キロバイトのブロックサイズです。WORK コンテナファイルは 1 つで、16 キロバイトのブロックサイズです。ファイル番号の 1～3 はそれぞれシステムファイルとして使用されます。

例：**SORT** および **TEMP** のフォーマット

```
adafrm: sortsize = (10M,10M)
adafrm: tempsize = 10M
```

説明：2 つのコンテナファイルが SORT1 と SORT2 として、それぞれ 10 メガバイトの長さでフォーマットされます。TEMP も TEMP1 として、10メガバイトの長さでフォーマットされます。

18 ADAINV（インバーテッドリストの作成、削除および検証）

この章では ADAINV ユーティリティについて説明します。

機能概要

インバーテッドリストユーティリティ ADAINV は、データベース内にロードされたファイルに対するインバーテッドリストの作成、削除および検証を行います。Adabas ニュークリアスがアクティブである必要はありません。ADAINV の実行中に、ニュークリアスはアクティブな場合も、シャットダウンされている場合もあります。利用できる機能は次のとおりです。

- INVERT 機能は、新しいディスクリプタを確立します。
- REINVERT 機能は、RELEASE および INVERT を実行します。
- RELEASE 機能は、既存のディスクリプタを削除します。
- RESET_UQ 機能は、ディスクリプタのユニークステータスを解除します。
- SET_UQ 機能は、既存のディスクリプタをユニークステータスに設定します。
- SUMMARY 機能は、指定ディスクリプタのスペースの全体的な状態と、これらのディスクリプタを処理するために必要なサイズを表示します。
- VERIFY 機能は、インバーテッドリストの整合性をチェックします。

LOB ファイルは、REINVERT 機能、SUMMARY 機能、VERIFY 機能のみに指定できます。

これらの機能は相互に排他的であり、このユーティリティの 1 回の実行では、これらの機能の 1 つのみを実行できます。

注意:

1. 照合ディスクリプタ用の ADAINV INVERT または REINVERT を実行すると、サポートされている ICU の最も高いバージョンを使用して照合ディスクリプタが作成されます（Adabas バージョン 6.5 の場合：ICU 5.4、古い ADABAS バージョンの場合：ICU 3.2）。
2. ICU バージョン 3.2 で作成された照合ディスクリプタを再インバートして ICU バージョン 5.4 にアップグレードする場合、照合指定の構文またはセマンティクスが異なる場合があります。例えば、ICU バージョン 3.2 ではロケール「fr」が FRENCH オプションを意味しますが、ICU バージョン 5.4 では FRENCH オプションを明示的に指定する必要があります。このような場合、ADAINV REINVERT を実行する代わりに、まず照合ディスクリプタの ADAINV RELEASE を実行し、新しい指定で ADAINV INVERT を実行します（これは、古い ICU バージョン 3.2 の指定と同じです）。

ADAINV がオンラインモードで実行されている場合は、ADAINV の実行中に、ファイルにアクセスする Adabas ユーザーがアクティブになっている可能性があります。

機能	ACC ユーザー	UPD ユーザー
INVERT	○	×
REINVERT	×	×
RELEASE	×	×
RESET_UQ	○	×
SET_UQ	○	×
SUMMARY	○	○
VERIFY	○	×

Adabas ユーザーは、ファイルの読み取りまたは検索処理のみを実行した場合、または適切なオープンコマンドを実行した場合、ACC ユーザーとなります。また、ファイルの挿入、更新、または削除処理を実行した場合、ファイルの ISN を排他的ホールド状態にした場合、または適切なオープンコマンドを実行した場合は、UPD ユーザーになります。

上記の表で定義されているように、許可されていないファイルのユーザーが存在する場合、ADAINV はエラー ADA048 で失敗します。

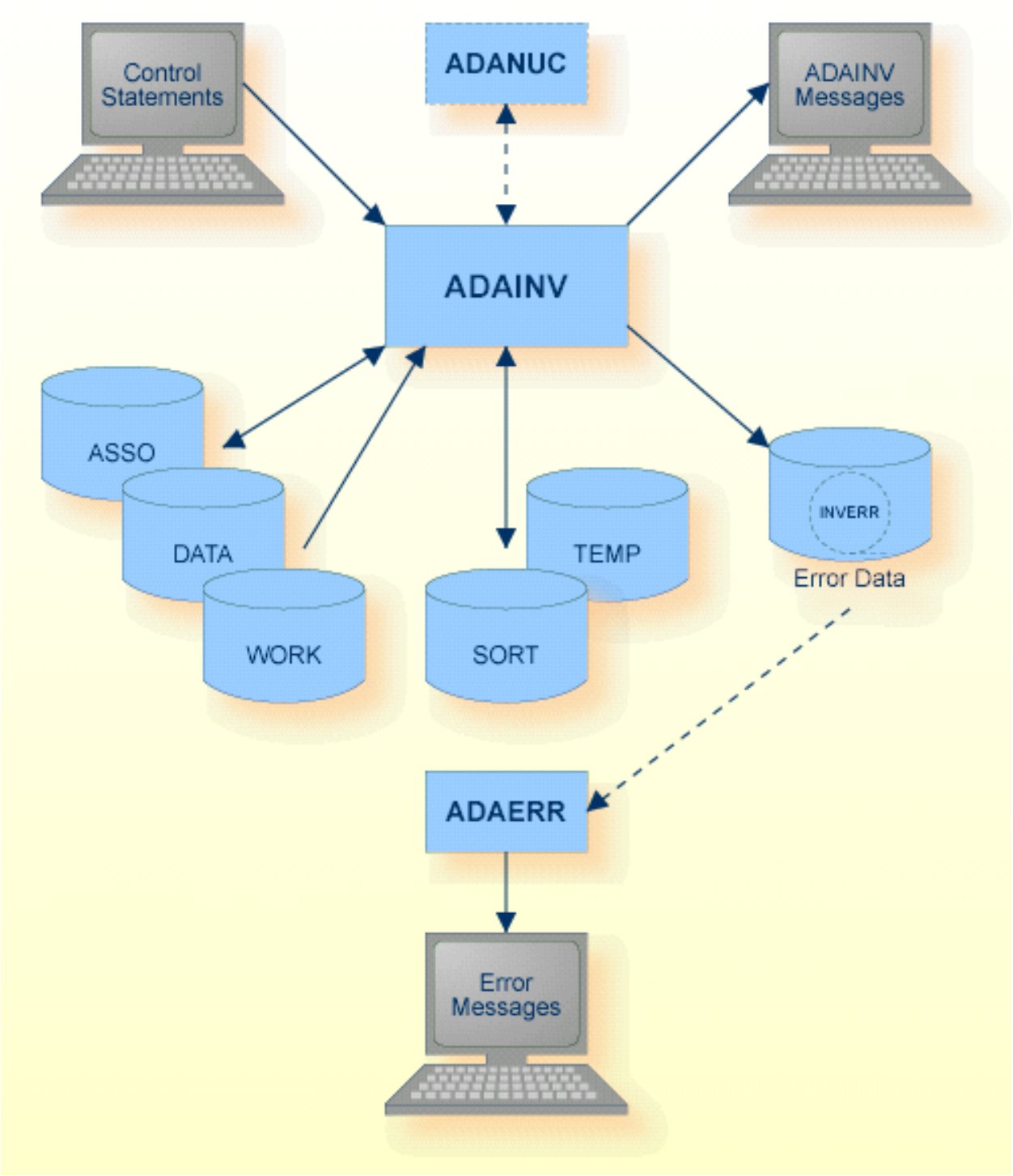
注意:

1. UPD ユーザーは、現在のトランザクションの終了後、ユーザーセッションの終了まで UPD ユーザーのままになります。
2. ADAOPR DISPLAY=UQ_FILES コマンドを使用すれば、現在どのユーザーがファイルにアクセスしているかを確認できます。

エラーファイルにレコードが書き込まれると、ユーティリティはゼロ以外のステータスで終了します。

このユーティリティは単一機能ユーティリティです。

処理フロー


Control Statements
ADANUC
ADAINV Messages
ADAINV
ASSO
DATA
WORK
TEMP
SORT
INVERR
Error Data
ADAERR
Error Messages

シーケンシャルファイル INVERR は複数エクステントを持つことがあります。複数のエクステントを持つシーケンシャルファイルの詳細については、『*Adabas Basics*』の「*ユーティリティの使用*」を参照してください。

データセット	環境 変数／ 論理名	記憶 媒体	追加情報
アソシエータ	ASSOx	ディスク	
データストレージ	DATAx	ディスク	
拒否データ	INVERR	ディスク、テープ（* 注参照）	ADAINVの出力
ソートストレージ	SORTx TEMPLOCx	ディスク	Adabas Basics マニュアル、一時ワークスペース
コントロールステートメント	stdin/ SYS$INPUT		ユーティリティマニュアル
ADAINV メッセージ	stdout/ SYS$OUTPUT		メッセージおよびコード
一時ストレージ	TEMPx	ディスク	
ワークストレージ	WORK1	ディスク	

注意: （*）このシーケンシャルファイルには、名前付きパイプを使用できます（OpenVMS にはない。詳細については、『Adabas Basics』の「ユーティリティの使用」を参照）。

ニュークリアスがアクティブでなく、かつ保留中の AUTORESTART がない場合は、WORK コンテナのパス名またはRAWディスクセクションに環境変数／論理名TEMP1を設定して、WORK を TEMP として使用できます。

チェックポイント

チェックポイント

次の表は、各機能に対するニュークリアス条件と記録チェックポイントを示しています。

機能	ニュークリアスのアクティブ化が必要	ニュークリアスの非アクティブ化が必要	ニュークリアスは不要	記録チェックポイント	有効なニュークリアス操作
INVERT			X	SYNP	R
REINVERT		X（* 注参照）	X	SYNP	
RELEASE			X	SYNP	R
RESET_UQ			X	SYNP	R

機能	ニュークリアスのアクティブ化が必要	ニュークリアスの非アクティブ化が必要	ニュークリアスは不要	記録チェックポイント	有効なニュークリアス操作
SET_UQ			X	SYNP	R
SUMMARY			X		W
VERIFY		X（* 注参照）	X	SYNX	R

注意: （*）Adabas システムファイルの処理時。

R：処理するファイルに許可される読み取り操作。
W：処理するファイルに許可される読み取り操作と書き込み操作。

制御パラメータ

次のコントロールパラメータを使用できます。

```
M   DBID = number

    INVERT = number,
            FIELDS {field_name [,UQ] [,TR] | derived_descriptor_definition | FDT},
              ... [END_OF_FIELDS]
              [,FDT]
D             [,LWP = number[K|M]]
D             [,UQ_CONFLICT = keyword]

D   [NO]LOWER_CASE_FIELD_NAMES

    REINVERT = number,
                {ALL_FIELDS | FIELDS {descriptor_name | FDT}, ...  [END_OF_FIELDS]}
                [,FDT]
D               [,[NO]FORMAT]
D               [,LWP = number[K|M]]
D               [,UQ_CONFLICT = keyword]

    RELEASE = number,
               {ALL_FIELDS | FIELDS {descriptor_name | FDT}, ...  [END_OF_FIELDS]}
               [,FDT]
D              [,[NO]FORMAT]

    RESET_UQ = number,
                {ALL_FIELDS | FIELDS {descriptor_name | FDT}, ...  [END_OF_FIELDS]}
                [,FDT]

    SET_UQ = number,
             {ALL_FIELDS | FIELDS {descriptor_name | FDT}, ...  [END_OF_FIELDS]}
             [,FDT]
```

```
D            [,UQ_CONFLICT = keyword]

    SUMMARY = number,
               {ALL_FIELDS | FIELDS
               {descriptor_name | derived_descriptor_definition | FDT},
               ...  [END_OF_FIELDS]}
               [,FDT]
D              [,FULL]

    VERIFY = number,
              {ALL_FIELDS | FIELDS {descriptor_name | FDT}, ...  [END_OF_FIELDS]}
D             [,ERRORS = number]
              [,FDT]
D             [,LWP = number[K|M]]
```

DBID

```
DBID = number
```

このパラメータは、使用対象となるデータベースを選択するためのものです。

INVERT

```
INVERT = number,
         FIELDS {field_name [,UQ] [,TR] | derived_descriptor_definition | FDT},
          ... [END_OF_FIELDS]
         [,FDT]
         [,LWP = number[K|M]]
         [,UQ_CONFLICT = keyword]
```

この機能は、ファイルが初めてロードされた後の任意の時点で、新しいエレメンタリディスクリプタ、サブディスクリプタ、スーパーディスクリプタ、ハイパーディスクリプタ、フォネティックディスクリプタ、および照合ディスクリプタを確立するものです。"number" には、インバート対象フィールドを含むファイルを指定します。LOB ファイルの数を指定することはできません。

FDT

このパラメータは、選択したファイルの FDT を表示します。このパラメータは、フィールド指定リストの前または中に指定できます。

FIELDS {field_name [,UQ] [,TR] | derived_descriptor_definition | FDT}, ...[END_OF_FIELDS]

このパラメータは、インバート対象フィールドを指定します。このパラメータには、次の要素を任意の組み合わせで指定できます。

- フィールド名
- フォネティックディスクリプタ
- サブディスクリプタ、スーパーディスクリプタ、ハイパーディスクリプタ、照合ディスクリプタ

各要素は別々の行に指定します。フィールド名、フォネティックディスクリプタ、サブディスクリプタ、スーパーディスクリプタ、ハイパーディスクリプタ、または照合ディスクリプタの有効な指定については、『*管理マニュアル*』の「*FDT のレコード構造*」を参照してください。

オプション UQ と TR の用途は、目的のフィールドがユニークディスクリプタかどうか、またはインデックス切り捨てが実行されるかどうかを指定することです。UQ オプションと TR オプションの詳細については、『*管理マニュアル*』の「*定義オプション*」を参照してください。

注意: 基本ファイルに値が保存されているフィールドのみを、ディスクリプタとして、または派生ディスクリプタの親フィールドとして使用できます。このため、インバートするフィールド、または作成する派生ディスクリプタの親フィールドに LA オプションか LB オプションがあって、値が LOB ファイルに保存される場合、INVERT 機能は異常終了します。LA フィールドと LB フィールドは、ディスクリプタとして、または派生ディスクリプタの親フィールドとして使用できますが、その場合はすべての値が 16 KB - 3 に制限され、LA フィールド値か LB フィールド値を含む基本レコードが 1 つのデータブロックに収まる必要があります。

END_OF_FIELDS パラメータを使用してフィールド定義を消去する場合、LOWER_CASE_FIELD_NAMES パラメータを使用するときに、このパラメータを大文字で指定する必要があります。また、LOWER_CASE_FIELD_NAMES を使用する場合は、FDT パラメータも大文字で指定する必要があります。

LWP = number[K|M]

いくつかのディスクリプタ値に対して、ADAINV がメモリ内のワークプールを使用します。多くの場合、ワークプールのデフォルトサイズは、ADAINV に最適なパフォーマンスをもたらします。LWP パラメータにより、ワークプールを増やすことができます。デフォルトのワークプールサイズに追加されるスペースをバイト、キロバイト（K）、またはメガバイト（M）単位で定義します。

ワークプールサイズを大きくすると、次の場合に役立ちます。

- お使いの環境で、大規模なワークプールを使用するとパフォーマンスが向上する場合。
- SORT コンテナがディスクリプタ値をソートするには小さすぎる場合。適切な LWP パラメータにより、SORT コンテナの必要なサイズを縮小できます。

このパラメータに必要な値は、SUMMARY 機能を使用して特定できます。

UQ_CONFLICT = keyword

このパラメータは、ユニークディスクリプタに重複する値が見つかった場合に実行されるアクションを特定します。"keyword" に使用できる値は ABORT または RESET です。ABORT を指定した場合、重複する UQ ディスクリプタ値が見つかると ADAINV は実行を終了し、エラーステータスを返します。RESET を指定した場合、問題のディスクリプタの UQ ステータスが削除され、処理が続行されます。

デフォルトは UQ_CONFLICT = ABORT です。

[NO]LOWER_CASE_FIELD_NAMES

```
[NO]LOWER_CASE_FIELD_NAMES
```

LOWER_CASE_FIELD_NAMES が指定されている場合、Adabas フィールド名は、大文字に変換されません。NOLOWER_CASE_FIELD_NAMES が指定されている場合、Adabas フィールド名は、大文字に変換されます。デフォルトは、NOLOWER_CASE_FIELD_NAMES です。

このパラメータは、FIELDS パラメータの前に指定する必要があります。

REINVERT

```
REINVERT = number,
           {ALL_FIELDS | FIELDS {descriptor_name | FDT}, ...  [END_OF_FIELDS]}
           [,FDT]
           [,[NO]FORMAT]
           [,LWP = number[K|M]]
           [,UQ_CONFLICT = keyword]
```

この機能は、RELEASE および INVERT を実行します。これは、入力ミス、特にサブディスクリプタとスーパーディスクリプタに対する入力ミスの可能性を減少させます。

注意: ADAINV REINVERT の目的は、多数の更新の結果としてインデックスツリーのバランスが崩れた場合、またはインデックスエラーが発生した場合に、ディスクリプタを再作成することです。ディスクリプタは、常に前と同じ定義で再作成されます。例えば、スーパーディスクリプタなど、ディスクリプタの定義を変更する場合は、ADAINV RELEASE の後に、新しいディスクリプタ定義で ADAINV INVERT を実行する必要があります。

ALL_FIELDS

このパラメータは、選択ファイルの全ディスクリプタを解放／インバートすることを指定します。

FDT

このパラメータは、選択したファイルの FDT を表示します。このオプションは、フィールド指定リストの前またはリスト内に指定できます。

FIELDS {descriptor_name | FDT}, ...[END_OF_FIELDS]

このパラメータは、解放／再インバートされるディスクリプタを指定します。その後には、1つまたは複数のフィールド名を指定できます。各フィールド名は独立した行に指定します。有効なフィールド名の指定方法については、『*管理マニュアル*』の「*FDT レコード構造*」を参照してください。

END_OF_FIELDS パラメータを使用してフィールド定義を消去する場合、LOWER_CASE_FIELD_NAMES パラメータを使用するときに、このパラメータを大文字で指定する必要があります。また、LOWER_CASE_FIELD_NAMESを使用する場合は、FDTパラメータも大文字で指定する必要があります。

[NO]FORMAT

ディスクリプタが解放または再インバートされた場合、通常古いインデックスより小さく、必要なディスクスペースが少ない新しいインデックスが作成されます。FORMAT オプションは、インデックスによって使用されなくなったにもかかわらず、まだファイルに割り当てられているブロックをフォーマットするために使用します。

デフォルトは NOFORMAT です。

LWP = number[K|M]

いくつかのディスクリプタ値に対して、ADAINV がメモリ内のワークプールを使用します。多くの場合、ワークプールのデフォルトサイズは、ADAINV に最適なパフォーマンスをもたらします。LWPパラメータにより、ワークプールを増やすことができます。デフォルトのワークプールサイズに追加されるスペースをバイト、キロバイト（K）、またはメガバイト（M）単位で定義します。

ワークプールサイズを大きくすると、次の場合に役立ちます。

- お使いの環境で、大規模なワークプールを使用するとパフォーマンスが向上する場合。
- SORT コンテナがディスクリプタ値をソートするには小さすぎる場合。適切な LWP パラメータにより、SORT コンテナの必要なサイズを縮小できます。

このパラメータに必要な値は、SUMMARY 機能を使用して特定できます。

UQ_CONFLICT = keyword

このパラメータは、ユニークディスクリプタに重複する値が見つかった場合に実行されるアクションを特定します。"keyword" に使用できる値は ABORT または RESET です。ABORT を指定した場合、重複する UQ ディスクリプタ値が見つかると ADAINV は実行を終了し、エラーステータスを返します。RESET を指定した場合、問題のディスクリプタの UQ ステータスが削除され、処理が続行されます。

デフォルトは UQ_CONFLICT = ABORT です。

RELEASE

```
RELEASE = number,
          {ALL_FIELDS | FIELDS {descriptor_name | FDT}, ...  [END_OF_FIELDS]}
          [,FDT]
          [,[NO]FORMAT]
```

この機能は、"number" で指定するファイルからエレメンタリディスクリプタ、サブディスクリプタ、スーパーディスクリプタ、ハイパーディスクリプタ、フォネティックディスクリプタ、および照合ディスクリプタを除去するものです。LOB ファイルの数を指定することはできません。

ALL_FIELDS

このパラメータは、選択ファイルの全ディスクリプタを解放することを指定します。

FDT

このパラメータは、選択したファイルの FDT を表示します。このオプションは、フィールド指定リストの前またはリスト内に指定できます。

FIELDS {descriptor_name | FDT}, ...[END_OF_FIELDS]

このパラメータは、解放されるディスクリプタを指定します。その後には、1 つまたは複数のフィールド名を指定できます。各フィールド名は独立した行に指定します。有効なフィールド名の指定方法については、『*管理マニュアル*』の「*FDT レコード構造*」を参照してください。

END_OF_FIELDS パラメータを使用してフィールド定義を消去する場合、LOWER_CASE_FIELD_NAMES パラメータを使用するときに、このパラメータを大文字で指定する必要があります。また、LOWER_CASE_FIELD_NAMES を使用する場合は、FDT パラメータも大文字で指定する必要があります。

[NO]FORMAT

ディスクリプタが解放または再インバートされた場合、通常古いインデックスより小さく、必要なディスクスペースが少ない新しいインデックスが作成されます。FORMAT オプションは、インデックスによって使用されなくなったにもかかわらず、まだファイルに割り当てられているブロックをフォーマットするために使用します。

デフォルトは NOFORMAT です。

RESET_UQ

```
RESET_UQ = number,
           {ALL_FIELDS | FIELDS {descriptor_name | FDT}, ...  [END_OF_FIELDS]}
           [,FDT]
```

この機能は、"number" で指定するファイル内に定義されたエレメンタリディスクリプタ、サブディスクリプタ、ハイパーディスクリプタ、スーパーディスクリプタ、および照合ディスクリプタのユニークステータスを解除します。LOB ファイルの数を指定することはできません。

ALL_FIELDS

このパラメータは、指定ファイルにあるすべてのユニークディスクリプタからユニークステータスを解除することを指定するものです。

FDT

このパラメータは、選択ファイルのフィールド定義テーブル（FDT）を表示します。このオプションは、フィールド指定リストの前またはリスト内に指定できます。

FIELDS {descriptor_name | FDT}, ...[END_OF_FIELDS]

このパラメータは、ユニークステータスを解除するディスクリプタを指定します。その後には、1つまたは複数のフィールド名を指定できます。各フィールド名は独立した行に指定します。有効なフィールド名の指定方法については、『*管理マニュアル*』の「*FDT レコード構造*」を参照してください。

END_OF_FIELDS パラメータを使用してフィールド定義を消去する場合、LOWER_CASE_FIELD_NAMES パラメータを使用するときに、このパラメータを大文字で指定する必要があります。また、LOWER_CASE_FIELD_NAMESを使用する場合は、FDT パラメータも大文字で指定する必要があります。

SET_UQ

```
SET_UQ = number,
         {ALL_FIELDS | FIELDS {descriptor_name | FDT}, ...  [END_OF_FIELDS]}
         [,FDT]
         [,UQ_CONFLICT = keyword]
```

この機能は、"number" で指定するファイル内に定義されたエレメンタリディスクリプタ、サブディスクリプタ、ハイパーディスクリプタ、スーパーディスクリプタ、および照合ディスクリプタにユニークステータスを設定します。LOB ファイルの数を指定することはできません。

ALL_FIELDS

このパラメータは、指定ファイル内に定義されたすべてのエレメンタリディスクリプタ、サブディスクリプタ、ハイパーディスクリプタ、スーパーディスクリプタ、および照合ディスクリプタにユニークステータスを設定することを指定します。

FDT

このパラメータは、選択したファイルの FDT を表示します。このオプションは、フィールド指定リストの前またはリスト内に指定できます。

FIELDS {descriptor_name | FDT}, ...[END_OF_FIELDS]

このパラメータは、ユニークステータスを設定するディスクリプタを指定します。その後には、1つまたは複数のフィールド名を指定できます。各フィールド名は独立した行に指定します。有効なフィールド名の指定方法については、『*管理マニュアル*』の「*FDT レコード構造*」を参照してください。

END_OF_FIELDS パラメータを使用してフィールド定義を消去する場合、LOWER_CASE_FIELD_NAMES パラメータを使用するときに、このパラメータを大文字で指定する必要があります。また、LOWER_CASE_FIELD_NAMESを使用する場合は、FDTパラメータも大文字で指定する必要があります。

UQ_CONFLICT = keyword

このパラメータは、ユニークディスクリプタに重複する値が見つかった場合に実行されるアクションを特定します。"keyword" に使用できる値は ABORT または RESET です。ABORT を指定する場合、重複するディスクリプタ値が検出されると、ADAINV は実行を終了してエラーステータスを返します。RESET を指定する場合、重複するディスクリプタ値があってもユニークステータスにはならず、処理が続行されます。

デフォルトは UQ_CONFLICT=ABORT です。

SUMMARY

```
SUMMARY = number,
          {ALL_FIELDS | FIELDS
          {descriptor_name | derived_descriptor_definition | FDT},
           ...  [END_OF_FIELDS]}
          [,FDT]
          [,FULL]
```

この機能は、指定したディスクリプタに対するディスクリプタスペースサマリ（DSS）と、ディスクリプタを処理するために必要なサイズを表示します。

注意: 表示されるサイズの精度は、状況に応じていろいろと変わります。正確なサイズは、表示される値よりも少し小さ目のサイズになることもあります。SUMMARY 機能の実行中または実行後にファイルが更新された場合には、表示される値がかなり小さなサイズになることもあります。

ADAINV SUMMARY処理の詳細については、『*管理マニュアル*』の「*ADAMUP およびADAINV 処理の最適化*」を参照してください。

ALL_FIELDS

このパラメータを指定すると、選択されたファイルの全ディスクリプタがチェックされます。

FDT

このパラメータは、選択したファイルの FDT を表示します。このオプションは、フィールド指定リストの前またはリスト内に指定できます。

FIELDS {descriptor_name | derived_descriptor_definition | FDT}, ...[END_OF_FIELDS]

このパラメータは、ユニークステータスを設定するディスクリプタを指定します。それぞれ個別の行で開始している1つ以上のフィールド名、フォネティックディスクリプタ、サブディスクリプタ、スーパーディスクリプタ、ハイパーディスクリプタ、または照合ディスクリプタの前に指定できます。ディスクリプタであるフィールドやディスクリプタでないフィールドを指定することができます。有効なフィールド名の指定方法については、『*管理マニュアル*』の「*FDT レコード構造*」を参照してください。

END_OF_FIELDS パラメータを使用してフィールド定義を消去する場合、LOWER_CASE_FIELD_NAMES パラメータを使用するときに、このパラメータを大文字で指定する必要があります。また、LOWER_CASE_FIELD_NAMESを使用する場合は、FDTパラメータも大文字で指定する必要があります。

FULL

これを指定すると、各ディスクリプタの表示に、ディスクリプタに必要なサイズが加わります。指定するフィールドすべてを処理するとは限らない場合、これは有用です。

VERIFY

```
VERIFY = number,
         {ALL_FIELDS | FIELDS {descriptor_name | FDT}, ...  [END_OF_FIELDS]}
         [,ERRORS = number]
         [,FDT]
         [,LWP = number[K|M]]
```

この機能は、"number" に指定するファイルのインバーテッドリストの整合性をチェックするものです。

ALL_FIELDS

このパラメータは、選択ファイルの全ディスクリプタがチェックされることを指定します。

ERRORS = number

このパラメータは、ディスクリプタの検証を終了するまでに何件エラーをレポートするかを指定します。

デフォルトは、20 です。

FDT

このパラメータは、選択したファイルの FDT を表示します。このオプションは、フィールド指定リストの前またはリスト内に指定できます。

FIELDS {descriptor_name | FDT}, ...[END_OF_FIELDS]

このパラメータは、検証対象ディスクリプタフィールドを指定します。その後には、1つまたは複数のフィールド名を指定できます。各フィールド名は独立した行に指定します。有効なフィールド名の指定方法については、『*管理マニュアル*』の「*FDT レコード構造*」を参照してください。

END_OF_FIELDS パラメータを使用してフィールド定義を消去する場合、LOWER_CASE_FIELD_NAMES パラメータを使用するときに、このパラメータを大文字で指定する必要があります。また、LOWER_CASE_FIELD_NAMES を使用する場合は、FDT パラメータも大文字で指定する必要があります。

LWP = number[K|M]

いくつかのディスクリプタ値に対して、ADAINV がメモリ内のワークプールを使用します。多くの場合、ワークプールのデフォルトサイズは、ADAINV に最適なパフォーマンスをもたらします。LWP パラメータにより、ワークプールを増やすことができます。デフォルトのワークプールサイズに追加されるスペースをバイト、キロバイト（K）、またはメガバイト（M）単位で定義します。

ワークプールサイズを大きくすると、次の場合に役立ちます。

- お使いの環境で、大規模なワークプールを使用するとパフォーマンスが向上する場合。
- SORT コンテナがディスクリプタ値をソートするには小さすぎる場合。適切な LWP パラメータにより、SORT コンテナの必要なサイズを縮小できます。

このパラメータに必要な値は、SUMMARY 機能を使用して特定できます。

再スタートに関する考慮事項

ADAINV は再スタート機能を備えていません。しかし、異常終了した ADAINV を最初から再実行できるかどうかは状況により異なります。

ADAINV が異常終了した場合、通常、再スタートすることができます。しかし、ADAINV がインデックスを修正した場合、次の点を考慮する必要があります。

- REINVERT ...FIELDS 機能は、RELEASE ...FIELDS 機能の後に INVERT ...FIELDS 機能が続いているものと同じです。そのため、ADAINV が INVERT フェーズで異常終了した場合は、処理を再スタートするために INVERT ...FIELDS 機能を実行してください。
- ADAINV をオフラインで実行する場合、ディスクに書き込まれる、論理的な単位を形成するレコードの件数が少ないため、ディスクへの書き込み時間は非常に短くなります。これらのレコードの最初のレコードが書き込まれてから、最後のレコードの書き込みが終了するまでの間に、ADAINV が終了すると、ADAINV は再スタートできません。この場合、REINVERT ...ALL_FIELDS 機能が必要です。このような事態は、ADAINV をオンラインで実行した場合には起きません。
- ADAINV が異常終了すると、インデックスブロックが失われることがあります。これらのインデックスブロックは REINVERT ...ALL_FIELDS 機能、ADAORD ユーティリティの使用、または ADAULD と ADAMUP ユーティリティの使用によってのみリカバリできます。

例

例 1

```
adainv: dbid=1
adainv: invert=10, fields
adainv: HO
```

データベース 1 のファイル 10 にあるエレメンタリフィールド HO がインバートされます。

例 2

```
adainv: dbid=1
adainv: invert=10
adainv: lwp=600k
adainv: fields
adainv: ph=phon(na)
adainv: sp=na(1,3),yy(1,2),uq
adainv: bb,uq
```

データベース1のファイル10に対して、新しいディスクリプタが作成されます。PHは、フィールド NA から派生したフォネティックディスクリプタです。SP は、フィールド NA のバイト 1 から 3 およびフィールド YY の 1 から 2 から派生するユニークスーパーディスクリプタです。エレメンタリフィールド BB はディスクリプタステータスに変更され、ユニークフラグが設定されます。ソートに使用されるワークプールの大きさは 600K に増えます。

例 3

```
adainv: dbid=1
adainv: release=10
adainv: fields
adainv: ho
adainv: ph
```

上記の 2 つのディスクリプタ HO および PH が解除されます。

例 4

```
adainv: dbid = 1, verify = 10
adainv: errors = 5
adainv: fields
adainv: sp
adainv: na
adainv: end_of_fields
```

ディスクリプタ SP および NA が検証されます。ディスクリプタ NA に対して生成されたディスクリプタバリューテーブルエントリと、このフィールドの非圧縮値が照合されます。各ディスクリプタのエラー数が 5 つを超えると、検証は終了します。

例 5

```
adainv: dbid = 1, reinvert = 10
adainv: fields
adainv: na
```

データベース 1 のファイル 10 にあるディスクリプタ NA が再インバートされます（これは、例 4 でエラーが検出された場合に必要になることがあります）。

例 6

```
adainv: db=12
adainv: reinvert=9
adainv: all_fields
```

データベース 12 のファイル 9 の完全なインデックスが作成されます。

次の出力が生成されます。

```
%ADAINV-I-FILE, file 9, EMPLOYEES

%ADAINV-I-UIUPD, upper index being modified

%ADAINV-I-SORTDESC, sorting descriptor KA
%ADAINV-I-LOADDESC, loading descriptor KA

%ADAINV-I-SORTDESC, sorting descriptor S3
%ADAINV-I-LOADDESC, loading descriptor S3

%ADAINV-I-SORTDESC, sorting descriptor S2
%ADAINV-I-LOADDESC, loading descriptor S2

%ADAINV-I-SORTDESC, sorting descriptor PA
%ADAINV-I-LOADDESC, loading descriptor PA
```

```
%ADAINV-I-SORTDESC, sorting descriptor FB
%ADAINV-I-LOADDESC, loading descriptor FB

%ADAINV-I-SORTDESC, sorting descriptor AA
%ADAINV-I-LOADDESC, loading descriptor AA

%ADAINV-I-SORTDESC, sorting descriptor BC
%ADAINV-I-LOADDESC, loading descriptor BC

%ADAINV-I-SORTDESC, sorting descriptor CN
%ADAINV-I-LOADDESC, loading descriptor CN

%ADAINV-I-SORTDESC, sorting descriptor JA
%ADAINV-I-LOADDESC, loading descriptor JA

%ADAINV-I-SORTDESC, sorting descriptor H1
%ADAINV-I-LOADDESC, loading descriptor H1

%ADAINV-I-SORTDESC, sorting descriptor EA
%ADAINV-I-LOADDESC, loading descriptor EA

%ADAINV-I-SORTDESC, sorting descriptor LC
%ADAINV-I-LOADDESC, loading descriptor LC

%ADAINV-I-SORTDESC, sorting descriptor S1
%ADAINV-I-LOADDESC, loading descriptor S1

%ADAINV-I-SORTDESC, sorting descriptor AC
%ADAINV-I-LOADDESC, loading descriptor AC

%ADAINV-I-NULLDESC, no values for descriptor IJ
%ADAINV-I-LOADDESC, loading descriptor IJ

%ADAINV-I-NULLDESC, no values for descriptor IB
%ADAINV-I-LOADDESC, loading descriptor IB

%ADAINV-I-NULLDESC, no values for descriptor FI
%ADAINV-I-LOADDESC, loading descriptor FI

%ADAINV-I-UIUPD, upper index being modified
%ADAINV-I-DSPASSES, data storage passes : 17
%ADAINV-I-REMOVED, dataset SORT1, file C:\Program Files\Software AG\Adabas/db012
\SORT01_3664.012 removed
%ADAINV-I-IOCNT,          1 IOs on dataset SORT
%ADAINV-I-IOCNT,         85 IOs on dataset DATA
%ADAINV-I-IOCNT,         49 IOs on dataset ASSO
```

注意:

1. メッセージ NULLDESC は、このディスクリプタにディスクリプタ値が存在しないことを示します。これは、すべてのレコードでフィールドに空値/SQL 空値が含まれている場合に、オプション NU または NC で定義されたフィールドで発生することがあります。
2. メッセージの DSPASSES には、ファイルのデータレコードが読み込まれた頻度が表示されます。この場合、データストレージパスの数は 17 です。つまり、ディスクリプタ値を保存できる TEMP コンテナが定義されていないので、各ディスクリプタのデータレコードが再読み込みされています。TEMP コンテナを定義して、データストレージパスの数を減らすことができます。これは特に、大きなファイルの場合に推奨されます。必要な I/O 処理の数が大幅に減少します。ADAINV パラメータ SUMMARY を使用して、TEMP コンテナに有益なサイズを確認できます。
3. メッセージ REMOVED は、ADAINV によって作成された一時的な SORT コンテナが削除されたことを示します。また、永続的な SORT コンテナを使用することもできます。これは、ADAINVによって作成および削除されるものではありません（詳細については、「**ADAFRM**」を参照してください）。

例 7

```
adainv: dbid = 1, set_uq=10
adainv: fields
adainv: na
adainv: end_of_fields
adainv: uq_conflict=reset
```

データベース 1 のファイル 10 にあるディスクリプタ NA に対してユニークステータスが設定されます。1 つのディスクリプタ値に対して複数の ISN があると、ISN の不整合がエラーログに書き込まれ、ユニークステータスが解除されます。

例 8

```
adainv: dbid = 1, reset_uq=10
adainv: fields
adainv: sp
```

データベース 1 のファイル 10 にあるディスクリプタ SP からユニークステータスが解除されます。

例 9

```
adainv: db=33
adainv: summary=112
adainv: fields
adainv: ab
adainv: ae
adainv: s1=ap(1,1),aq(1,1),ar(1,1)
adainv: s2=ac(1,3),ad(1,8),ae(1,9)
adainv: s3=ao(2,3)
```

この例の出力結果は次のとおりです。

```
Descriptor summary:
===================

Descriptor AB :           1,194,469 bytes,             581,209 occ
Descriptor AE :           3,605,545 bytes,             538,769 occ
Descriptor S1 :           1,566,501 bytes,             581,209 occ
Descriptor S2 :           1,520,169 bytes,              72,389 occ
Descriptor S3 :           1,340,949 bytes,             446,983 occ


Required sizes to process these descriptors:
============================================
           - SORTSIZE (LWP=          0 KB)        =          8 MB
           - LWP for incore sort                  =     13,230 KB
           - TEMPSIZE (1 pass)                    =         24 MB
           - TEMPSIZE (2 passes)                  =         13 MB
           - TEMPSIZE (recommended minimum size)  =          5 MB

%ADAINV-I-IOCNT,     1710 IOs on dataset DATA
%ADAINV-I-IOCNT,        3 IOs on dataset ASSO
%ADAINV-I-TERMINATED,   24-NOV-2006 14:15:06, elapsed time: 00:04:03
```

19 ADAMON（データベースニュークリアスの監視）

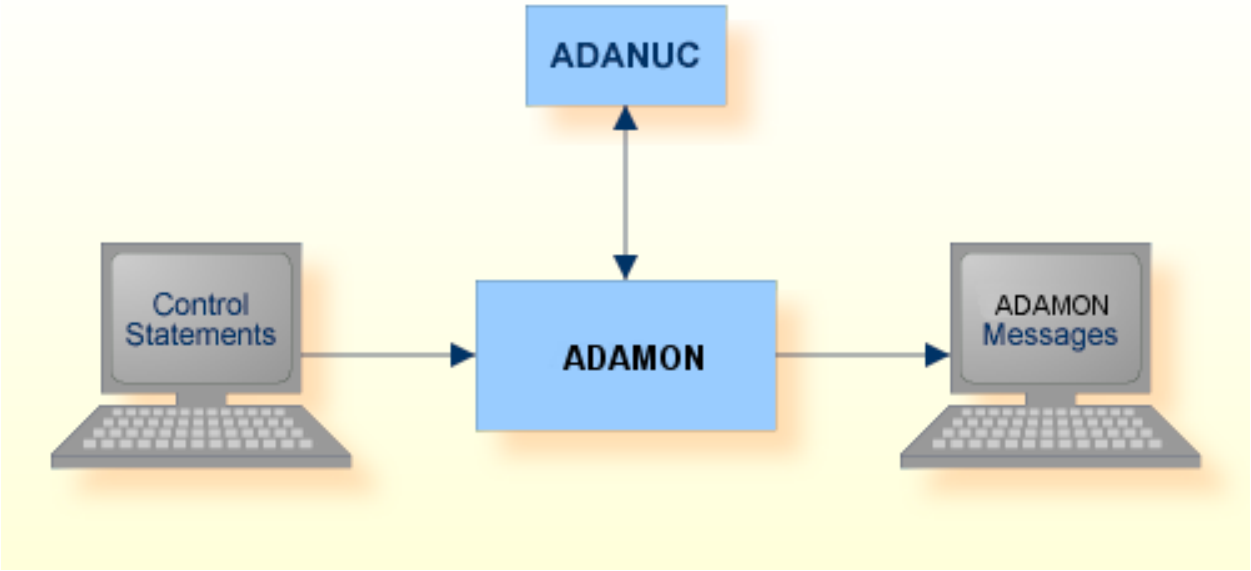
この章では ADAMON ユーティリティについて説明します。

機能概要

ADAMON ユーティリティは、パフォーマンスデータを収集するためのユーティリティで、Adabas セッションの監視に使用します。収集する情報の種類は、DISPLAY パラメータの設定によって決まり、情報を表示するときの出力単位はほとんどのものが秒単位です。収集した情報は、数値の集まりと、グラフィカルベースのどちらでも表示できます。ADAMON セッションは、Ctrl キーを押しながら C キーを押して終了します。LOOP パラメータに指定した値に達したときにも終了します。終了後には、監視したセッションの統計概要が表示されます。

このユーティリティは多機能ユーティリティです。

処理フロー



データセット	環境変数／論理名	記憶媒体	追加情報
コントロールステートメント	stdin/ SYS$INPUT		ユーティリティマニュアル
ADAMON メッセージ	stdout/ SYS$OUTPUT		メッセージおよびコード

チェックポイント

このユーティリティはチェックポイントを書き込みません。

制御パラメータ

次のコントロールパラメータを使用できます。

```
D    [NO]DATETIME

M    DBID = number

D    DISPLAY = keyword

D    [NO]GRAPHICAL

D    INTERVAL = number

D    LOOPS = number

     RCMD

     SUMMARY = filename

     SUMMARY_COMPARE = filename

     SUMMARY_COMPARE_FILES = (filename,filename)

     SUMMARY_INPUT = filename
```

DATETIME

```
[NO]DATETIME
```

このパラメータがDATETIMEに設定されている場合、非グラフィカル出力の各モニタリング行の前に、現在の日付と時刻が表示されます。デフォルトはNODATETIMEです。

DBID

```
DBID = number
```

このパラメータは、使用対象となるデータベースを選択するためのものです。DISPLAY=BACKUPを除くすべての機能でデータベースがアクティブになっている必要があります。

DISPLAY

```
DISPLAY = keyword
```

このパラメータでは、指定したキーワードに従ってデータベース情報が表示されます。表示は、パラメータ INTERVAL で指定した間隔で更新されます（デフォルトは 3 秒）。表示される情報の詳細については、「ADAOPR」セクションにある DISPLAY の例を参照してください。

利用できるキーワードは次のとおりです。

キーワード	説明
ACTIVITY	1 秒あたりのコマンド数など、データベースのスループットを表示します。ニュークリアスがアクティブな場合は、このキーワードがデフォルトになります。
BACKUP	ADABCK DUMP 機能または RESTORE 機能の実行を監視するグラフが表示されます。使用済みサイズをまとめるために、表示される値はブロックサイズに正規化されているので、実際のバックアップスペースとリストアスペースとは異なることがあります。出力は常にグラフィカルベースです。このキーワードは、ニュークリアスがアクティブでない場合も使用できます。ニュークリアスがアクティブでない場合は、このキーワードがデフォルトになります。
HIGH_WATER	重要なピーク時の値を表示します。出力は常にグラフィカルベースです。太字の行はパーセント単位の現在の値であり、ハイフン付きの行はピーク時の値を示します。太字の行のみである場合、現在の値とピーク時の値は同一です。［Write Limit］行には、フラッシュの制限に達するまでに変更されたブロック数がパーセント単位で表示されます。通常は、100 %に達すると、バッファフラッシュが開始されます。行内の数値は、変更されたスペースをバイト単位で示します。［WP1 Flush］行は、前回のバッファフラッシュレコードから変更された WORK パート 1 ブロックの数を示します。100 %に達すると、バッファフラッシュが開始されます。［Hit rate］行は、全体的なヒット率（ハイフン付きの行）、および計測間隔内で発生した現在の率（太字の行）を示します。［ASSO］行と［DATA］行は、使用済みコンテナスペースおよび割り当て済み合計コンテナスペースの間の比率を示します。［PLOG］行は、プロテクションログの使用済みスペースと割り当て済みスペースの比率を示します。このデータセット行の数値は、KB、MB、GB のいずれかの単位になります。ファイルシステムの PLOG が常に 100 %になることに注意してください。
INDEX	一部のカウンタおよびインデックス更新中に発生した例外（内部的な要因に起因する）を表示します。
IO	Adabas コンテナファイルごとの 1 秒間の物理 I/O 数を表示します。コンテナタイプ（ASSO、DATA）ごとに 10 エクステントずつ表示できます。上位のエクステントの I/O は、ASSOx または DATAx の I/O として収集されます。

データ収集中に例外的な状況が検出された場合は、追加情報が画面に表示されます。非グラフィカルモードでは最終列に表示され、グラフィカルモードではステータスがベースラインに表示されます。次のステータス情報を検出できます。

BF_ACTIVE
バッファフラッシュが進行中です。

SPACE_WAIT
複合コマンドでワークプールスペースを待機しているスレッド。

ET_SYNC
ニュークリアスは ET_SYNC モードであり、新しいトランザクションは開始されません。

HYX
ニュークリアスはハイパー出口を実行しています。

UEX
ニュークリアスはユーザー出口を実行しています。

LARGE_DWP
内部ワークプールが大きすぎ、バッファプールに及んでいます。

SHUTDOWN-P
ニュークリアスのシャットダウンが進行中です。

SHUTDOWN-C
ニュークリアスのシャットダウンが完了しました。

CRASHED
ニュークリアスが異常終了しました。

AUTORESTART を実行している場合は、フェーズ番号（1、2、3、4）および処理ブロック数を ADAMON で監視して表示できます。一般的にフェーズ3に最も長い時間がかかり、処理ブロックの割合が表示されます。これは、選択した機能に関係なく実行されます。AUTORESTART が完了すると、PT_RET が実行され、要求された機能に制御が戻ります。

GRAPHICAL

```
[NO]GRAPHICAL
```

このオプションを GRAPHICAL に設定すると、出力がグラフィカルフォーマットに切り替わります。表示機能BACKUP と HIGH_WATER の場合は、グラフィカルフォーマットのみがサポートされます。デフォルトは NOGRAPHICAL です。

INTERVAL

```
INTERVAL = number
```

このパラメータでは、データ収集のサンプリング間隔を秒単位で指定します。

デフォルトの間隔は 3 秒です。

LOOPS

```
LOOPS = number
```

このパラメータでは、データ収集ループの回数を制限します。

デフォルトの場合、ADAMON は継続的にループします。データ収集は、Ctrl キーを押したまま C キーを押して終了できます。

RCMD

パラメータ RCMD は、Adabas-Adabas レプリケーション（A2A）を搭載した Adabas Event Replicator を使用している顧客にのみ関係します。このパラメータを指定すると、さらに 2 つの列が DISPLAY=ACTIVITY の出力に追加されます。

列	説明
Rec. per sec	記録されたコマンドの数
Repl. per sec	レプリケートされたコマンドの数

さらに、ADAMONの実行の最後に、レプリケーションコマンドの数と1秒あたりのレプリケーションコマンドの比率がサマリに追加されます。

注意: ADAMON の開始時に保留中のレプリケーションがなく、かつ ADAMON の終了時に保留中のレプリケーションがない場合でも、「記録されたレプリケーションコマンド」の合計数が「レプリケートされたコマンド」の合計数よりも多くなることがあります。これは、ロールバックされたトランザクションに属するレプリケーションシステムファイルに保存されているコマンドが、レプリケートされることなくレプリケーションシステムファイルから再度削除されることが理由です。

SUMMARY

```
SUMMARY = filename
```

パラメータ SUMMARY を使用して、DISPLAY=ACTIVITY のサマリをファイルに書き込むことができます。結果のファイルはバイナリファイルです。パラメータ INTERVAL および LOOPS は ADAMON のランタイムを制限するために使用されます。

SUMMARY_COMPARE

```
SUMMARY_COMPARE = filename
```

パラメータ SUMMARY_COMPARE を使用して、比較レポートを作成できます。目的は、SUMMARY=filename で参照ファイルを作成してから、この参照ファイルを現在の ADAMON 実行のベースとして使用することです。パラメータ INTERVAL および LOOPS は ADAMON のランタイムを制限するために使用されます。

SUMMARY_COMPARE_FILES

```
SUMMARY_COMPARE_FILES = (filename,filename)
```

パラメータ SUMMARY_COMPARE_FILES を使用して、2 つの ADAMON サマリファイルを比較できます。

SUMMARY_INPUT

```
SUMMARY_INPUT = filename
```

SUMMARY_INPUT パラメータを使用して、SUMMARY=filename で作成されたサマリファイルの内容を表示できます。DBID パラメータは必要ありません。

例

例 **1**：

```
> adamon db=36
%ADAMON-I-STARTED,      29-MAR-2016 11:53:29, Version <version number>

Database 36, startup at 23-MAR-2016 16:58:41
ADANUC Version <version number>, PID 525



Commands      I/Os per second      Throw   Buffer pool
per sec   ASSO  DATA  WORK  PLOG   backs   Hit  Flushs
-----------------------------------------------------
       0     0     0     0     0       0   100%      0
       0     1     9     0     0       0    99%      0
     101     2     2    10     3      43    99%      0
       0     0     0     0     0       0   100%      0
       0     0     0     0     0       0   100%      0
      13     1     0     1     0       9    99%      0
       6     0     0     1     0       2   100%      0
      12     0     0     0     0       7   100%      0
      99     3     0    19     2      50    99%      0
       0     0     0     0     0       0   100%      0
       0     0     0     0     0       0   100%      0
```

```
      37     0     0     3     1      24   100%      0
^C

Summary (measurement time: 00:00:33)

                              Totals    Ratio per sec
------------------------------------------------------
Commands      :                  801               24
ASSO I/Os     :                   15                0
DATA I/Os     :                   31                0
WORK I/Os     :                   97                2
TEMP I/Os     :                  126                3
PLOG I/Os     :                   13                0
Throwbacks    :                  135                4
Buffer Hit    :                   99%
Buffer flushes:                    0

%ADAMON-I-TERMINATED,   29-MAR-2016 11:54:04, elapsed time: 00:00:35
```

DISPLAY パラメータを指定しない場合は、デフォルトの DISPLAY=ACTIVITY と同じになります。

例 **2**：

```
> adamon db=34 rcmd
%ADAMON-I-STARTED,      29-MAR-2016 12:53:53, Version <version number>

Database 34, startup at 29-MAR-2016 12:53:32
ADANUC Version <version number>, PID 6488



Commands Record.  Repl.       I/Os per second      Throw   Buffer pool
per sec  per sec per sec  ASSO  DATA  WORK  PLOG   backs   Hit  Flushs
-------------------------------------------------------------------------
     248      60      14     0     0    48     0       0   100%      0
     258      57       7     1     0    44     0       0    99%      0
     229      60      14     0     0    43     0       0   100%      0
     263      58      14   110    18    53     0       0   100%      1
     274      60       7     0     0    40     0       0   100%      0
^C

Summary (measurement time: 00:00:15)

                              Totals    Ratio per sec
------------------------------------------------------
Commands      :                 3809              253
Recorded      :                  878               58
Replicated    :                  160               10
ASSO I/Os     :                  330               22
DATA I/Os     :                   53                3
```

```
WORK I/Os     :                  679                45
TEMP I/Os     :                    0                 0
PLOG I/Os     :                    0                 0
Throwbacks    :                    0                 0
Buffer Hit    :                  99%
Buffer flushes:                    1
    WP1 Limit:                     1


%ADAMON-I-TERMINATED,   29-MAR-2016 12:54:04, elapsed time: 00:00:11
```

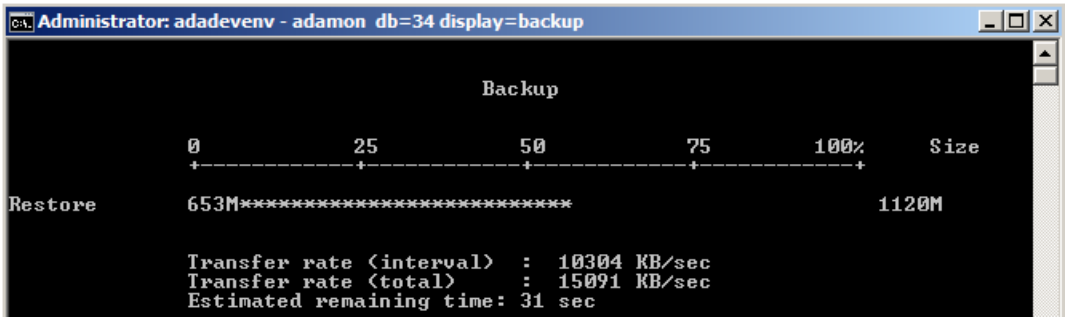
RCMD パラメータを指定した場合、DISPLAY=ACTIVITY にはさらに、「Rec. per second」（記録されたレプリケーションコマンド）および「Repl. per sec」（1 秒あたりのレプリケートされたコマンド数）の列が含まれます。

データベースに Adabas-to-Adabas レプリケーションが定義されている場合、サマリにはさらに、「Rec. per second」（記録されたレプリケーションコマンド）および「Repl. per sec」（1 秒あたりのレプリケートされたコマンド数）の値が含まれます。

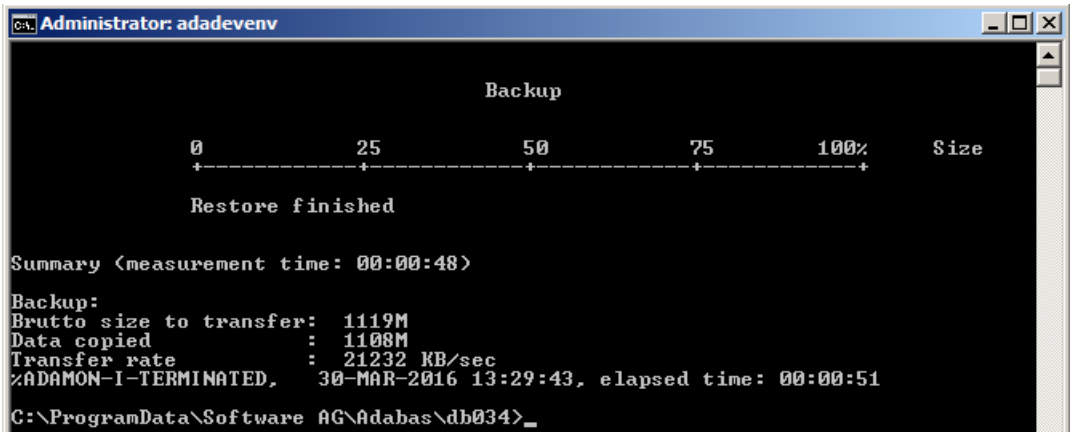
例 **3**：

```
C:\ProgramData\Software AG\Adabas\db034> adamon db=34 display=backup
```

このコマンドを実行すると、次の画面が3秒ごとに最新の値でリフレッシュされます。リフレッシュレートは INTERVAL パラメータで変更できます。



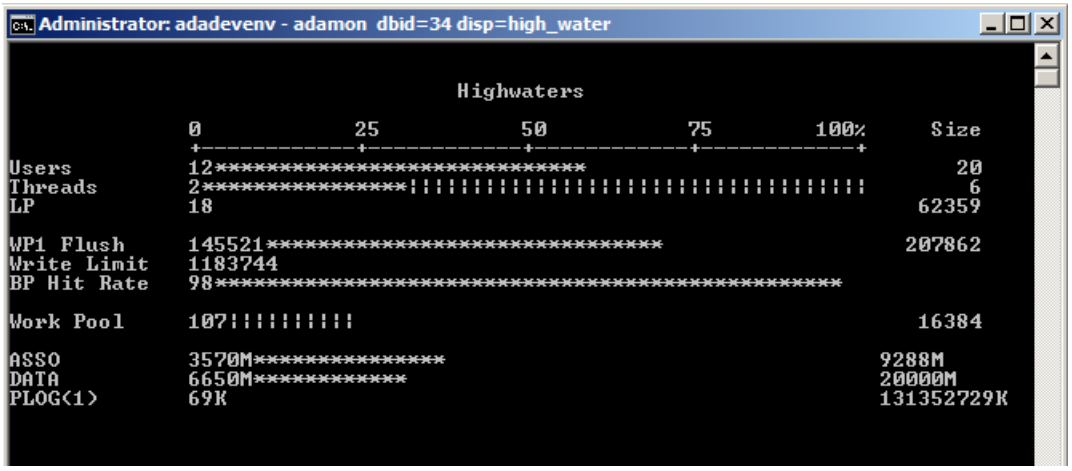
ダンプまたはリストアが完了した後で Control C キーを押すと、サマリが表示されます。



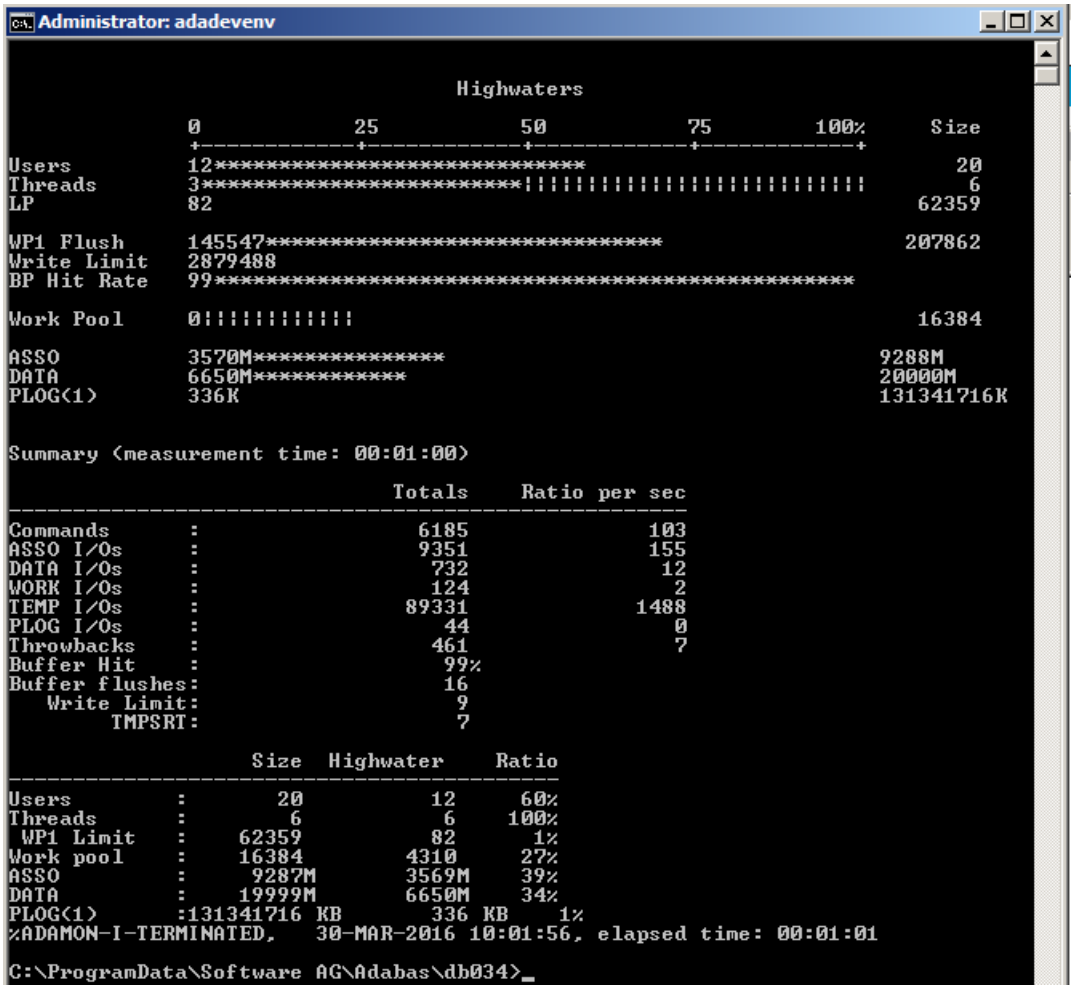
例 **4**：

```
C:\ProgramData\Software AG\Adabas\db034> adamon dbid=34 display=high_water
```

このコマンドを実行すると、次の画面が3秒ごとに最新の値でリフレッシュされます。リフレッシュレートは INTERVAL パラメータで変更できます。



Control C キーを押すと（Control C キーの代わりに、LOOPS パラメータで ADAMON を停止することもできます）、出力にサマリが追加されます。



例 5：

```
> adamon db=36 display=io
%ADAMON-I-STARTED,      29-MAR-2016 14:19:17, T-Version 6.5.0.0 (Solaris 64Bit)

Database 36, startup at 23-MAR-2016 16:58:41
ADANUC T-Version 6.5.0.0, PID 525



                          I/Os per sec
  A1    A2    A3    A4    A5    D1    D2    D3    D4    W1 PLOG     T
------------------------------------------------------------------------
   0     0     0     0     0     0     0     0     0     0     0     0
   0     0     0     0     0     0     0     0     0    10     1     0
   0     0     0     0     0     0     0     0     0   102     5     0
   0     0     0     0     0     0     0     0     0    63     6     0
   0     0     0     0     0     0     0     0     0   134     8     0
   0     0     0     0     0     0     0     0     0   127     8     0
   0     0     0     0     0     0     0     0     0    91     6     0
```

```
   0    0    0    0    0    0    0    0    0  105    7    0
   0    0    0    0    0    0    0    0    0  118    5    0
   0    0    0    0    0    0    0    0    0    0    0    0
   0    0    0    0    0    0    0    0    0    0    0    0
   0    0    0    0    0    0    0    0    0   64    5    0
   0    0    0    0    0    0    0    0    0   99    7    0
   0    0    1    0    0    0    0    0    0   88    6    0
   0    0    0    0    0    0    0    0    0   85    5    0
   0    0    0    0    0    0    0    0    0   81    7    0
   0    0    0    0    0    0    0    0    0  120    6    0
   0    0    0    0    0    0    0    0    0   79    5    0
^C

Summary (measurement time: 00:00:51)

                              Totals    Ratio per sec
-----------------------------------------------------
Commands      :                29597              580
ASSO I/Os     :                    1                0
DATA I/Os     :                    0                0
WORK I/Os     :                 4081               80
TEMP I/Os     :                    0                0
PLOG I/Os     :                  247                4
Throwbacks    :                 2913               57
Buffer Hit    :                   99%
Buffer flushes:                    0

ASSO1  I/Os  :        0               0
ASSO2  I/Os  :        0               0
ASSO3  I/Os  :        1               0
ASSO4  I/Os  :        0               0
ASSO5  I/Os  :        0               0
DATA1  I/Os  :        0               0
DATA2  I/Os  :        0               0
DATA3  I/Os  :        0               0
DATA4  I/Os  :        0               0
WORK1  I/Os  :     4081              80
TEMP   I/Os  :        0               0
%ADAMON-I-TERMINATED,   29-MAR-2016 14:20:08, elapsed time: 00:00:51
```

例 **6**：

この例の最初のステップでは、サマリ参照ファイルが作成されます。

```
>adamon db=34 summary=mon.sum interval=1 loops=100
%ADAMON-I-STARTED,      30-MAR-2016 10:47:48, Version <version number>

Database 34, startup at 30-MAR-2016 10:00:43
ADANUC Version <version number>, PID 2244



Summary file "mon.sum" created
%ADAMON-I-TERMINATED,   30-MAR-2016 10:49:30, elapsed time: 00:01:42
```

次に、サマリ参照ファイルの内容が表示されます。

```
>adamon db=34 summary_input=mon.sum interval=1 loops=100
%ADAMON-I-STARTED,      30-MAR-2016 11:29:41, Version <version number>

Database 34, startup at 30-MAR-2016 10:00:43
ADANUC Version <version number>, PID 2244



Summary (measurement time: 00:01:39)

                              Totals    Ratio per sec
-----------------------------------------------------
Commands      :                16053              162
ASSO I/Os     :                 2787               28
DATA I/Os     :                 1694               17
WORK I/Os     :                  420                4
TEMP I/Os     :                90434              913
PLOG I/Os     :                   54                0
Throwbacks    :                  355                3
Buffer Hit    :                    -
Buffer flushes:                   18

%ADAMON-I-TERMINATED,   30-MAR-2016 11:29:41, elapsed time: 00:00:00
```

最後のステップでは、サマリがデータベースの最新のアクティビティと比較されます。

```
>adamon db=34 summary_compare=mon.sum interval=1 loops=100
%ADAMON-I-STARTED,      30-MAR-2016 11:27:13, Version <version number>

Database 34, startup at 30-MAR-2016 10:00:43
ADANUC T-Version <version number>, PID 2244



Monitor Summary |        Reference         | Current
----------------+--------------------------+-------------------------
From date       : Wed Mar 30 10:59:18 2016 | Wed Mar 30 11:27:13 2016
To date         : Wed Mar 30 11:01:00 2016 | Wed Mar 30 11:28:55 2016
Duration in sec :                      99 |                       99
```

```
DBID            :                          34 |                          34
Version         :          <version number> |          <version number>
Structure level :                           1 |                           1
----------------+-------------------------+-------------------------

Monitor Summary |   Reference   |    Current    |   Absolute    |  Percentaged
----------------+---------------+---------------+---------------+-------------
Commands        :           16053 |           16326 |           +273 |          +2%
RPLCMDs         :               0 |               0 |             +0 |          +0%
Rec.            :               0 |               0 |             +0 |          +0%
Repl.           :               0 |               0 |             +0 |          +0%
ASSO I/Os       :            2787 |            5904 |          +3117 |        +112%
DATA I/Os       :            1694 |            4859 |          +3165 |        +187%
WORK I/Os       :             420 |             416 |             -4 |          -1%
TEMP I/Os       :           90434 |          145384 |         +54950 |         +61%
PLOG I/Os       :              54 |              82 |            +28 |         +52%
Throwbacks      :             355 |             706 |           +351 |         +99%
Buffer Hit      :              99 |              99 |             +0 |          +0%
Buffer flushs   :              18 |              27 |             +9 |         +50%

%ADAMON-I-TERMINATED,   30-MAR-2016 11:28:55, elapsed time: 00:01:42
```

参照ファイルを最新のデータベースアクティビティと比較する代わりに、2つ目のサマリファイルを作成して、2 つのファイルを比較することもできます。

```
>adamon db=34 summary=mon1.sum interval=1 loops=100
%ADAMON-I-STARTED,      30-MAR-2016 11:49:09, Version <version number>

Database 34, startup at 30-MAR-2016 10:00:43
ADANUC Version <version number>, PID 2244


Summary file "mon1.sum" created
%ADAMON-I-TERMINATED,   30-MAR-2016 11:50:51, elapsed time: 00:01:42

>adamon db=34 summary_compare_files=(mon.sum,mon1.sum)
%ADAMON-I-STARTED,      30-MAR-2016 12:02:19, Version <version number>

Monitor Summary |        Reference         | Current
----------------+--------------------------+-------------------------
From date       : Wed Mar 30 10:59:18 2016 | Wed Mar 30 11:49:10 2016
To date         : Wed Mar 30 11:01:00 2016 | Wed Mar 30 11:50:51 2016
Duration in sec :                       99 |                         99
DBID            :                       34 |                         34
Version         :          <version number> |          <version number>
Structure level :                           1 |                           1
----------------+-------------------------+-------------------------

Monitor Summary |   Reference   |    Current    |   Absolute    |  Percentaged
----------------+---------------+---------------+---------------+-------------
Commands        :           16053 |           14852 |          -1201 |          -7%
```

```
RPLCMDs          :              0 |              0 |             +0 |          +0%
Rec.             :              0 |              0 |             +0 |          +0%
Repl.            :              0 |              0 |             +0 |          +0%
ASSO I/Os        :           2787 |           5043 |          +2256 |         +81%
DATA I/Os        :           1694 |           6627 |          +4933 |        +291%
WORK I/Os        :            420 |            416 |             -4 |          -1%
TEMP I/Os        :          90434 |         174211 |         +83777 |         +93%
PLOG I/Os        :             54 |             57 |             +3 |          +6%
Throwbacks       :            355 |            554 |           +199 |         +56%
Buffer Hit       :             99 |             99 |             +0 |          +0%
Buffer flushs    :             18 |             31 |            +13 |         +72%

%ADAMON-I-TERMINATED,   30-MAR-2016 12:02:19, elapsed time: 00:00:00
```

20 ADAMUP（大量追加および削除）

この章では ADAMUP ユーティリティについて説明します。

機能概要

一括更新ユーティリティ ADAMUP は、データベースのファイルにレコードを追加またはファイルからのレコードの削除を行います。Adabas ニュークリアスがアクティブである必要はありません。

圧縮ユーティリティ ADACMP またはアンロードユーティリティ ADAULD によって生成された出力ファイルを、大量追加の入力として使用できます。

注意: ADAMUP ADD 機能は、以前の Adabas バージョンで作成された MUPDTA/MUPDVT ファイルを処理できますが、より新しい Adabas バージョンで作成された MUPDTA/MUPDVT ファイルは処理できません。

SINGLE_FILE オプションを指定した ADACMP または ADAULD によって生成された入力ファイルまたは LOG オプション指定の DELETE 機能を使用した ADAMUP の以前の実行によって生成された入力ファイルも使用できます。

処理対象のデータベースファイルにディスクリプタがなければ、ディスクリプタバリューテーブルなしに生成された入力ファイル（ADAULD の SHORT オプションまたは ADAMUP の LOG=SHORT オプション）が処理できます。

DELETE 機能に対する入力は、入力ファイルの指定の中に含まれます。各レコードには 1 つ以上の INS または ISN 範囲が含まれています。

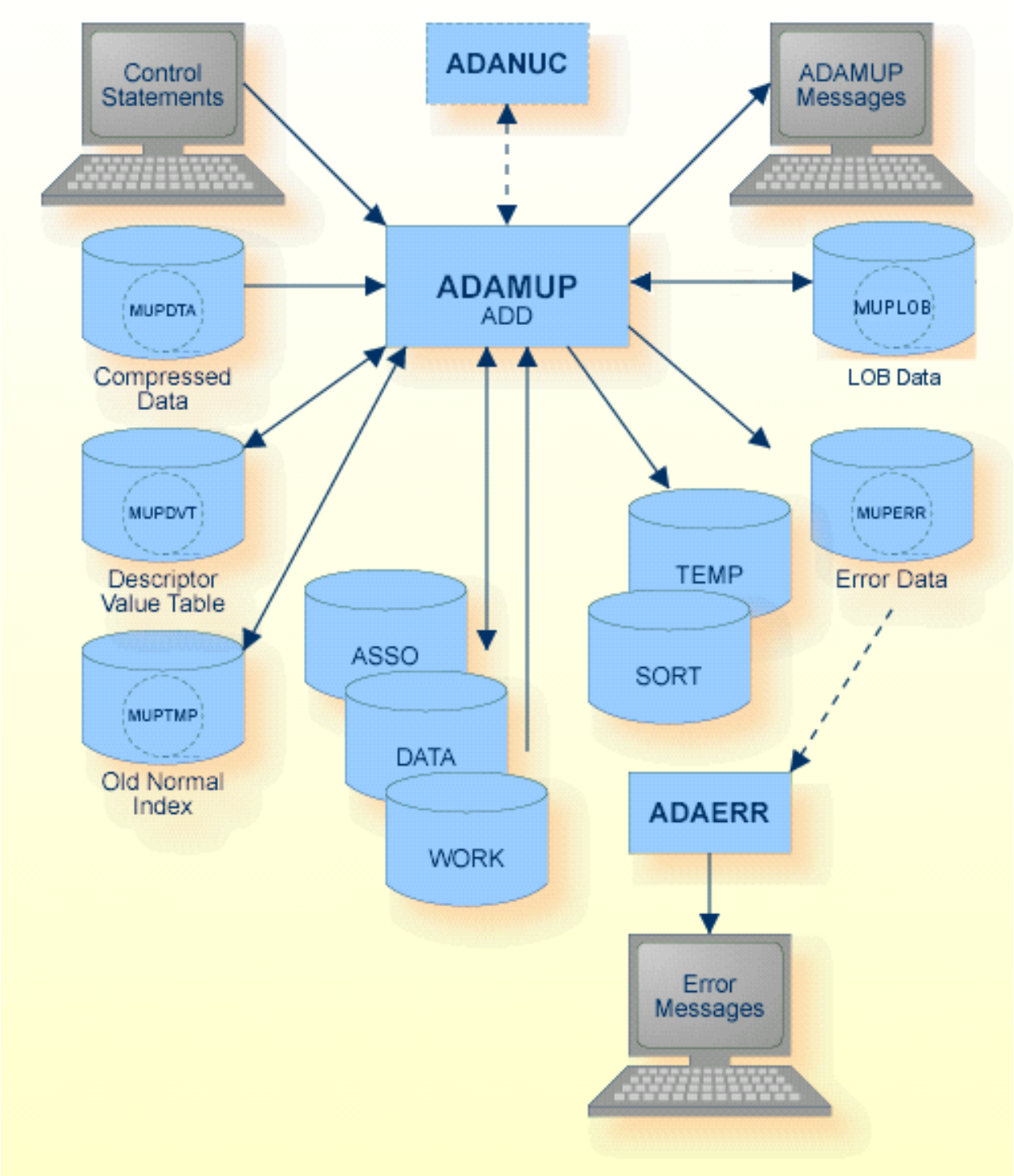
ADAMUP の 1 回の実行中に、データベースファイルへのレコードの追加とデータベースファイルからのレコードの削除の両方を行うことが可能です。

エラーファイルにレコードが書き込まれると、ユーティリティはゼロ以外のステータスで終了します。

注意: FDT が同じフィールドを含んでいても、照合ディスクリプタが異なる ICU バージョンに属している場合は、異なっていると見なされます。つまり、ICU バージョンが異なるファイルにデータをロードできるのは、ファイルが空で、かつ NEW_FDT パラメータを使用している場合だけです。

このユーティリティは単一機能ユーティリティです。

処理フロー

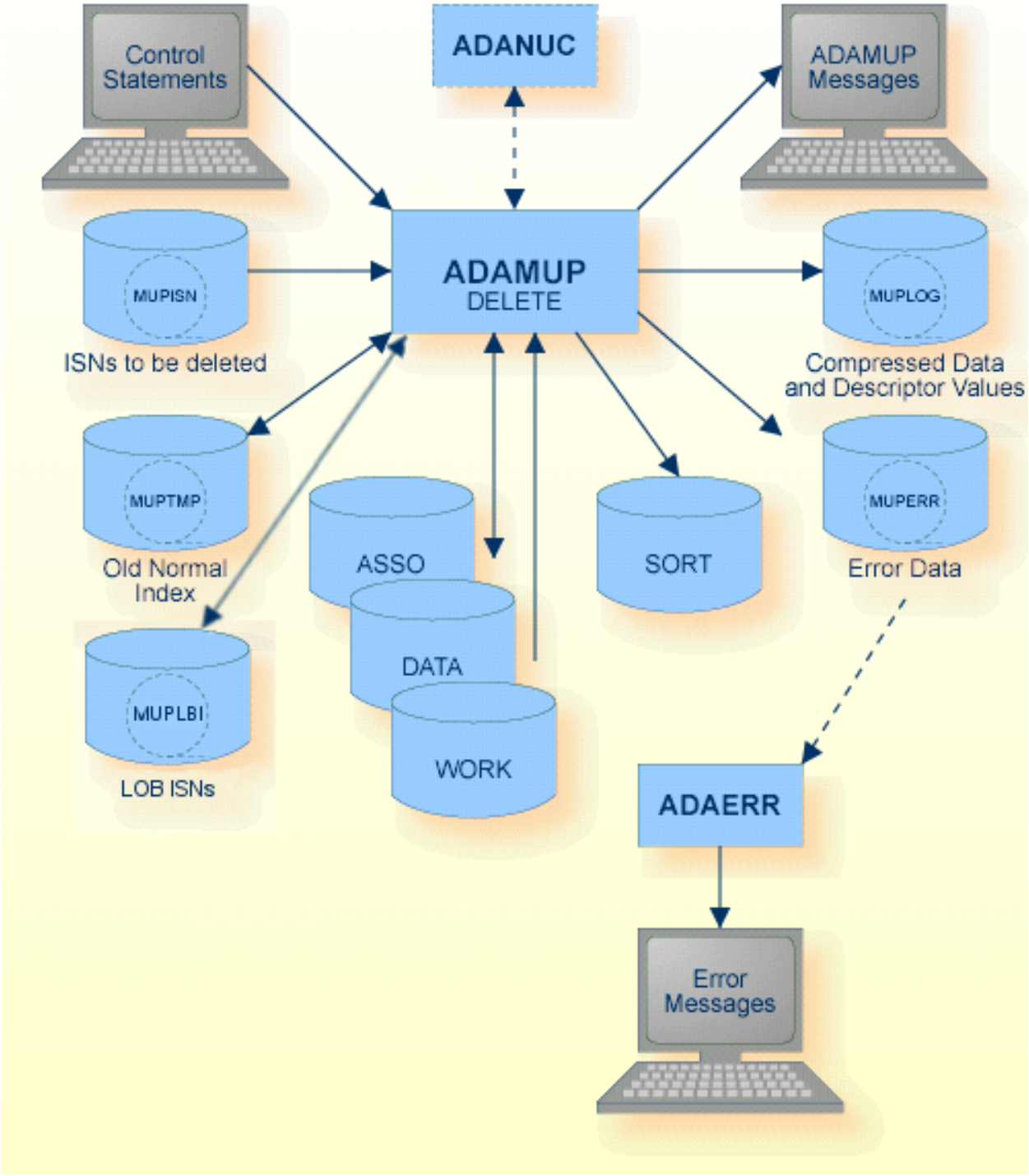

Control Statements
ADANUC
ADAMUP Messages
ADAMUP
ADD
MUPDTA
Compressed Data
MUPLOB
LOB Data
MUPDVT
Descriptor Value Table
MUPERR
Error Data
TEMP
SORT
ASSO
DATA
WORK
MUPTMP
Old Normal Index
ADAERR
Error Messages

シーケンシャルファイル MUPDTA、MUPDVT、MUPTMP、MUPLOB、および MUPERR は複数エクステントを持つことができます。複数のエクステントを持つシーケンシャルファイルの詳細については、『*Adabas Basics*』の「*ユーティリティの使用*」を参照してください。

データセット	環境 変数／ 論理名	記憶 媒体	追加情報
アソシエータ	ASSOx	ディスク	
データストレージ	DATAx	ディスク	
圧縮済み入力 データ	MUPDTA	テープ、ディスク	
ディスクリプタ値	MUPDVT	テープ、ディスク	
拒否データ	MUPERR	ディスク、テープ（* 注参照）	
LOB データ	MUPLOB	ディスク、テープ	一時ワークスペース（ADAMUP 終了時に再度削除されます）
ノーマルインデックス	MUPTMP	ディスク、テープ	一時ワークスペース（ADAMUP 終了時に再度削除されます）
ソートストレージ	SORTx TEMPLOCx	ディスク	Adabas Basics マニュアル、一時ワークスペース
コントロールステートメント	stdin/ SYS$INPUT		ユーティリティマニュアル
ADAMUP メッセージ	stdout/ SYS$OUTPUT		メッセージおよびコード
一時ストレージ	TEMPx	ディスク	
WORK	WORK1	ディスク	

注意: （*）このシーケンシャルファイルには、名前付きパイプを使用できます（OpenVMS にはない。詳細については、『*Adabas Basics*』の「*ユーティリティの使用*」を参照）。

アクティブなニュークリアスがなく、かつ保留になっている AUTORESTART もない場合は、環境変数／論理名 TEMP1 を WORK1 と同じ値に設定して、WORK を TEMP として使用できます。



シーケンシャルファイル MUPTMP、MUPLBI、MUPLOG および MUPERR は複数エクステントを持つことができます。複数のエクステントを持つシーケンシャルファイルの詳細については、『*Adabas Basics*』の「*ユーティリティの使用*」を参照してください。

データセット	環境 変数／ 論理名	記憶 媒体	追加情報
アソシエータ	ASSOx	ディスク	
データストレージ	DATAx	ディスク	
拒否データ	MUPERR	ディスク、テープ（* 注参照）	
削除する ISN	MUPISN	ディスク、テープ	
LOB ISN	MUPLBI	ディスク、テープ	一時ワークスペース（ADAMUP 終了時に再度削除されます）
圧縮データ	MUPLOG	ディスク、テープ	
ノーマルインデックス	MUPTMP	ディスク、テープ	一時ワークスペース（ADAMUP 終了時に再度削除されます）
ソートストレージ	SORTx TEMPLOCx	ディスク	Adabas Basics マニュアル、一時ワークスペース
コントロールステートメント	stdin/ SYS$INPUT		ユーティリティマニュアル
ADAMUP メッセージ	stdout/ SYS$OUTPUT		メッセージおよびコード
WORK	WORK1	ディスク	

注意: （*）このシーケンシャルファイルには、名前付きパイプを使用できます（OpenVMS にはない。詳細については、『*Adabas Basics*』の「*ユーティリティの使用*」を参照）。

チェックポイント

次の表は、各機能に対するニュークリアス条件と記録チェックポイントを示しています。

機能	ニュークリアスのアクティブ化が必要	ニュークリアスの非アクティブ化が必要	ニュークリアスは不要	記録チェックポイント
FDT			X	-
UPDATE	X（* 注 1 参照）	X（* 注 2 参照）	X（* 注 3 参照）	SYNP
SUMMARY			X	-

注意:

1. LOB データを含むファイルのレコードの削除時
2. Adabas システムファイルの更新時
3. LOB データを含むファイルのレコードの削除時以外

制御パラメータ

次のコントロールパラメータを使用できます。

```
M    DBID = number

     FDT

     SUMMARY

     UPDATE = number [,FDT]
              [ADD [,add_keywords]]
              [DELETE [,delete_keywords]]
 D            [,[NO]FORMAT]
 D            [LWP = number[K|M]]
```

DBID

```
DBID = number
```

このパラメータは、使用対象となるデータベースを選択するためのものです。

FDT

```
FDT
```

このパラメータは、データベースの選択されたファイルのフィールド定義テーブル（FDT）を表示します。ファイルにレコードを追加する場合、これらのレコードを含むシーケンシャル入力ファイルの FDT も表示することができます。このパラメータは ADD/DELETE 指定に使用することもできます。

FDT パラメータが使用される状況によって、シーケンシャル入力ファイル MUPDTA に含まれていたフィールド定義テーブルと、選択されたデータベースファイルに含まれていたフィールド定義テーブルのいずれか、または両方が表示されます。

例

```
adamup db=2 fdt update=11 add ↩
```

MUPDTA ファイルに保存された FDT が表示されます。

```
adamup db=2 update=11,fdt add ↩
```

データベース 2 のファイル 11 の FDT が表示されます。

```
adamup db=2 fdt update=11,fdt add ↩
```

MUPDTA ファイルに保存された FDT が最初に表示され、次にデータベース 2 のファイル 11 の FDT が表示されます。

SUMMARY

```
SUMMARY
```

このパラメータは、圧縮レコードが存在するシーケンシャル入力ファイルのディスクリプタスペースサマリ（DSS）を表示するものです。この表示は、この入力ファイルを生成したADACMP、ADAULD または ADAMUP の実行の最後のものと同一であり、インデックスに必要なスペースの見積もりに使用できます。

さらに、次の情報が表示されます。

- 必要な SORT サイズ（デフォルトの LWP に対する）
- TEMP の推奨サイズ（1 回のパスでインデックス更新を行うために必要なサイズ）
- SORT の現在のサイズ（存在する場合）
- メモリ内のソートに必要な LWP
- LWP および SORT の推奨サイズ（小規模な SORT サイズを使用できるほど LWP が十分な大きさの場合）

注意: デフォルトの LWP がメモリ内でソートを行うために十分な大きさの場合、SORT サイズは表示されません。

ADAMUP SUMMARY 処理の詳細については、『*管理マニュアル*』の「*ADAMUP および ADAINV 処理の最適化*」を参照してください。

UPDATE

```
UPDATE  = number [,FDT]
              [ADD [,add_keywords]]
              [DELETE [,delete_keywords]]
              [,[NO]FORMAT]
              [LWP = number[K|M]]
```

このパラメータは、レコードを追加／削除するファイルを指定するものです。ADAMUP にはファイルの排他制御が必要なので、ニュークリアスがアクティブの間は、Adabas システムファイルに ADAMUP を使用できません。LOB ファイルを指定することはできません。

ADD

```
ADD
   [,DE_MATCH = keyword]
   [,FDT]
   [,[NO]NEW_FDT]
   [,NUMREC = number]
   [,SKIPREC = number]
   [,UQ_CONFLICT = keyword]
   [,RI_CONFLICT = keyword]
   [,[NO]USERISN]
```

このパラメータは、UPDATE パラメータによって指定されるファイルにレコードを追加することを指示します。

大量追加に対する入力は、圧縮ユーティリティ ADACMP、アンロードユーティリティ ADAULD によって、または LOG パラメータをセットした DELETE パラメータを使用する一括更新ユーティリティ ADAMUP の以前の実行によって生成されます。

ADAMUP は、圧縮レコードが存在するシーケンシャル入力ファイルの FDT を指定データベースファイルの FDT と比較します。FDT は、同一レイアウトの必要があり、同じフィールド名、形式、長さおよびパラメータを使用しなければなりません。

データベースファイル内のディスクリプタは、シーケンシャル入力ファイルにおける FDT で定義されたディスクリプタのサブセットにすることができますが、入力ファイルは、データベースファイルに定義されたすべてのディスクリプタのためのディスクリプタバリューテーブル（DVT）のエントリを含んでいる必要があります。したがって、ディスクリプタバリューテーブルなし（SHORT オプション）で作成された入力ファイルは、現在のデータベースファイルにおいてディスクリプタ定義がない場合のみ更新処理が可能です。

一括更新の入力に LOB データが含まれる場合、Adabas ファイルには LOB ファイルが割り当てられている必要があります。

DE_MATCH = keyword

このパラメータは、入力データとともに提供されたディスクリプタがファイルの実際の FDT にあるディスクリプタではない場合の対応処置の指示に使用します。キーワードに IDENTICAL を指定した場合、ADAMUP は処理を終了してエラーメッセージを返します。キーワードに SUBSET を指定した場合、ADAMUP は入力ファイルにあるディスクリプタを無視し、そのディスクリプタはデータベースファイルから除去されます。

デフォルトは DE_MATCH=IDENTICAL です。

[NO]NEW_FDT

NEW_FDT が指定されると、ファイルの FDT は MUPDTA ファイルの FDT で置換されます。ADAMUP が開始するときにファイルが空であるときにだけ、NEW_FDT は指定することができます。

データベース内のファイルのFDTおよびMUPDTAファイルのFDTが異なる場合は、NEW_FDTを指定しなければなりません。FDT が異なっていて、ファイルが空でない場合は、一括更新は実行できません。

デフォルトは NONEW_FDT です。

NUMREC = number

このパラメータは、追加対象のレコード数を指定します。NUMRECを指定した場合、入力ファイルのエンドオブファイル条件によって ADAMUP 処理が終了しない限り、ADAMUP はあらかじめ NUMREC に定義されたレコード数の追加後に終了します。NUMREC および SKIPREC を指定しない場合、入力ファイルにある全レコードが追加されます。

SKIPREC = number

このパラメータは、レコードの追加を始める前にスキップする入力ファイルのレコード数を指定します。

UQ_CONFLICT = keyword

このパラメータは、ユニークディスクリプタに対して重複値が発見された場合の対応処置の指示に使用します。"keyword" の値は ABORT または RESET です。ABORT を指定した場合、重複するUQディスクリプタ値が検出されると、ADAMUPは処理を終了してエラーステータスを返します。RESET を指定した場合、矛盾する ISN はエラーログに書き込まれ、該当ディスクリプタの UQ ステータスが解除され、処理は続行します。

デフォルトは UQ_CONFLICT=ABORT です。

RI_CONFLICT

このパラメータは、参照整合性に違反した場合に実行するアクションを示すために使用します。"keyword" の値は ABORT または RESET です。ABORT を指定した場合、ADAMUP は終了し、エラーステータスが返されます。これらのインデックスはアクセス不可としてマークされます。RESET が指定されている場合、違反している制約は削除されます。どちらの場合も、違反した ISN はエラーログに保存されます。

デフォルトは RI_CONFLICT=ABORT です。

[NO]USERISN

このオプションは、各レコードに割り当てられる ISN として入力ファイルの ISN を使用するかどうかを指示します。

ユーザーは、データベースファイルに追加する各レコードに対する ISN 割り当てを制御したい場合は、このオプションをUSERISNに設定する必要があります。指定する各ISNは次の条件を満たさなければなりません。

- 各データレコードの直前に付く 4 バイトの 2 進数であること。

- ファイルに対する現在の制限（MAXISN）の範囲に入っていること。ファイルのアドレスコンバータは自動的に拡張されません。
- 指定ファイル内でユニークであること

上記の条件に当てはまらないと、ADAMUP は処理を終了してエラーメッセージを返します。

マルチプルバリューフィールドであるディスクリプタに基づいてアンロードによって作成された入力ファイルに対して、このパラメータを USERISN に設定すると、問題が発生する可能性があるので注意してください。その理由は、同じレコードが 2 回以上アンロードされていることがあるためです。詳細については、ADAULD ユーティリティの説明の中にある、SORTSEQ パラメータに関する説明を参照してください。

このオプションを NOUSERISN に設定する場合、各レコードの ISN は ADAMUP によって割り当てられます。ただし、ハイパー出口によって以前に再設定された DVT レコードの ISN は ADAMUP で変更されません。

デフォルトは NOUSERISN です。

DELETE

```
DELETE
      [,DATA_FORMAT = keyword]
      [,FDT]
      [,ISN_NOT_PRESENT = keyword]
      [,LOG = keyword | ,NOLOG]
```

このパラメータは、UPDATE パラメータによって指定されるファイルからレコードを削除するために使用します。削除対象レコードの ISN は、入力ファイル内に与えられます。

DATA_FORMAT = keyword

このパラメータは、入力ファイルの削除対象 ISN を指定するレコードのデータタイプを定義します。各レコードには 1 つ以上の INS または ISN 範囲が含まれています。

正しい ISN は 1 から MAXISN までの範囲です。

サポートされるフォーマットに従って、次の "keyword" の値を指定できます。

キーワード	説明
BINARY	単一の ISN は 4 バイトのバイナリ値に含まれます。ISN 範囲は 2 つの連続するバイナリ値に含まれ、2 番目の値の最上位ビットが設定されます。 このファイル内のブロックは 2 バイトの長さフィールド（長さフィールド自体の長さを除く）から始まります。 **注意:** 2**31（2147483648）以上の ISN は、DATA_FORMAT=BINARY で削除することはできません。

キーワード	説明
DECIMAL	各レコードは次のレイアウトを持ちます。 `[number[-number] [,number[-number]]...] [;comment]` "number" は 1 から 10 桁までの 10 進数です。

ADAMUP は、すべての入力レコードを第 1 ステップで検証します。ADAMUP は、検出されたエラーごとに行番号とオフセットを表示します。エラーが検出されるか、または入力ファイルがすべて解析されると、ADAMUP は実行を終了します。

デフォルトは DATA_FORMAT=BINARY です。

ISN_NOT_PRESENT = keyword

このパラメータは、入力ファイル内で指定された削除対象レコードの ISN が次のような場合の対応処置を指示します。

- 現在の制限（MAXISN）内にない場合
- ファイルのアドレスコンバータ内にない場合

"keyword" に使用できる値は、次のとおりです。

キーワード	説明
ABORT	矛盾する ISN が検出されると、ADAMUP は実行を強制終了してエラーメッセージを返します。
IGNORE	ADAMUP は、矛盾する ISN をエラーログに書き込み、処理を続行します。

デフォルトは ISN_NOT_PRESENT=IGNORE です。

LOG = keyword
NOLOG

LOG=keyword は、削除レコードをシーケンシャルファイルに記録することを指示します。レコードは圧縮形式で書き込まれ、圧縮ユーティリティ ADACMP およびアンロードユーティリティ ADAULD によって生成されるものと同じです。各データレコードの先頭にはその ISN が付くので、このファイルの再ロード時や大量追加時にこれらの ISN をユーザー ISN として使用できます（前述の USERISN パラメータを参照）。

"keyword" に使用できる値は、次のとおりです。

キーワード	説明
FULL	インデックスの構築に必要なディスクリプタ値を出力ファイルに含める。
SHORT	インデックスの構築に必要なディスクリプタ値を出力ファイルに含めない。

ADAMUPは、生成された圧縮データレコードとディスクリプタ値の両方を1つのファイルに書き込みます。

デフォルトは NOLOG です。

[NO]FORMAT

このパラメータは、（修正が行われた後）新インデックスが旧インデックスよりも少ないスペースしか必要としないときに、ファイルのノーマルインデックス（NI）のエクステントとアッパーインデックス（UI）のエクステントの最後にあるブロックの、フォーマット化を行うことができます。必要スペースが少なくなるというのは、インデックス内の削除、失われたインデックスブロックの修復またはパディングファクタの再確保の結果です。

これらのブロックはファイルの未使用ブロックに戻されるので、これらのブロックに格納されたデータが削除されていない場合は影響がありません。このパラメータをFORMATに設定されると、ADAMUP はこれらのブロックをバイナリのゼロで上書きします。

デフォルトは NOFORMAT です。

LWP = number[K|M]

ディスクリプタ値のソートのために、ADAMUPはメモリ内のワークプールを使用します。多くの場合、ワークプールのデフォルトサイズで、ADAMUPに最適なパフォーマンスがもたらされます。LWPパラメータにより、ワークプールを増やすことができます。デフォルトのワークプールサイズに追加されるスペースをバイト、キロバイト（K）、またはメガバイト（M）単位で定義します。

ワークプールサイズを大きくすると、次の場合に役立ちます。

- お使いの環境で、大規模なワークプールを使用するとパフォーマンスが向上する場合。
- SORT コンテナがディスクリプタ値をソートするには小さすぎる場合。適切な LWP パラメータにより、SORT コンテナの必要なサイズを縮小できます。

このパラメータに必要な値は、SUMMARY 機能を使用して特定できます。

再スタートに関する考慮事項

ADAMUP は再スタート機能を備えていません。異常終了した ADAMUP は、最初から再実行しなければなりません。

データストレージスペースを使い果たした場合、ADAMUP は異常終了せずに、ロード済みのレコードのインデックスの構築を試みます。つまり、そのファイルの内容は変わらないので、追加のデータストレージスペースを割り当てれば、SKIPREC オプションを使って残りのレコードをロードできます。

SORT データセットの格納先

SORT データセットは、アソシエータ、およびディスクリプタバリューテーブルが存在する入力ファイルと同じボリュームに置かないことをお勧めします。

SORT データセットは、少量のデータのみの追加時は省略できます。このとき、ADAMUP は、インコアソートを実行します。

必要な SORT および LWP サイズについての情報を得るには、SUMMARY 機能を使用します。

TEMP データセットの格納先

TEMP データセットは、ディスクリプタバリューテーブルが存在する入力ファイルおよび SORT と同じボリュームに置かないことがお勧めします。TEMP のサイズはノーマル／メインインデックスのロード時のパフォーマンスと密接な関係がありますが、ロードが正常に終了するどうかは、TEMP のサイズや有無とは関係がありません。

TEMP の推奨サイズを表示するには、SUMMARY 機能を使用します。

例

例 **1**：

```
adamup: dbid=1
adamup: update=10
adamup: add, userisn
```

データベース 1 のファイル 10 に、新規レコードを追加します。各入力レコードに付けられた ISN が使用されます。

例 **2**：

```
adamup: dbid=1
adamup: update=10
adamup: delete
```

入力ファイルに指定される ISN によって識別されるレコードが、データベース 1 のファイル 10 から削除されます。削除対象 ISN はバイナリ形式です。

例 **3**：

```
adamup: dbid=1
adamup: update=10
adamup: add, skiprec=1000
adamup: delete, data_format=decimal, log=full
```

データベース 1 のファイル 10 を対象に、新規レコードの追加と、古いレコードの削除を行います。入力ファイルにある最初の 1000 レコードは追加されません。追加される各レコードの ISN は、ADAMUP によって割り当てられます。削除対象レコードの ISN は、入力ファイルに 10 進数形式で格納されます。削除されたすべてのレコードは、出力ファイルに記録されます。再ロードする場合、インバーテッドリストの再作成に必要な値はログに記録されています。

21 ADANUC（データベースの起動とニュークリアスパラメータの定義）

この章ではADANUCユーティリティについて説明します。ADANUCは、データベースニュークリアスタスクです。

機能概要

ニュークリアスパラメータは、Adabas ニュークリアス実行環境を定義するのに使用します。

ニュークリアスパラメータは、ニュークリアスの起動時に設定されます。

ニュークリアスパラメータには、次の情報を指定します。

- 使用されるデータベース
- 最大Adabasバッファサイズおよびトランザクションとユーザーの非アクティブ状態の時間制限など、各種 Adabas セッションパラメータの設定
- Adabas セッション時に記録するコマンドデータのタイプおよび数これらのパラメータはあくまでも統計的な情報の指定に使用するものであって、Adabas データプロテクションログにデータベース更新履歴を記録するのに使用するものではありません。

このユーティリティは単一機能ユーティリティです。

注

1. ニュークリアス起動中に ADANUC が STP055 や STP997 などのストップエラーで終了した場合、指定したパラメータでニュークリアスを起動するために必要なオペレーティングシステムのリソース（メモリなど）がおそらく十分でなかったためです。いくつかのニュークリアスパラメータ（NT や LBP など）の値を減らすことで、このストップエラーを回避することができます。
2. データベースの作成後またはリストア後、最初のニュークリアスセッションの開始時に、Adabasニュークリアスは WORK コンテナのすべてのブロックを初期化します。WORK コンテナが大きい場合は、数分かかることがあります。WORK コンテナの初期化の終了後、データベースは使用可能になります。
3. 比較的小さい Adabas バッファで Adabas コールを実行するユーザーの数が 20 以内である場合は、ニュークリアスパラメータのデフォルト値を使用できます。その他のニュークリアスパラメータ値は、どのような状況で使用する必要あるかについて、次に説明します。
   - Adabas コマンドの応答時間が一時的に長くなる場合（Adabas バッファフラッシュ中）、またはハードウェアには問題がないにもかかわらず非同期 I/O 中に I/O エラーが発生する場合は、BFIO_PARALLEL_LIMIT を設定することを考慮してください。
   - LOB処理の場合など、大きいAdabasバッファでAdabasコマンドを実行している場合は、LAB と LBP の値を大きくすることを考慮してください。
   - マルチスレッドアプリケーションが Adabas コールを実行している場合は、NCL パラメータの値を大きくすることを考慮してください。

- 並行 Adabas セッションの数が 20 より大きい場合、NU パラメータの値を増やします。
- ニュークリアスのクラッシュ後に自動再スタートの時間を短縮する必要がある場合、大きいバッファプールを使用しているのであれば、WRITE_LIMIT を小さい正の値に設定します。

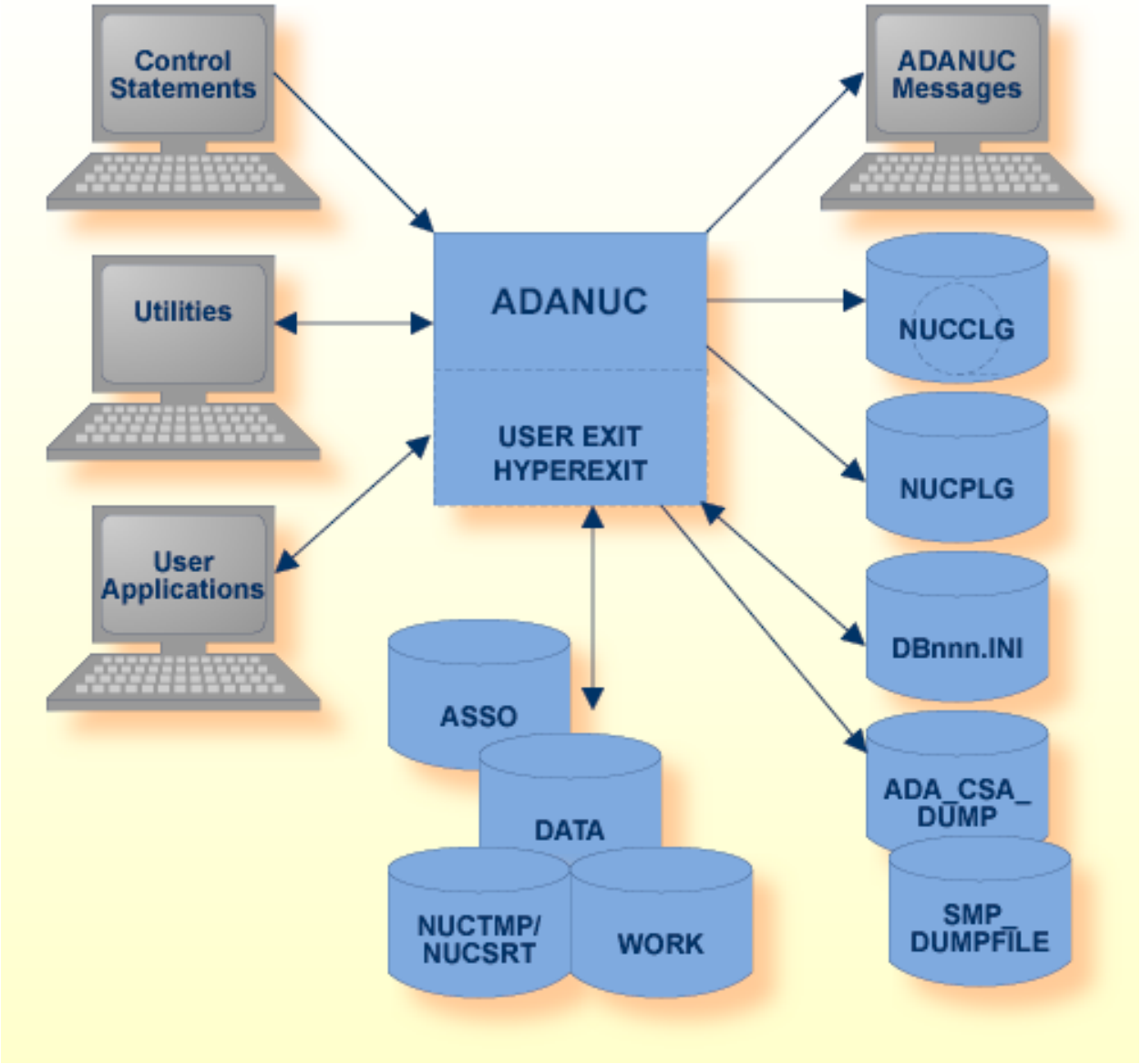
4. Adabas コンテナの書き込み権限を無効にするか、OPTIONS=READ_ONLY を指定すると、データベースを読み取り専用モードで実行できます。データベースを読み取り専用モードで実行する際に、一時ワークスペースの場所（環境変数 `TEMPLOCn`、または *DBnnn.INI* ファイルの `TEMPORARY_LOCATION` エントリ）を明示的に指定していると、ディスクに一時ワークスペースが作成されます。詳細については、『*Adabas Basics*』の「*一時ワークスペース*」を参照してください。
5. 前の Adabas セッションが SHUTDOWN または CANCEL によって正常に終了されなかった場合、Adabasは自動再スタートを実行します。ニュークリアスがクラッシュしたときにアクティブだったすべてのトランザクションがロールバックされ、不足しているすべてのデータベース更新が ASSO および DATA に書き込まれます。この目的のために、すべての更新処理が WORK コンテナに記録されています。また、すべての更新処理が NUCPLG ファイルに記録されます。これは、ディスク障害などにより 1 つ以上のデータベースコンテナが破損した場合に、データベースの最新の状態を回復するために必要です。ニュークリアスのクラッシュが発生した場合は、両方のログに同じ情報が含まれている必要があります。同じでないと、データベースに追加のトランザクションが含まれたり、後でリストア／リカバリを実行した場合にトランザクションが失われたりする可能性があります。これをチェックするには、自動再スタートが実行されるときに、PLOG ファイルが依然として同じ名前で使用可能になっている必要があります。PLOG ファイルの名前を変更していた場合、または別の場所に移動していた場合は、次の警告が表示されます。

```
%ADANUC-W-PLNOF, Last plog not found, so consistency check is not possible. New ↩
backup required.
```

この警告が表示された場合でも、データベースの整合性はまだ維持されていますが、後でリストア／リカバリを実行したときに、データベースの整合性が失われる可能性があります。リストア／リカバリのベースとして使用する新しいバックアップを作成すると、リストア／リカバリの整合性が再度保証されます。

6. UNIX プラットフォームでは、ADAOPR SHUTDOWN または ADAOPR CANCEL を使用してニュークリアスを正常にシャットダウンしていないと、Adabasニュークリアスによって割り当てられた IPC リソースが削除されません。これらのリソースが削除された後にのみ、ニュークリアスを再起動できます。そのため、ニュークリアスは `showipc -k <dbid>` コマンドを実行します。重要：推奨されるユーザー "sag"、グループ "sag" で Adabas をインストールしなかった場合は、環境変数 SIPUSER または SIPGROUP（あるいはその両方）を設定する必要があります。詳細については、『*管理マニュアル*』の「*showipc*」を参照してください。Windows プラットフォームでは、使用されなくなった IPC リソースがオペレーティングシステムによって自動的に削除されます。

処理フロー



シーケンシャルファイル NUCPLG および NUCCLG は複数エクステントを持つことができます。複数のエクステントを持つシーケンシャルファイルの詳細については、『*Adabas Basics*』の「*ユーティリティの使用*」を参照してください。

データセット	環境 変数／ 論理名	記憶媒体	追加情報
アソシエータ	ASSOx	ディスク	
コマンドログ	NUCCLG	ディスク、テープ	ユーティリティマニュアル、ADACLP
データストレージ	DATAx	ディスク	
DBnnn.INI		ディスク	Adabas 拡張オペレーションマニュアル
プロテクションログ	NUCPLG	ディスク	ユーティリティマニュアル、ADAPLP
コントロールステートメント	stdin/ SYS$INPUT		ユーティリティマニュアル
ADANUC メッセージ	stdout/ SYS$OUTPUT		メッセージおよびコード
WORK	WORK1	ディスク	
CSA ダンプ	ADA_CSA_DUMP	ディスク	Adabas CSA ダンプ（「ADAOPR RESPONSE_CHECK」を参照）
SMP ダンプ	SMP_DUMPFILE	ディスク	SMP ダンプ
一時ワークスペース（NUCTMPx、NUCSRTx）	TEMPLOCx	ディスク	Adabas Basics マニュアル（「一時ワークスペース」を参照）

注意:

1. 環境変数／論理名 ADA_CSA_DUMP では、Adabas CSA ダンプを作成するディレクトリを指定する必要があります。
2. Adabas CSA ダンプファイルのデフォルトディレクトリは、データベースディレクトリです。Adabas CSA ダンプのファイル名のレイアウトについては、ADAOPR、パラメータ CSA を参照してください。
3. Adabas CSA ダンプの詳細については、ADAOPR パラメータ ABORT および RESPONSE_CHECK も参照してください。
4. 環境変数／論理名 SMP_DUMPFILE には、（ディレクトリではなく）SMP_DUMPFILE 自体の名前を指定する必要があります。既存のファイルが上書きされます。
5. SMP_DUMPFILE が指定されていない場合、SMP ダンプは、SAGSMP.xxx.hh:mm:ss（UNIX）、SAGSMP.xxx.hh-mm-ss（Windows）、および SAGSMP-xxx-hh-mm-ss（OpenVMS）というファイル名で、Adabas CSA ダンプディレクトリに書き込まれます。ここで、xxx はデータベース ID で、hh:mm:ss/hh-mm-ss はニュークリアスの起動時刻です。

チェックポイント

次の表は、ニュークリアスが記録するチェックポイントを示しています。

機能	記録チェックポイント
ニュークリアス起動	SYNC
ニュークリアス終了	SYNC

制御パラメータ

次のコントロールパラメータを使用できます。

```
D    [NO]AB32BIT

D    ADABAS_ACCESS = {ALL | GROUP}

     APU = (n1,n2,n3)

D    AR_CONFLICT = keyword

     BFIO_PARALLEL_LIMIT = number

D    [NO]BI

D    CLOGBMAX = number [K | M]

D    CLOGLAYOUT =  [5 | 6]

M    DBID = number

D    LAB = number [K | M]

D    LABX = number [K | M]

D    LBP = number [K | M]

D    LOGGING = (keyword [,keyword]...)

D    LPXA = number

D    LWP = number [K | M]

D    NCL = number
```

```
D    NISNHQ = number

D    NT = number

D    NU = number

D    OPTIONS = (keyword[,keyword]...)

D    [NO]PLOG

D    READ_PARALLEL_LIMITS = (records,blocks,total)

D    TNAA = number

D    TNAE = number

D    TNAX = number

D    TT = number

D    UNBUFFERED = ALL | CLEAR | (keyword [, keyword [, keyword]])

     USEREXITS = (number[,number]...)

     WCHARSET = <ICU encoding>

D    WRITE_LIMIT = number
```

[NO]AB32BIT

注意: このパラメータは、HP-UX にのみ関係します。

HP-UXでは、32ビットアプリケーションから共有メモリにアクセスする場合に、32ビット共有メモリセグメントを作成する必要があります。そのため、32ビットアプリケーションでAdabasコールを実行する場合は、クライアントプログラムとADANUCの間での通信に必要なアタッチドバッファ用の共有メモリを、32ビット共有メモリとして作成する必要があります。ただし、32ビット共有メモリセグメントの合計サイズには制限があります。マシン上に複数のデータベースがある場合は、32 ビットの共有メモリ容量をそれ以上利用できないことが原因で、LABTB（LAB が大きすぎる）というメッセージが表示され、ADANUC を起動できないことがあります。32 ビットアプリケーションがない場合は、パラメータ NOAB32BIT を指定すれば、このエラーを回避できます。このようにすれば、アタッチドバッファ用の共有メモリは、64 ビット共有メモリとして作成されます。

デフォルトはAB32BITで、共有メモリは32ビット共有メモリセグメントとして作成されます。

注意: ニュークリアスを NOAB32BIT パラメータで起動し、Adabas コールを実行する 32 ビットアプリケーションを起動すると、アプリケーションは Adabas レスポンスコード 148（Adabas がアクティブではない）を受け取ります。

ADABAS_ACCESS

```
ADABAS_ACCESS = {ALL | GROUP}
```

ADABAS_ACCESS = ALL を指定する場合、すべてのユーザー が Adabas コールを実行できるようになります。

ADABAS_ACCESS = GROUP を指定している場合は、Adabas を起動するユーザーのグループ（例えば *sag*）に属しているユーザーのみが Adabas コールを実行できます。

デフォルトは、ALL です。

ADABAS_ACCESS パラメータの使用法の詳細については、『*管理マニュアル*』の「*Adabas のセキュリティ機能*」にある「*UNIX のグループ概念の使用*」を参照してください。

注意: このパラメータは、UNIX 環境でローカル Adabas コールを実行する場合にのみ使用できます。

APU

```
APU = (n1,n2,n3)
```

このパラメータは、Adabas セッション用に確立される Adabas 処理ユニットの数とレイアウトを指定します。

Adabas 処理ユニットは、特に NUMA アーキテクチャのコンテキストで、Adabas コマンドの並列実行を最適化します。Adabas 処理ユニットは、レシーバースレッドとワーカースレッドの論理的な集合と、専用のコマンドキューで構成されています。レシーバースレッドは、受信したユーザー要求を取得して、コマンドキューに入れます。一方、ワーカースレッドは、コマンドキュー内のコマンドを処理します。

各 Adabas 処理ユニットには、1 つのコマンドキューがあり、少なくとも 1 つのレシーバースレッドと 1 つのワーカースレッドが必要です。

注意: 制御パラメータ NT および APU の両方が指定されている場合、パラメータ NT の設定は無視されます。Adabasニュークリアスは、APUに定義された設定を使用してセットアップされます。

n1

Adabas ニュークリアス用に定義される Adabas 処理ユニットの数を指定します。最小値は 1、最大値は 8 です。

n2

Adabas処理ユニットごとのワーカースレッド数を指定します。最小値は1、最大値は100/n1です。ワーカースレッドの総数は 100 を超えないようにする必要があります。

n3

Adabas 処理ユニットごとのレシーバースレッド数を指定します。最小値は 1、最大値は 100/n1 です。

例

```
APU=(4,4,2)
```

これは、それぞれが4つのワーカースレッドと2つのレシーバースレッドを持つ、4つのAdabas処理ユニットを備えたAdabasニュークリアスを定義します。この値は、合計16個のワーカースレッドと8つのレシーバースレッドがあることを意味しています。

```
APU=(8,2,1)
```

これは、それぞれが2つのワーカースレッドと1つのレシーバースレッドを持つ、8つのAdabas処理ユニットを備えたAdabasニュークリアスを定義します。この値は、合計16個のワーカースレッドと8つのレシーバースレッドがあることを意味しています。

```
APU=(1,6,2)
```

これは、それぞれが6つのワーカースレッドと2つのレシーバースレッドを持つ、1つのAdabas処理ユニットを備えたAdabasニュークリアスを定義します。この値は、合計6個のワーカースレッドと2つのレシーバースレッドがあることを意味しています。Adabas処理ユニットを1つだけ定義すると、NTパラメータのみを使用して定義した実行時環境に類似した実行時環境になります。

AR_CONFLICT

```
AR_CONFLICT = keyword
```

このパラメータは、バッファフラッシュ時に最終システムが破壊され、それを再スタートによって検出された場合の対処方法を指定します。次のキーワードを使用できます。

キーワード	説明
ABORT	再起動しません。
CONTINUE	システムは再起動を試みます。

AR_CONFLICT=ABORTに指定しておくことをお勧めします。バッファフラッシュの割り込みでニュークリアスが起動しない場合だけ、一時的にAR_CONFLICT=CONTINUEを設定する必要があります。

注意: AR_CONFLICT=CONTINUEで再スタートした場合、不整合が発生する可能性があることに注意してください。この場合、ADAINV VERIFY機能で整合性のチェックを行ってください。

デフォルトはABORTです。

BFIO_PARALLEL_LIMIT

```
BFIO_PARALLEL_LIMIT = number
```

このパラメータは、バッファフラッシュによって並行に行われるI/O要求数の上限を指定するのに使用し、他のスレッドから同時に発生したI/Oのうち、処理の早いものを先に処理することを可能にします。例えば、大規模なバッファフラッシュが発生した場合、I/Oキューがビジーになり、他のI/O（バッファプール読み取りI/OやWORK I/Oなど）が長時間キューに入れられ、コマンドスループットが遅くなり、バッファフラッシュがアクティブな場合はアプリケーションが停止したまま動かなくなる可能性もあります。

BFIO_PARALLEL_LIMIT を指定した場合、バッファフラッシュは指定された数の I/O を設定し、次のパケットを発行する前にこれらのフラッシュの処理が完了するまで待機します。"number" に指定する最大値は、UNIX カーネルパラメータ AIO_LISTIO_MAX など、オペレーティングシステムによって決まります。値0を指定すると、許容されている最大値を指定したものとみなされます。デフォルト値は 50 です。

注意: BFIO_PARALLEL_LIMIT の値が高すぎる場合（または 0 の場合）、非同期 I/O の実行中に I/O エラー（ADRERR というユーティリティエラーメッセージ）が発生することがあります。この現象が発生するのは、オペレーティングシステム内で非同期I/Oに利用可能なメモリがなくなったためです。必要なメモリは、少なくともデータベースに書き込まれるブロックサイズに多少の追加スペースを加えた容量だと考えられます。この ADRERR エラーが発生しないようにするための BFIO_PARALLEL_LIMIT の最大値は、オペレーティングシステムの構成、および同一マシンでアクティブになっているその他のプロセスによって決まります。BFIO_PARALLEL_LIMIT の値としては 50 が適しているようですが、それでも ADRERR エラーが発生する場合は、オペレーティングシステム構成を確認してください。

例

```
adanuc: bfio_parallel_limit = 20
```

1 回のバッファフラッシュに対し、並行に行われる I/O 要求が 20 まで許可されます。

```
adanuc: bfio_parallel_limit = 0
```

並行バッファフラッシュで使用する I/O 数には制限がありません。

[NO]BI

```
[NO]BI
```

このパラメータは、ビフォーイメージをPLOGに書き込む（BI）か書き込まない（NOBI）かを指定するのに使用できます。

PLOG 上のビフォーイメージは、（正しい PLOG が使われているかなどの）データの整合性を検証するために再生成時に使用されます。NOBIを設定すると、PLOGは小さくなり、整合性の検証は行われません。

デフォルトは BI です。

NOBI を指定すると、ADAREC REGENERATE 機能を使用するときに、有効な整合性のチェック量が減少します。詳細については、「*ADAREC*」を参照してください。

CLOGBMAX

```
CLOGBMAX = number[K | M]
```

このパラメータは、コマンドログに記録されるAdabasバッファの最大長を指定します。Adabasバッファが指定値より大きくなると、バッファは切り捨てられてコマンドログに記録されます。0 を指定した場合は、バッファ全体が常に記録されます。

デフォルト値は 0 です（バッファ全体がログされる）。

例

```
adanuc: clogbmax = 4k
```

Adabas バッファのロギングは、それぞれのバッファの最初の 4 キロバイトに制限されます。

CLOGLAYOUT

```
CLOGLAYOUT = [5|6]
```

CLOGLAYOUT=5を指定すると、コマンドロギングを有効にしたとき、コマンドログはAdabasバージョン 5 と同じレイアウトになります。このレイアウトでは ACBX インターフェイスがサポートされないので、ACBX コールの場合には、従来の ACB インタフェイスでも使用可能な情報のサブセットのみが記録されます。コマンドログは、Adabasの旧バージョンにすでに存在していた ADACLP ユーティリティで評価できます。

CLOGLAYOUT=6を指定した場合、Adabasバージョン6でサポートされる新しいレイアウトでコマンドログが生成されます。コマンドログを評価するプログラム例 `prilogc` が提供されています。詳細については「*付録 B*」を参照してください。

デフォルト値は 5 です。

DBID

```
DBID = number
```

このパラメータは必須であり、使用対象データベースを選択するものです。"number" には 1 から 255 までの範囲内の数字を指定することができます。

例：

```
adanuc: dbid = 2
```

データベース 2 が使用されます。

LAB

```
LAB = number[K | M]
```

Adabas セッション中に使用されるアタッチドバッファエリアのサイズを指定します。アタッチドバッファエリアは、コマンド実行中、ユーザーバッファの割り当てに使用されます。

Adabas は、アタッチドバッファエリアに割り当てるため、指定された値を 32 キロバイトの倍数に切り上げます。

このパラメータの上限は、ADAOPR の DISPLAY パラメータを使用して表示できます。

最小値は 1 メガバイトで、デフォルト値は 1 メガバイトです。しかし、指定された値が NCL パラメータの値（キロバイト単位）より小さい場合、LAB の値は自動的にその値に拡張されます。

注意:

1. アタッチドバッファは、各ローカルクライアントスレッドについて、このスレッドで実行される Adabas コールのユーザーバッファの最大の全体サイズを含むように十分に大きくなければなりません。ユーザーバッファの管理にスペースが必要になったり、スレッドごとに別々のサイズが余計に必要になったりして、それよりもスペースが必要になることもあります。
2. 64 KB より大きいアタッチドバッファは、別のアタッチドバッファ拡張エリアに割り当てられます（パラメータ **LABX** を参照）。

LABX

```
LABX = number[K | M]
```

このパラメータは、LOB処理、または大型のAdabasバッファエリアを使用するその他のAdabasコールに使用するアタッチドバッファ拡張エリアのサイズを指定します。通常のアタッチドバッファは、ユーザーのAdabasセッション全体で割り当てられたまま残りますが、アタッチドバッファ拡張は、64 KB 以上の大型のアタッチドバッファスペースを必要とするそれぞれの Adabas コールの前に割り当てられ、コールの最後に再び解放されます。

アタッチドバッファ拡張エリアが導入されたのは、通常のアタッチドバッファエリアに大型バッファを保存する場合、フラグメントのために非常に大きいスペースが必要になるためです。

Adabas は、アタッチドバッファエリアに割り当てるため、指定された値を 32 キロバイトの倍数に切り上げます。

このパラメータの上限は、ADAOPR の DISPLAY パラメータを使用して表示できます。

最小値は 1 メガバイト、デフォルト値は 20 メガバイトです。

LBP

```
LBP = number[K | M]
```

このパラメータは、セッション時の Adabas バッファプールの大きさをバイト単位で指定します。

このパラメータの上限は、ADAOPR の DISPLAY パラメータを使用して表示できます。

最小値は 32 メガバイト、デフォルト値は 100 メガバイトです。

注意: バッファプールの実際のサイズは、NT パラメータに応じて異なります。LBP に指定されたサイズが NT * 2M 未満の場合、自動的に NT * 2M に増やされます。

Adabas でバッファプールを使用する理由は次のとおりです。

- 同じデータベースブロックが複数回アクセスされる場合、そのブロックを再び読み取るための物理 I/O を回避できます。
- データベースを更新するコマンドは、すべての対応するデータベースブロックがディスクに書き込まれる前に終了できます。

バッファプールが小さすぎる場合は、十分なバッファプールサイズで同じデータベース操作を実行するときよりも I/O が多くなります。場合によっては、Adabas が Adabas コマンドを正常に処理できないことがあります。この場合、コマンドにはレスポンスコード 162（バッファプールが小さすぎる）が返されます。

特にLOBを処理する場合は、バッファプールのサイズを大きくしてください。その他多くのデータベースブロックを削除せずに、バッファプールに LOB を配置できるようにする必要がありま

す。一方、バッファプール I/O がオペレーティングシステムのページング I/O で置き換わることに利点はないので、バッファプールを大きくしすぎないでください。

バッファプールの最適なサイズを探すには、ADAOPR パラメータ DISPLAY=BP_STATISTICS を使用できます。

例：

```
adanuc: lbp=32m
```

Adabas バッファプールに対して 32 メガバイトが割り当てられます。

LOGGING

```
LOGGING = (keyword [,keyword]...)
```

このパラメータは、キーワードリスト内で定義として、バッファのロギングを指定します。

キーワードリストに指定できるキーワードは、次のとおりです。

キーワード	説明
CB	Adabas コントロールブロックを記録します。
FB	フォーマットバッファログを記録します。
RB	レコードバッファを記録します。
SB	サーチバッファを記録します。
VB	バリューバッファを記録します。
IB	ISN バッファを記録します。
ABD	Adabas バッファ記述を記録します。
IO	入出力状況を記録します。
OFF	ロギングは実行されません。

オペレータコマンドを使用して、セッション実行中にロギングパラメータを指定することもできます。詳細については、「ADAOPR」のセクションを参照してください。

例

```
adanuc: logging=(cb)
```

現在の Adabas セッションに対してコマンドロギングが実行され、全 Adabas コントロールブロックが記録されます。

LPXA

```
LPXA = number
```

このパラメータは、OPTIONS=XA が指定されている場合に、XA 環境の Adabas WORK のデータプロテクション情報用に予約されたブロック数を指定します。これらのブロックを使用して、XA 環境で実行しているトランザクションのうち、ニュークリアスのシャットダウン時に保留状態になっているか、または長時間保留状態（準備されているがまだコミットしていない）のもののプロテクション情報を保存します。それぞれこのようなトランザクションは、これらの最低1ブロックを使用します。このエリアに空きブロックが存在しない場合、Adabas はこのようなトランザクションをヒューリスティックにコミットします。

最小値は 1 で、デフォルト値は 10 です。

LWP

```
LWP = number[K | M]
```

このパラメータは、Adabas ワークプールのサイズを指定します。Adabas ワークプールは、Adabas ニュークリアスセッション用に使用される、メモリ内のワークエリアです。

Adabas ワークプールは、次の情報を格納するために使用されます。

- ディスクリプタバリューテーブル（DVT）
- コマンド実行時の WORK I/O エリア

このパラメータの上限は、ADAOPR の DISPLAY パラメータを使用して表示できます。

デフォルト値は 16 M、最小値は 500 K、最大値は 4000 M です。

例：

```
adanuc: lwp=750k
```

Adabas ワークプールの大きさは、750 K バイトです。

NCL

```
NCL = number
```

このパラメータは、データベースにローカルにアクセスしているクライアントスレッドの数を指定します。通常、クライアントスレッドの数は Adabas ユーザーセッションの数（= ユーザーキューエレメントの数）と同じです。これは NU パラメータによって制限されます。ただし、マルチスレッドアプリケーションの場合、同じ Adabas セッションに複数のスレッドがアクセスできるので、クライアントスレッドの数が多くなる可能性があります。一方、同じプロセスが異なるユーザー用に異なる Adabas ユーザーセッションを使用するアプリケーションサーバーの場

合、クライアントスレッドの数がAdabasユーザーセッションの数よりも少なくなることがあります。

最小値は 2 です。デフォルト値は NU パラメータの値になりますが、最低でも 50 になります。このパラメータでは最大値が決まっておらず、オペレーティングシステムで使用できる IPC リソースのみによって制限されます（詳細については、「*System V リソースの増加（AIX 以外）*」を参照してください）。

Net-Work バージョン 7 は、データベースにアクセスするために、いくつかのスレッドを使用することに注意してください。詳細については、Net-Work のドキュメントを参照してください。

注意:

1. NCL パラメータに指定する値を高くしすぎないでください。Adabas セッションがクローズコマンドで正常に終了したとき、そのコマンドに使用されたメッセージキューは削除されますが、セッションがクローズコマンドなしで異常終了した場合、メッセージキューは削除されません。メッセージキューのクリーンアップは、クライアントメッセージキューの数がNCL パラメータの指定値と等しいときに限って実行されます。このため、アクティブな Adabas ユーザーの実際の数が非常に少なくても、NCL パラメータの値が高すぎると、システムの許容最大数よりも多いメッセージキューを Adabas が作成しようとしてしまいます。このような場合、Adabas コマンドには応答 255 が返されます（詳細については『*メッセージおよびコード*』を参照）。
2. メッセージキューのクリーンアップは、ADAOPR FREE_CLQ コマンドを使用して手動で開始できます。

NISNHQ

```
NISNHQ = number
```

このパラメータは、単一のユーザーによっていつでもホールド状態にできる最大レコード数を指定します。

ユーザーが認められた件数を超えてレコードをホールド状態にしようとすると、ホールドキューにスペースがあっても、ゼロ以外のレスポンスコードが返されます。

このパラメータの上限は、ADAOPR の DISPLAY パラメータを使用して表示できます。

最小値は 0（0 は無制限であることを意味します）で、デフォルト値は 0 です。

例

```
adanuc: nisnhq=50
```

単一ユーザーがホールド状態にできる最大レコード件数は 50 件です。

NT

```
NT = number
```

このパラメータは、Adabas セッションに対して確立できるスレッドの数を指定します。

Adabas コマンドごとに 1 スレッドへの割り当てが行われます。スレッドは該当コマンドの処理が完了すると解放されます。

このパラメータの上限は、ADAOPR の DISPLAY パラメータを使用して表示できます。

最小値は 1、最大値は 100 で、デフォルト値は 6 です。

注意:

1. NT > 1 の場合にのみ実行できる内部コマンドがあるので、1 より大きい NT パラメータ値を使用することを強くお勧めします。NT が 1 の場合、これらの内部コマンドを使用する SQL Gateway などの一部のユーティリティやアプリケーションが失敗することがあります。
2. NT = 1 は一部の特殊な状況でのみ妥当です。例えば、サポートが ADANUC トレースを要求していて、すべての Adabas コマンドのトレースを 1 つのファイルに含める必要がある場合などです。
3. NTパラメータを大きくすると、通常、オーバーヘッドが発生します。これにより、パフォーマンスが低下する場合があります。そのため、ハードウェアスレッド数よりも大幅に大きいNTパラメータ値を使用することは、通常、適切ではありません。ハードウェアスレッドの数が多い場合は、NTがハードウェアスレッドの数より少ない場合でも、パフォーマンスが向上する場合があります。これは、マシンで実行され、CPU時間を使用するプログラムが、Adabasニュークリアスだけではないことが理由です。
4. NTパラメータの値を増やすことが必要な場面の1つとして、完了までに長い時間が必要な、複数の複雑なコマンドが並行して存在する場合が挙げられます。すべてのニュークリアススレッドがこのようなコマンドによってブロックされていると、実行時間の短いコマンドを長時間スケジュールできません。その結果、Adabasの全体的なパフォーマンスが大きく低下することがあります。
5. 以前のバージョンの Adabas では、実行時間の長い複雑なコマンドによってすべてのスレッドがブロックされて、全体的なパフォーマンスが大きく低下するので、実行時間の短いコマンドをスケジュールできなくなる可能性がありました。この状況を回避するために、このように長時間実行されるコマンドを並行して処理する場合は、NT/2 スレッドのみを使用できるようになりました。その結果、スケジュール待ちのコマンドキューに十分なコマンドがあっても、複数の空きスレッドが表示される場合があります。

6. 制御パラメータ NT および APU の両方が指定されている場合、パラメータ NT の設定は無視されます。Adabas ニュークリアスは、APU に定義された設定を使用してセットアップされます。

例：

```
adanuc: nt=8
```

8 スレッドがセッションに対して確立されます。

NU

```
NU = number
```

このパラメータは、Adabas セッションに対して確立されるユーザーキューエレメントの数を指定するものです。

アクティブな Adabas ユーザーごとに 1 つのユーザーキューエレメントが割り当てられます。ユーザーキューエレメントは、ユーザーが OP コマンドを発行するか、または最初の Adabas コマンドが発行されると割り当てられます。またユーザーキューエレメントは、ユーザーが CL コマンドを発行するか、タイムアウトによって削除された場合に解放されます。

このパラメータの上限は、ADAOPR の DISPLAY パラメータを使用して表示できます。

最小値は 2、デフォルト値は 20、最大値は 99999 です。

例：

```
adanuc: nu=100
```

Adabas ユーザーキューは 100 個のエレメントから構成されます。

注意: NU パラメータは 2 つの異なる目的のために使用されます。

- ユーザーキューエントリの数
- コミュニケーションブロックの数

シングルスレッドアプリケーションの場合、必要となるコミュニケーションブロックの数は、ユーザーキューエレメントの数より多くありません。しかし、マルチスレッドアプリケーションの場合、Adabas コールを実行するアプリケーションのスレッドごとに 1 つのコミュニケーションブロックが必要となります。したがって、必要となる NU パラメータの値は、Adabas ユーザーの数よりはるかに大きくなることもあります。

OPTIONS

```
OPTIONS = (keyword[,keyword]...)
```

このパラメータは、ニュークリアスの起動モードの定義に使用します。

次のキーワードが使用できます。

キーワード	説明
AUTO_EXPAND	AUTO_EXPANDを選択した場合、データベースは、既存コンテナエクステントを拡張します。または、既存のAdabasファイルを拡張したり、新規Adabasファイルを追加したりするための空きスペースがない場合は新規コンテナエクステントを作成します。詳細については、『*Adabas Basics*』の「*コンテナファイル*」セクションを参照してください。
AUTORESTART_ONLY	AUTORESTART_ONLYキーワードは、ニュークリアスの起動シーケンスが完了した直後にニュークリアスをシャットダウンします。自動再スタートが保留状態の場合、自動再スタートが実行されます。ユーザーコマンドやユーティリティコールを実行しようとしても、ニュークリアスに拒否されます。
FAULT_TOLERANT_AR	FAULT_TOLERANT_AR を選択すると、自動再スタートの実行中に Adabas エラーが発生した場合のニュークリアスの動作を制御できます。ファイルにエラーが発生した場合、詳細なエントリがニュークリアスログに作成されますが、自動再スタートは継続します。自動再スタートの完了時に、DBA は ADABCK RESTORE および ADAREC REGENERATE を関連するファイルに使用して、ファイルをリストアし、再生成することができます。ただし、ファイルのインデックスにエラーが発生したときには、ADAINV の REINVERT パラメータを使用してファイルのインデックスを再構築するだけで十分です。 FAULT_TOLERANT_AR を選択しないと、エラーが検出されたとき、自動再スタートは異常終了するため、データベースに不整合が発生する可能性があります。
LOCAL_UTILITIES	LOCAL_UTILITIESを選択すると、ADANUCはすべてのリモートユーティリティコールを排除します。つまり、Adabas ユーティリティはリモートノードからネットワーク経由で実行できないということです。オペレーティング環境でセキュリティが重要であれば、この設定をお勧めします。 LOCAL_UTILITIES および UTILITIES_ONLY は、データベースを終了させることなく、動的に有効／無効を切り替えできます（詳細については「*ADAOPR*」を参照）。
OPEN_REQUIRED	OPEN_REQUIREDを選択する場合、オープン（OP）コマンドをユーザーセッションの第 1 番目に発行する必要があります。 アプリケーションサーバーから Adabas を呼び出すとき、Entire Net-Work を使用するときに lnk_set_adabas_id を使用する場合、このオプションを指定する必要があります。また、このようなケースで、このオプションを指定しない場合、ADANUCを再起動した後のトランザクションの完全性が保証されなくなります。
READONLY	READONLY オプションを指定すると、ADANUC は読み取り専用モードで実行します。詳細については、『管理マニュアル』を参照してください。

キーワード	説明
TRUNCATION	TRUNCATIONキーワードは、英数字フィールドの桁落ちを制御します。このキーワードを設定すると、必要に応じて英数字フィールドの桁が落とされますが、返されるレスポンスコードは0になります。このキーワードを設定しない場合には、桁落ちが発生するとレスポンスコード55が返されます。
UTILITIES_ONLY	UTILITIES_ONLYを選択すると、データベースのファイルにDBA排他制御がかかり、ユーティリティ以外へのすべてのコールは拒否されます。ただし、この制限は新規ユーザーにのみ適用され、OPTIONS=UTILITIES_ONLYを指定したときにすでにアクティブであったユーザーは通常の処理を続行できます。ファイルまたはデータベース全体に対する排他的なユーティリティ制御が必要な場合は、ADAOPRのLOCK機能を使用してください。 LOCAL_UTILITIESおよびUTILITIES_ONLYは、データベースを終了させることなく、動的に有効／無効を切り替えできます（詳細についてはこのマニュアルの「*ADAOPR*」を参照）。
XA	キーワードXAは、サーバーがX/Open XA仕様に従って、分散トランザクション処理をサポートすることを示します。Adabas XAインターフェイスがアプリケーションによって使用される場合、OPTION=XAを使用する必要があります。詳細については、「*XA のサポート*」を参照してください。

デフォルトでは、パラメータの値は設定されていません。

例：

```
adanuc: options = utilities_only
```

ユーティリティ以外へのコールはすべて拒否され、DBA はデータベースファイルを排他的に制御できます。

[NO]PLOG

```
[NO]PLOG
```

PLOGは、どのプロテクションログを記録するかを指定します。

ディスクが物理的に損傷したときに、プロテクションログがなければ、そのデータベースは再生成できません。この場合、そのデータベースは最新のデータベースダンプを使用して修復しなければなりません。しかし、最新のダンプの後に作成された更新内容は、修復後にすべて消失します。

PLOGがデフォルトです。

READ_PARALLEL_LIMITS

```
READ_PARALLEL_LIMITS = (records,blocks,total)
```

シーケンシャルコマンドの場合、Adabas は次のレコードに必要なデータベースブロックを前もって並行して読み取り、パフォーマンスを上げようとします。したがって、ディスクのストライプ化のメリットが活き、I/O のシーケンスも最適化されます。物理 I/O レートが高い場合にのみ、大幅なパフォーマンスの向上が期待できます。READ_PARALLEL_LIMITS パラメータによってパフォーマンスを改善できる一般的なケースとして、大容量ファイル全体（プログラムの起動時にバッファプールにない多数のデータベースブロックが含まれています）を読み込むバッチプログラムの実行が挙げられます。しかし、このようにデータベースブロックをプリフェッチすると、パフォーマンスに悪影響を及ぼすこともあります。

- 後続のレコードに必要なブロックを決めながら処理を行うことになるため、その処理の分だけ余計に CPU 時間がかかることになります。しかし、すべてのブロックがバッファプールにすでに格納されているのであれば、このような処理を行う意味がありません。
- コマンドシーケンスの後続のレコードを読み取らないのであれば、このコマンドシーケンスのブロックの読み取りにかかるオーバーヘッドは本来、不要なものです。
- ブロックの I/O を初期化するときのブロック数が多すぎると、並行して実行されるコマンドが他にもある場合には、そのコマンドに必要な I/O が大幅に遅れることがあります。

READ_PARALLEL_LIMITS パラメータでは、先読み動作を制御できます。3 つの数値を指定することができますが、その意味は次のとおりです。

records
コマンドシーケンスの次から処理されるレコードの最大数。バッファプールにまだ入っていないブロックがあるかどうかチェックされます。指定できる最大値は 65,535 です。

blocks
1 つのコマンドシーケンスで先読みするブロックの最大数。ブロックに指定する数値は、"records"に指定する数値以下にする必要があります。最大値はオペレーティングシステムによって異なります。

total
データベース全体で同時に先読みするブロックの最大合計数。"total" に指定する数値は、"blocks" に指定する数値以上にする必要があります。最大値はオペレーティングシステムごとに異なります。

デフォルトは (0,0,0) であり、先読みは実行されません。

注意: `total` に指定した数値が高すぎる場合は、非同期 IO 中に I/O エラー（ADRERR というユーティリティエラーメッセージ）が発生することがあります。この現象が発生するのは、オペレーティングシステム内で非同期I/Oに利用可能なメモリがなくなったためです。必要なメモリは、少なくともデータベースに書き込まれるブロックサイズに多少の追加スペースを加えた容量だと考えられます。この ADRERR エラーが発生しない `total` の最大値は、オペレーティングシステムの構成、および同一マシンでアクティブになっているその他のプロセスによって決まります。

例

```
adanuc: read_parallel_limits = (100,20,50)
```

シーケンシャルコマンドを処理するとき、コマンドシーケンスの次の100レコードまでが、まだバッファプールに入っていないブロックが必要な分だけあるかどうかがチェックされます。読み取りの対象となるブロックの検索は、20 ブロックが見つかったとき、または見つかったブロックの数および別のコマンドで現在読み取られているブロックの数の合計が 50 になったときに停止します。次に、見つかったブロックを対象に読み取り I/O が始まります。

TNAA

```
TNAA = number
```

このパラメータは、アクセスオンリーユーザーがAdabasコマンドを発行しないでアクティブでいられる最大経過時間（秒単位で）を指定するものです。この値は、ADAOPRユーティリティによってダイナミックに変更されることがあります。

この値は、OP コマンドを使用して上書きできます。詳細については、『*コマンドリファレンス*』の「*OP コマンド*」を参照してください。

タイムアウト条件を持つテーブルについては、『*コマンドリファレンス*』の「*タイムリミット*」を参照してください。

このパラメータに指定する数値は概算値であるという点に注意してください。実際の時間は、この値から最大10秒異なっていることがあります。

最小値は 20、デフォルト値は 900、最大値は 2592000 です。

例：

```
adanuc: tnaa=180
```

アクセスオンリーユーザーの非アクティブ時間制限は 180 秒となります。

TNAE

```
TNAE = number
```

このパラメータは、ET ロジックユーザーが Adabas コマンドを発行しなくてもアクティブでいられる最大経過時間（秒単位）を指定するものです。この値は、ADAOPR ユーティリティによってダイナミックに変更されることがあります。

この値は、OP コマンドを使用して上書きできます。詳細については、『*コマンドリファレンス*』の「*OP コマンド*」を参照してください。

タイムアウト条件を持つテーブルについては、『*コマンドリファレンス*』の「*タイムリミット*」を参照してください。

このパラメータに指定する数値は概算値であるという点に注意してください。実際の時間は、この値から最大10秒異なっていることがあります。

最小値は 20、デフォルト値は 900、最大値は 2592000 です。

例：

```
adanuc: tnae=180
```

ET ロジックユーザーの非アクティブ時間制限は 180 秒となります。

TNAX

```
TNAX = number
```

このパラメータは、ET ロジックを使用していない排他更新ユーザーが Adabas コマンドを発行しなくても、アクティブでいられる最大経過時間（秒単位）を指定するものです。この値は、ADAOPR ユーティリティによってダイナミックに変更されることがあります。

この値は、OP コマンドを使用して上書きできます。詳細については、『*コマンドリファレンス*』の「*OP コマンド*」を参照してください。

タイムアウト条件を持つテーブルについては、『*コマンドリファレンス*』の「*タイムリミット*」を参照してください。

このパラメータに指定する数値は概算値であるという点に注意してください。実際の時間は、この値から最大10秒異なっていることがあります。

最小値は 20、デフォルト値は 900、最大値は 2592000 です。

例：

```
adanuc: tnax=180
```

排他更新ユーザーの非アクティブ時間制限は 180 秒となります。

TT

```
TT = number
```

このパラメータは、ET ロジックユーザーによって発行される論理トランザクションに許可される最大経過時間（秒単位）を指定するものです。この値は、ADAOPR ユーティリティによってダイナミックに変更されることがあります。

この値は、OP コマンドを使用して上書きできます。詳細については、『*コマンドリファレンス*』の「*OP コマンド*」を参照してください。

論理トランザクションに対する時間の測定は、レコードをホールド状態に置いた最初のコマンドが発行されたときに開始され、ET、BT、または CL コマンドが発行されると終了します。この時間制限を超えると、Adabas は次の処置を行います。

1. トランザクション中のデータベースへの更新はすべてバックアウトされます。
2. トランザクション中にホールドされたレコードはすべて解放されます。
3. 該当ユーザーのコマンド ID がすべて解放されます。
4. ユーザーが次に発行するコールで、レスポンスコード 9 が返されます。

この時間制限は、非 ET ロジックユーザーには適用されません。このパラメータに指定した値は、直接 Adabas WORK データセットの必要サイズに影響を与えます。

このパラメータの上限は、ADAOPR の DISPLAY パラメータを使用して表示できます。

このパラメータに指定する数値は概算値であるという点に注意してください。実際の時間は、この値から最大10秒異なっていることがあります。

最小値は 20、デフォルト値は 300、最大値は 2592000 です。

例：

```
adanuc: tt=50
```

ET ロジックユーザーに対するトランザクションタイムリミットは 50 秒となります。

UNBUFFERED

```
UNBUFFERED = ALL | CLEAR | (keyword [, keyword [, keyword]]) ↩
```

このパラメータは、プラットフォームが UNIX プラットフォームであり、データベースコンテナまたはプロテクションログがファイルシステムに保存されている場合にだけ使用可能なパラメータです。一般的には、書き込みI/Oはファイルシステムによってバッファリングされます。しかし、書き込み I/O に相当する OPEN コールに O_DSYNC オプションが指定されていると、書き込み I/O 処理はバッファリングされません。このようにバッファリングを回避すると、パフォーマンスが上がることがあります。特に WORK と PLOG に、このオプションを指定すると効果があります。これは、トランザクションの終了時に、該当するログ情報がディスク上に実際に存在している必要があるからです。パラメータ UNBUFFERED を使用すれば、O_DSYNC 処理の使用方法を定義できます。

注意: このパラメータの使用は、Adabas のパフォーマンスにのみ影響します。
UNBUFFERED パラメータの実際の設定が最適なパフォーマンスを得るかどうかは、オペレーティングシステムと使用されているストレージシステムによって異なります。データベースの整合性は、UNBUFFERED パラメータのすべての値に対して保証されます。

キーワードリストに指定できるキーワードは、次のとおりです。

キーワード	説明
DATABASE	ファイルシステムの ASSO コンテナと DATA コンテナは、オプション O_DSYNC で開きます。
NODATABASE	ファイルシステムの ASSO コンテナと DATA コンテナは、オプション O_DSYNC で開きません。
WORK	ファイルシステムの WORK コンテナは、オプション O_DSYNC で開きます。
NOWORK	ファイルシステムの WORK コンテナは、オプション O_DSYNC で開きません。
PLOG	ファイルシステムのプロテクションログは、オプション O_DSYNC で開きます。
NOPLOG	ファイルシステムのプロテクションログは、オプション O_DSYNC で開きません。

キーワード "ALL" は (DATABASE, WORK, PLOG) と同じです。

キーワード "CLEAR" は (NODATABASE, NOWORK, NOPLOG) と同じです。

デフォルトは (NODATABASE, WORK, PLOG) です。

例：

```
adanuc: unbuffered=(nowork)
```

NODATABASE がデフォルトであるため、O_DSYNC は ASSO コンテナと DATA コンテナで使用されません。

NOWORK が指定されているので、O_DSYNC は WORK コンテナで使用されません。

PLOG がデフォルトなので、O_DSYNC はプロテクションログで使用されます。

USEREXITS

```
USEREXITS = (keyword [,keyword]...)
```

このパラメータは、ユーザー出口機能と組み合わせて使用し、1つまたは複数のユーザー出口を指定します。指定するユーザー出口は、実行時にロード可能でなければいけません。

キーワードとして、1、2、4、11、14 を指定できます。

注意:

1. ユーザー出口 1 と 11 は相互に排他的です。
2. ユーザー出口 4 は、CLOGLAYOUT=5 を指定した場合に限ってアクティブになります。ユーザー出口 14 は、CLOGLAYOUT=6 を指定した場合に限ってアクティブになります。

使用可能なユーザー出口に関しての詳細については、『*管理マニュアル*』の「*ユーザー出口とハイパー出口*」を参照してください。

WCHARSET

```
WCHARSET = <ICU encoding>
```

このパラメータは、ユーザーセッションの場合の W フィールドに対するデフォルトエンコードを指定します。このエンコーディングは、OP コールのレコードバッファ、または L または A/N コールのフォーマットバッファでエンコードが指定されていない場合に使用されます。

例

```
adanuc: wcharset=utf-16be
```

WRITE_LIMIT

```
WRITE_LIMIT = number
```

このパラメータは、暗黙のバッファフラッシュが実行されるまでの、バッファプール内で変更されたデータベースブロックの割合を指定します。バッファフラッシュの詳細については、『*管理マニュアル*』の「データベースのモニタリングとチューニング」にある「*バッファプール管理*」を参照してください。

有効な値は 1～50 です。デフォルト値は 50 です。互換性を維持するために、0 と 51～70 の値も使用できるようになっていますが、これらは 50 と等しくなります。

注意:

1. Adabas バージョン 6.4 以降では、WRITE_LIMIT に小さな値が設定されたときに一時ブロックの書き込み I/O 数を減らすために、一時ブロック（NUCTMP、NUCSRT）とデータベースブロック（ASSO、DATA）用に異なるフラッシュリストとバッファフラッシュが導入されました。一時ブロックは、バッファフラッシュを回避する場合にのみディスクに書き込むようにする必要があります。したがって、WRITE_LIMIT の値に関係なく、変更されたブロック（一時ブロックとデータベースブロックの合計）のパーセンテージが 50 の場合は、暗黙のバッファフラッシュが実行されます。このようなバッファフラッシュは、データの多くを占める変更済みブロックの種類に応じて、一時ブロックまたはデータベースブロックのいずれかに書き込まれます。
2. バッファプールが大きい場合、ニュークリアスの自動再スタートに必要な時間を削減するために、WRITE_LIMIT に小さな値を指定すると役に立ちます。自動再スタート中、ADANUC は、WORK コンテナの更新ログを読み取り、まだディスクに書き込まれていないすべての変更をデータベースに再適用する必要があります。WRITE_LIMIT の値が小さいということは、ディスクにまだ書き込まれていないデータ量が少ないことを意味しています。したがって、自動再スタート中に行われる更新は少なくなります。

例：

```

adanuc: write_limit=10
```

変更されたデータベースブロックのサイズがバッファプールサイズの 10% になると、暗黙的なバッファフラッシュが実行されます。

ADANUC パラメータの一覧

パラメータ	説明	最小値	最大値（注 1 を参照）	デフォルト値	動的な使用可（注 3 を参照）
[NO]AB32BIT	32 ビット共有メモリとして定義されたアタッチドバッファにより、32 ビットアプリケーションからの Adabas アクセスが可能（HP-UX にのみ関連）			AB32BIT	×
ADABAS_ACCESS	Adabas コールを実行できるユーザーを制限			ALL	×
APU	Adabas 処理ユニット			ありません。	×
AR_CONFLICT	バッファフラッシュ時のクラッシュ後の再起動			ABORT	×
BFIO_PARALLEL_LIMIT	並行に実行されるバッファフラッシュ I/O の上限	0（注 4 参照）	ありません。	50	○
[NO]BI	ビフォーイメージを PLOG に書き込む			BI	×
CLOGBMAX	ログに記録された Adabas バッファの長さ（CLOGLAYOUT=6 の場合のみ）	0（完全なバッファがログに記録されます）	4GB	4096	×
CLOGLAYOUT	CLOG のレイアウトを選択	5	6	5	×
DBID	データベース	1	255	ありません。	×
LAB	アタッチドバッファエリア	1MB	2GB	最大（1MB、NCL 値 * 1024）	×
LABX	アタッチドバッファ拡張エリア	1MB	ありません。	20MB	×
LBP	Adabas バッファプールサイズ（注 2 を参照）	32MB	ありません。	100MB	×
LOGGING	コマンドロギング			OFF	○
LPXA	XA データプロテクションエリアの大きさ	1 ブロック	65535 ブロック	XA オプション設定時は 10 ブロック	×

パラメータ	説明	最小値	最大値（注 1 を参照）	デフォルト値	動的な使用可（注 3 を参照）
LWP	Adabas ワークプール長	500 KB	4000 MB	16 MB	×
NCL	ローカルクライアントスレッドの数	2	ありません。	=NU、最低でも 50	×
NISNHQ	ホールド状態にできるユーザー当たりの最大 ISN 数	0（注 4 参照）	ありません。	0	○
NT	スレッド数	1	100	6	×
NU	ユーザーキューの大きさ	2 ユーザー	99999 ユーザー	20 ユーザー	×
OPTIONS	さまざまなオプション			オプションなし	一部のオプション
[NO]PLOG	プロテクションロギング			PLOG	×
READ_PARALLEL_LIMITS	先読み I/O パフォーマンスの最適化			(0,0,0)	○
TNAA	非アクティビティタイムリミット（アクセスオンリーユーザー）	20	2592000	900 秒	○
TNAE	非アクティビティタイムリミット（ET ロジックユーザー）	20	2592000	900 秒	○
TNAX	非アクティビティタイムリミット（EXU、EXF ユーザー）	20	2592000	900 秒	○
TT	トランザクションタイムリミット	20	2592000	300 秒	○
UNBUFFERED	O_DSYNC オプションの設定（UNIX プラットフォームのみ）			NODATABASE、WORK、PLOG	×
USEREXITS	使用するユーザー出口			ありません。	×
WCHARSET	W フィールドのデフォルトエンコード			なし（UTF-8）	×
WRITE_LIMIT	バッファプール変更リミット	1（注 5 参照）	50	50（注 5 参照）	○

注意:

1. 「なし」は、明示的な最大値がないことを意味しますが、共有メモリセグメントの最大サイズなど、使用可能なオペレーティングシステムリソースによって制限が生じる場合があります。
2. 指定した値が小さい場合、実際のバッファプールサイズは NT 値 * 2 MB に増やされます。共有メモリとして使用できないほど大きなバッファプールサイズを指定した場合、Adabasはより小さなバッファプールの割り当てを試みます。
3. ADAOPRを使用してダイナミックパラメータの値を変更すると、ニュークリアスの再起動を行わなくても、新しい値がすぐに反映されます。ダイナミックパラメータではない場合、新しい値を有効にするためには、まずニュークリアスを停止し、その後に再起動を行う必要があります。
4. 値 0 は制限がないことを意味します。
5. 互換性のため、値 0 および 51～70 も指定できます。これらは 50 と等しくなります。

22 ADAOPR（オペレータユーティリティ）

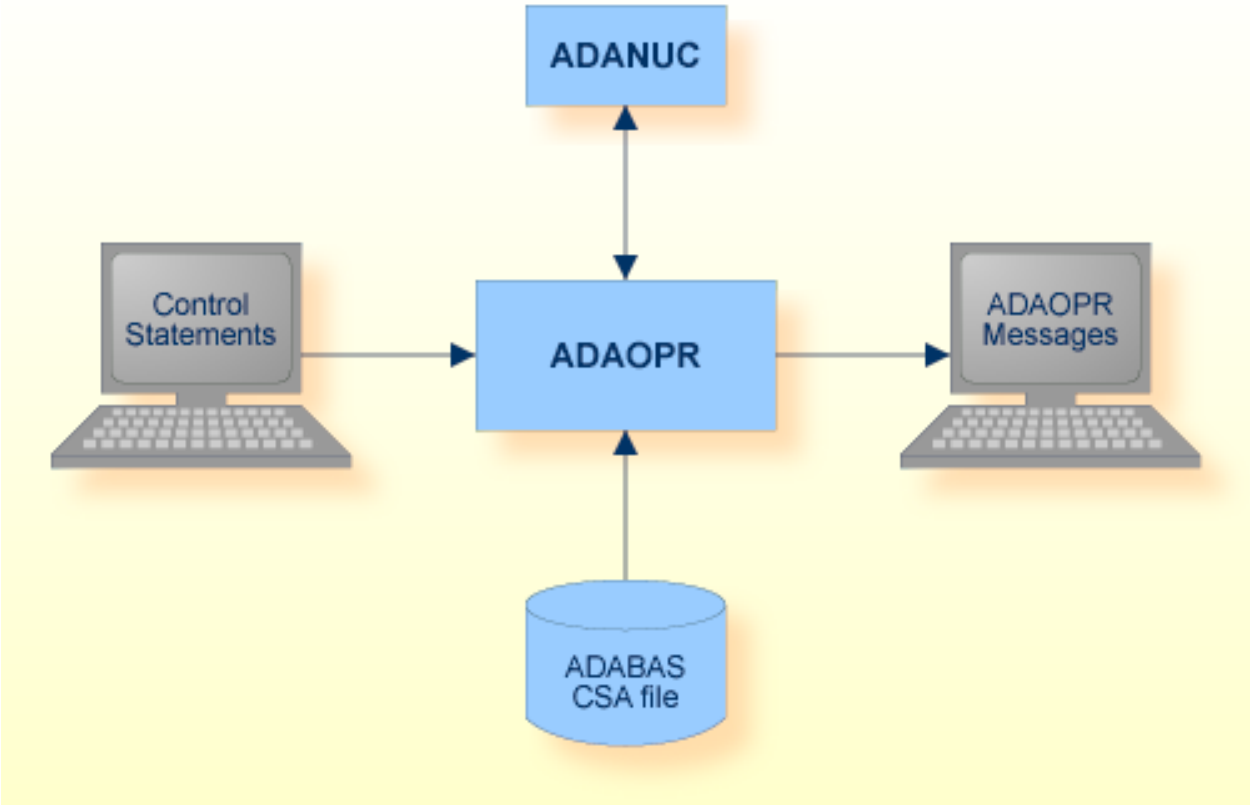
この章では ADAOPR ユーティリティについて説明します。

機能概要

DBA は、このユーティリティを用いて Adabas ニュークリアスを操作することができます。

このユーティリティは多機能ユーティリティです。

処理フロー



データセット	環境変数／論理名	記憶媒体	追加情報
コントロールステートメント	stdin/ SYS$INPUT		ユーティリティマニュアル
ADAOPR メッセージ	stdout/ SYS$OUTPUT		メッセージおよびコード

チェックポイント

次の表は、各機能に対するニュークリアス条件と記録チェックポイントを示しています。

機能	ニュークリアスのアクティブ化が必要	ニュークリアスの非アクティブ化が必要	ニュークリアスは不要	記録チェックポイント
FEOF=PLOG	x			SYNC（注 1 を参照）
EXT_BACKUP=PREPARE	x			SYNX (EXT_BACKUP STARTED)（注 2 を参照）
EXT_BACKUP=CONTINUE	x			SYNX (EXT_BACKUP) SYNC (FEOF=PLOG)（注 1 を参照）

注意:

1. FEOF=PLOG チェックポイントの後、ADANUC は新しい PLOG セッションの開始時に SYNC チェックポイントを書き込みます。
2. EXT_BACKUP=PREPARE のチェックポイントの書き込みは、Adabas バージョン 6.3 SP4 および Adabas バージョン 6.4 SP2 で導入されました。

制御パラメータ

次のコントロールパラメータを使用できます。

```
ABORT

ADD_REPLICATION [= number]
,FILE = number
,TARGET_DBID = number
,TARGET_FILE = number

BFIO_PARALLEL_LIMIT = number

CANCEL

CHANGE_REPLICATION keyword
,REPLICATION_ID = (number [ - number] [ , number [- number]] …

CLEAR_FILE_STATS = (number [- number] [, number [- number] ] ... )
```

```
     CSA = string

     DBID = number

     DELETE_REPLICATION = (number [ - number] [ , number [- number]] …

     DISPLAY = (keyword [,keyword]...)

     ES_ID = number

D    [NO]ET_SYNC

     [NO]EVENTING

     EXT_BACKUP = [PREPARE | CONTINUE | ABORT]

     FEOF = (keyword [,keyword])


     FILE = number


     FREE_CLQ

     ID = number

D    [NO]IO_TIME

     ISN = ( number [- number] [,number [- number] ] ... )

     [UN]LOCK = (number [,number]...)

     LOGGING = (keyword [,keyword]...)

     LOGIN_ID = string

     NISNHQ = number

     NODE_ID = string

     OPTIONS = (keyword [,keyword]...)

     READ_PARALLEL_LIMITS = (records,blocks,total)

     RESET = keyword

D    [NO]RESPONSE_ABORT

     RESPONSE_CHECK = (number[-number][,number[-number]]...)

     SET_FILE_STATS = (number[-number][,number[-number]]...)
```

```
    SHUTDOWN

    STATUS = (keyword [,keyword]...)

    STOP = (number[-number][,number[-number]]...)

    THREAD = number

    TNAA = number

    TNAE = number

    TNAX = number

    TT = number

    USER_ID = string

    WCHARSET = <ICU encoding>

    WRITE_LIMIT = [number]

    XA_RESPONSE_CHECK =  (keyword [,keyword]...)
```

ABORT

```
ABORT
```

この機能はAdabasセッションを直ちに終了します。すべてのコマンド処理は直ちに停止されます。該当セッションは、自動再スタートが保留のまま異常終了となります。

ABORT はデータベースのデフォルトディレクトリに次のファイルを書き出します。

- Adabas ニュークリアスからのステータス情報を格納する CSA ダンプファイル。ファイルの名前は、ADABAS.*xxx.hh:mm:ss*（UNIX）、ADABAS*xxx.hh-mm-ss*（Windows）、またはADABAS-*xxx-hh-mm-ss*（OpenVMS）です。ここで、*xxx*はデータベースID、*hh:mm:ss*（または *hh-mm-ss*）は、ファイルが作成された時刻です。ADAOPR の CSA パラメータを指定すると、実行中のニュークリアスからの情報を格納する CSA ダンプファイルから取得できるのと同じ情報が表示されます。
- 診断情報を含むSMPダンプファイル。ファイルの名前はSAGSMP.*xxx.hh:mm:ss*（UNIX）、SAGSMP*xxx.hh-mm-ss*（Windows）、またはSAGSMP-*xxx-hh-mm-ss*です。ここで、*xxx*はデータベースID、*hh:mm:ss*（または *hh-mm-ss*）は、ファイルが作成された時刻です。

ADD_REPLICATION

```
ADD_REPLICATION [= number]
                ,FILE = number
                ,TARGET_DBID = number
                ,TARGET_FILE = number
```

このパラメータは、Adabas を搭載した Event Replicator（Adabas レプリケーション）を使用している顧客にのみ関連します。

新規の Adabas-Adabas レプリケーションは、ステータス Inactive で定義されます。レプリケーション ID として、任意でゼロ以外の番号を指定できます。この番号には、既存のレプリケーションのレプリケーション ID は指定できません。番号を指定していない場合は、Adabas によってレプリケーション ID が作成されます。レプリケーションのソースファイル、ターゲットデータベース、およびターゲットファイル番号を指定する必要があります。

BFIO_PARALLEL_LIMIT

```
BFIO_PARALLEL_LIMIT = number
```

このパラメータは、バッファフラッシュによって並行に行われる I/O 要求数の上限を指定するのに使用し、他のスレッドから同時に発生した I/O のうち、処理の早いものを先に処理することを可能にします。例えば、大規模なバッファフラッシュが発生した場合、I/O キューがビジーになり、他の I/O（バッファプール読み取り I/O や WORK I/O など）が長時間キューに入れられ、コマンドスループットが遅くなり、バッファフラッシュがアクティブな場合はアプリケーションが停止したまま動かなくなる可能性もあります。

BFIO_PARALLEL_LIMIT を指定した場合、バッファフラッシュは指定された数の I/O を設定し、次のパケットを発行する前にこれらのフラッシュの処理が完了するまで待機します。"number" に指定できる最大値は、使用している Adabas システムによって決まります。値に 0 を指定した場合、バッファフラッシュ I/O の数には制限がありません。

CANCEL

```
CANCEL
```

この機能は Adabas セッションを直ちに終了します。アクティブな ET ユーザーごとに BT コマンドが発行された後、セッションは終了します。

データベースへのコミュニケーションリンクは切断されますが、共有メモリは維持されます。この場合、ADAOPR を使用することで表示機能は実行可能ですが、パラメータの変更はできなくなります。

CHANGE_REPLICATION

```
CHANGE_REPLICATION = keyword
                   , REPLICATION_ID = (number [ - number] [ , number [- number]] …
```

このパラメータは、Adabas を搭載した Event Replicator（Adabas レプリケーション）を使用している顧客にのみ関連します。

CHANGE_REPLICATIONは、1つまたは複数のレプリケーションのステータスの変更に使用できます。レプリケーションには、次のステータス値のいずれかを設定できます。

ステータス	説明
Inactive	現在、ターゲットファイルにはデータがレプリケートされておらず、レプリケーションを開始するためのアクティビティも実行されていません。
Prepare	これは、レプリケーションの初期状態処理を実行する予定であることを示します。このステータスは、パラメータ REPLICATION を指定した ADABCK を使用して、レプリケートするファイルのバックアップを作成するための前提条件です。
Initialization	これは、パラメータ REPLICATION を指定した ADABCK が実行中であり、レプリケートされるファイルの初期状態を含むバックアップが作成されていることを示します。
Recording	Adabasは、レプリケーションコマンドファイルおよびレプリケーショントランザクションファイル内の更新トランザクションを記録していますが、現在はターゲットデータベースに更新処理をレプリケートしていません。
Active	レプリケーションがアクティブです。ソースファイルのすべての変更は、ターゲットファイルにレプリケートされます。
Error	レプリケーション中に、不明なエラーが発生しました。レプリケーションを続けるには、新しい初期状態処理が必要です。

次のオプションを指定して、レプリケーションステータスを変更できます。

キーワード	説明
INACTIVE	現在、ターゲットファイルにはデータがレプリケートされておらず、レプリケーションを開始する予定もありません。ターゲットデータベースにまだレプリケートされていないトランザクションは削除されます。
INITIALIZATION	初期状態の処理を準備します。ステータスは Prepare に設定されます。次に、通常は、レプリケートするファイルのパラメータ REPLICATION を使用して、ADABCK DUMP/EXUDUMP を呼び出す必要があります。ADABCK は最初にステータスを初期化に設定し、次に ET 同期中に、データベース内の現在の状態がバックアップファイルと同じ場合、ADABCK はステータスを Recording に設定します。 または、独自の初期状態処理を実行してから、ADAOPR CHANGE_REPLICATION=RECORDING を実行することもできます。
RECORDING	次のいずれかになります。 ■ レプリケートされるファイルのデータは、パラメータ REPLICATION を指定した ADABCK を使用せずに保存されます。レプリケーションステータスが Recording に設定されます。これは、ターゲットファイルへのデータのコピーが完了して、ス

キーワード	説明
	テータスをActiveに設定するとすぐに、新しいデータベース変更をターゲットファイルにレプリケートするために記録されることを意味します。 **注意:** ADAOPR CHANGE_REPLICATION = RECORDING の実行は、パラメータ REPLICATION を指定した ADABCK でデータを保存する場合は必要ありません。この場合、ADABCK はステータスを Recording に設定します。 ■ レプリケーションが停止され、新しいデータベース更新は記録のみ行われます。レプリケーションが再度アクティブ化されるとすぐに、データベース更新がターゲットデータベースにレプリケートされます。
ACTIVE	レプリケーションがアクティブです。ソースファイルのすべての変更は、ターゲットファイルにレプリケートされます。

ステータス変更の実行対象に対して、レプリケーション ID を指定する必要があります。

注意:

1. 変更するレプリケーションのリストに参照整合性制約のあるファイルが含まれている場合は、関連ファイルおよび同じターゲットデータベースのレプリケーションも指定する必要があります。
2. 次のマトリックスは、現在のレプリケーションステータスおよびその結果のステータス変更に応じて許可されるオプションを示しています。

ステータス／キーワード	INACTIVE	INITIALIZATION	RECORDING	ACTIVE
Inactive	Inactive	準備	-	-
準備	Inactive	準備	記録（注意 1 を参照）	-
初期化	Inactive	Init	-	-
記録	Inactive	記録	記録	Active
Active	Inactive	準備	Recording（注 2 を参照）	Active
エラー	Inactive	準備	-	-

注意:

1. 初期状態の処理としては、パラメータ REPLICATION を指定して ADABCK を使用する方法が推奨されます。次に、バックアップが完了すると、ADABCK はステータスをまず Initialization に設定し、その後に Recording に設定します。初期状態の処理に ADABCK を使用しない場合（例えば空のファイルでレプリケーションを開始する場合）のみ、ステータスを Recording に設定する必要があります。
2. メンテナンスなどのためにターゲットデータベースをシャットダウンした場合は、レプリケーションを Recording に設定する必要はありません。その後、データベースのステータスは Active のままになり、ターゲットデータベースは再び使用可能になるまでポーリングされます。日中に実行した更新を夜間にレプリケートする場合や、ターゲットデータベースに前日

のデータベース状態が含まれている必要がある場合は、ステータスを Recording に設定すると便利です。

CLEAR_FILE_STATS

```
CLEAR_FILE_STATS = (number [- number] [, number [- number] ] ... )
```

このパラメータは、特定のファイルに SET_FILE_STATS を指定することで有効にした I/O 統計の収集を無効にします。

CSA

```
CSA = string
```

"string" には Adabas ニュークリアスからのステータス情報を格納するファイル（CSA ダンプファイル）を指定します。このファイルは、ADAOPR ABORT 機能を実行したとき、Adabas が異常終了したとき、レスポンスチェック（詳細については RESPONSE_CHECK パラメータを参照）をトラップしたときに作成されます。

ファイルには次の命名規則が適用されます。

UNIX

```
ADABAS.xxx.hh:mm:ss
ADABAS.xxx.RSPyyy.hh:mm:ss
```

Windows

```
ADABAS.xxx.hh-mm-ss
ADABAS.xxx.RSPyyy.hh-mm-ss
```

OpenVMS

```
ADABAS-xxx-hh-mm-ss
ADABAS-xxx-RSPyyy.hh-mm-ss
```

NORESPONSE_ABORT パラメータを設定していることを前提としています。値の意味は次のとおりです。

- "xxx" は 3 桁のデータベース ID
- "yyy" はトラップした 3 桁のレスポンスコード
- "hh:mm:ss" はファイルが作成された時刻（UNIX）
- "hh-mm-ss" はファイルが作成された時刻（Windows および OpenVMS）

例えば、データベース ID が 5 で、トラップしたレスポンスコード 113 でファイル作成が始まった場合、ファイル名は ADABAS.005.RSP113 で始まります。そして、ADABAS.005.RSP113.12:16:50（UNIX）、ADABAS.005.RSP113.12-16-50（Windows）、ADABAS-005-RSP113.12-16-50（OpenVMS）のように作成時刻が付加されます。

ファイルは、環境変数／論理名 ADA_CSA_DUMP でポイントされているディレクトリに作成されます。デフォルトはニュークリアスの起動ディレクトリです。同じ名前を持つファイルがこのディレクトリにすでに存在している場合、それは上書きされます。

DBID パラメータと CSA パラメータは相互に排他的です。

DBID

```
DBID = number
```

このパラメータは、後続の ADAOPR コマンドに適用するデータベースを選択します。1 つのセッションで複数の DBID がサポートされます。

DBID パラメータと CSA パラメータは相互に排他的です。

例：

```
adaopr: dbid=1
adaopr: shutdown
adaopr: dbid=2
adaopr: shutdown
adaopr: dbid=3
adaopr: shutdown
adaopr: quit
```

DELETE_REPLICATION

```
DELETE_REPLICATION = (number [ - number] [ , number [- number]] …
```

このパラメータは、Adabas-Adabas レプリケーションで Adabas Event Replicator を使用している顧客にのみ関連します。

指定されたレプリケーション ID を持つレプリケーションがアクティブな場合は停止され、ターゲットファイルにまだレプリケートされていないコマンドやトランザクションも含めて削除されます。

DISPLAY

```
DISPLAY = (keyword [,keyword]...)
```

このパラメータは、Adabas セッション時の各種情報の表示に使用します。

次のキーワードを使用できます。

キーワード	説明
ACTIVITY	データベースの稼動状況を表示します。
BF_STATISTICS	バッファフラッシュの統計を表示します。
BP_STATISTICS	バッファプールの統計を表示します。
COMMANDS	コマンドテーブルを表示します。
CQ	コマンドキューを表示します。
DYNAMIC_PARAMETERS	動的なニュークリアスパラメータを表示します。
FILE_IO	ファイル I/O を表示します。
FP_STATISTICS	フォーマットプールの統計を表示します。
HIGH_WATER	最大使用量を表示します。
HQ	ホールドキューを表示します。
ICQ	内部コマンドキューを表示します。
IO_TIMES	コンテナの I/O 時間を表示します。
PLOG_STATISTICS	プロテクションログの統計を表示します。
REPLICATIONS	Adabas-Adabas レプリケーション。
RPL_STATS	内部収集されたレプリケーション統計。
STATIC_PARAMETERS	静的なニュークリアスパラメータを表示します。
TT	スレッドテーブルを表示します。
UCB	ユーティリティコミュニケーションブロックを表示します。
UQ	ユーザーキューを表示します。
UQ_FILES	ユーザーファイルリストを表示します。
UQ_FULL	ユーザーキューエレメントについての完全な情報を表示します。
UQ_TIME_LIMITS	ユーザーのタイムリミットを表示します。

ここからは、各種のキーワードで出力される情報と、表示される情報について説明します。

表示例の中には、割合を含んでいるものもあります。該当する値は、どれも切り捨てられた値です。未定義の値（ゼロ割り）は " *%"、オーバーフローは "***%" として表されます。

例：DISPLAY=ACTIVITY

```
adaopr: display=activity

                      ADANUC Version <version number>
       Database 76           Activity          on 22-JAN-2014 13:19:30


I/O Activity                     Total   Throwbacks                       Total
------------                     -----   ----------                       -----
Buffer Pool                      5,440   Waiting for UQ context              87
WORK Read                          728   Waiting for ISN                     53
WORK Write                         647   ET Sync                              0
PLOG Write                         194   DWP Overflow                         0
NUCTMP                           1,600
NUCSRT                             531

Pool Hit Rate                    Total   Interrupts       Current         Total
-------------                    -----   ----------       -------         -----
Buffer Pool                        99.6% WP Space Wait           0             0
Format pool                        98%
```

情報の意味は次のとおりです。

- ［I/O ACTIVITY］は、次の処理回数の合計を示します。
  - 物理バッファプール I/O（物理読み込み I/O 回数 + 物理書き込み I/O 回数）
  - WORK および PLOG に対する読み込みおよび書き込み I/O 回数
  - NUCTMP および NUCSRT の I/O 回数
- ［INTERRUPTS］は、ワークスペースのスペース待ちの現在の回数と合計回数を示します。
- ［POOL HIT RATE］には、次の情報が表示されます。
  - バッファプールヒット率。これは論理的な読み込み I/O と、物理的な読み込み I/O との間の関係を表します。バッファプールヒット率は次の計算式で求められます。

    ```
    hit rate (in %) =  ((logical read I/Os - physical read I/Os) * 100)
        / logical read I/Os
    ```

  - フォーマットプールヒット率。フォーマットバッファの要求回数（要求された FB 数）と、フォーマットプールですでに変換された要求済みフォーマットバッファの数（変換された FB 数）との間の関係を表します。フォーマットプールヒット率は次の計算式で求められます。

    ```
    hit rate (in %) = ((translated FBs * 100) / required FBs)
    ```

- ［THROWBACKS］には、次の情報が表示されます。

- 内部コマンドが実行していたためにセッションコンテキスト待ち状態になっているコマンドの数。
- ISN が他のユーザーにホールドされているために待ち状態になっているコマンドの数。
- ET の同期のために待ち状態になっているコマンドの数。
- 動的なワークプールのオーバーフローのために戻されたコマンドの数。

例：DISPLAY=BF_STATISTICS

```
adaopr: disp=bf_statistics
%ADAOPR-I-STARTED,      18-OCT-2016 16:05:03 Version <version number>

Database 37, startup at 18-OCT-2016 16:04:40
ADANUC Version 6.5.1.0, PID 10448

                        ADANUC Version 6.5.1.0
        Database 37     Buffer Flush Statistics  on 18-OCT-2016 16:05:02


Buffer flush statistics:
------------------------

Buffer flush          Write Number of Type    Size  Average IO Duration Rejected
start time            Limit    Blocks         (MB) time (msec)    (sec)    Locks
-------------------------------------------------------------------------------
18-OCT-2016 16:04:40     2         4 DB      0.04         0.00     0.00         0
18-OCT-2016 16:04:59     2       128 DB      0.54         0.35     0.04         0
18-OCT-2016 16:05:00     2       128 DB      0.53         0.96     0.12         1
18-OCT-2016 16:05:00     2         9 DB      0.06         0.00     0.00         0
18-OCT-2016 16:05:00     2         5 DB      0.04         3.00     0.01         0
18-OCT-2016 16:05:01     2       126 DB      0.53         0.98     0.12         1
18-OCT-2016 16:05:01     2         7 DB      0.05        11.00     0.07         0
18-OCT-2016 16:05:01     2        12 DB      0.07         0.00     0.00         0
18-OCT-2016 16:05:02     2       128 DB      0.54         0.85     0.10         0
18-OCT-2016 16:05:03     2       131 DB      0.55         0.10     0.01         0

Total number of flushes:        10
Explicit               :         1
Write limit            :         0
WORK limit             :         5
Space                  :         0
Emergency              :         0
Ignored blocks         :         4
```

バッファフラッシュの統計を表示します。現在のニュークリアスセッションで 100 回を超えるバッファフラッシュが実行された場合は、最後の100回のバッファフラッシュが表示されます。各バッファフラッシュについて、次の情報が表示されます。

- バッファフラッシュの開始時刻。

- 現在の書き込み制限。データベースブロックの書き込み制限は、ADAOPR WRITE_LIMIT を使用して変更できます。一時ブロックの書き込み制限は変更できません。
- バッファフラッシュに含まれるブロックの数。
- バッファフラッシュのタイプ：
  - DB は、データベースブロックのフラッシュ
  - Temp は、一時ブロックのフラッシュ
- バッファフラッシュに含まれるブロックのサイズ（メガバイト単位）。
- バッファフラッシュで実行される I/O の平均 I/O 時間（ミリ秒単位）。
- バッファフラッシュの期間（秒単位）。
- 拒否されたロックの数とは、バッファフラッシュがブロックの書き込みを試行したときにブロックが排他的にロックされていたことが理由で、バッファフラッシュ中にすぐに書き込まれなかったブロックの数です。拒否されたブロックは、他のブロックの書き込み後に書き込まれるか（その場合バッファフラッシュはロックが許可されるまで待機します）、別の ignore-blocks バッファフラッシュによって書き込まれます。
-

表の後には、バッファフラッシュの総数が表示され、バッファフラッシュの理由の内訳が示されます。

注意:

1. 上の例で示したデータベースでは、書き込み制限を超える前に WORK 制限バッファフラッシュが発生する、小さなWORKコンテナが使用されています。その理由から、上の例に示したデータベースでは、WORK 制限バッファフラッシュが表示されていますが、書き込み制限バッファフラッシュは表示されていません。
2. 2つのスレッドでほぼ同時に、バッファフラッシュが必要だと判断されることがあります。次に、両方のスレッドがバッファフラッシュに必要なフラグを設定します。最初のスレッドがフラグを設定し、バッファフラシュスレッドがバッファフラッシュを開始して、フラグをリセットします。2番目のスレッドが再度フラグを設定します。バッファフラッシュが終了すると、新しいバッファフラシュがすぐに開始されます。このような不要なバッファフラッシュではエラーが発生しないので、このようなバッファフラッシュを回避するロジックは実装されていません。この例では、5 番目と 8 番目のバッファフラッシュがそのような不要なバッファフラッシュです。これらは "Ignored blocks" バッファフラッシュとして表示されるので、無視されたブロックが2つだけでも、4つの Ignored blocks バッファフラッシュが表示されます。

例：DISPLAY=BP_STATISTICS

```
adaopr: display=bp_statistics

                      ADANUC Version  <version number>
        Database 34   Buffer Pool Statistics   on  5-JUN-2014 13:11:28


Buffer Pool Size   :   419,430,400

Pool Allocation                         RABNs present
---------------                         -------------
Current     (  7%) :    32,835,584      ASSO               :            33
Highwater   ( 10%) :    42,676,224      DATA               :             5
Internal    (  7%) :    30,770,176      WORK               :             0
Workpool    (  0%) :     1,408,000      NUCTMP             :             0
                                        NUCSRT             :             0

I/O Statistics                          Buffer Flushes
--------------                          --------------
Logical Reads      :           340      Total              :             3
Physical Reads     :            17      To Free Space      :             0
Pool Hit Rate      :            95.0%   Temporary Blocks   :             0

                                        Write Limit  (  2%):     8,388,600
Physical Writes    :            41      Modified     (  0%):       108,544

                                        Limit Temp.B.( 50%):   209,715,000
                                        Modified T.B.(  0%):             0
```

情報の意味は次のとおりです。

- ［POOL ALLOCATION］には、次の情報が表示されます。
  - 現在使用中のバッファプールのサイズ（バイト数）と割合。
  - バッファプールのピーク時のサイズ（バイト数）および割合（DISPLAY=HIGH_WATERのときの表示も参照）。
- ［RABNs PRESENT］には、次の情報が表示されます。
  - バッファプールで現在使用されている ASSO、DATA および WORK RABN の数。
- ［I/O STATISTICS］には、次の情報が表示されます。
  - 論理および物理バッファプール読み込みI/Oの合計（バッファプールのヒット率を求めるために両方の回数が必要）。
  - バッファプールのヒット率（バッファプールヒット率の計算式は DISPLAY=ACTIVITY の例を参照）。
  - 物理バッファプール書き込み I/O の合計回数。

- ［BUFFER FLUSHES］には、次の情報が表示されます。
  - バッファフラッシュの合計回数。
  - フリースペースを取得するために行われたバッファフラッシュの合計回数。
  - 一時ブロックのバッファフラッシュの合計数。
  - データベースブロックのバッファプール WRITE LIMIT のサイズと割合。
  - 変更されたデータベースブロックのサイズ（バイト数）と割合。
  - 一時ブロックのバッファプール WRITE LIMIT のサイズと割合。
  - 変更された一時ブロックのサイズ（バイト数）と割合。

例：DISPLAY=COMMANDS

```
adaopr: display=commands

                      ADANUC Version <version number>
        Database 76           Commands          on 19-JAN-2014 14:58:10


ADABAS Commands:             9,884

        A1             892         L2             553        OP             25
        BT             736         L3           1,124        RC             89
        C1              40         L4             569        RE              0
        C3               0         L5             420        RI              0
        C5              10         L6             436        S1          1,511
        CL              32         L9             456        S2             81
        E1           1,006         LF              20        S4             12
        ET              72         MC               0        S8            230
        HI               0         N1             877        S9             50
        L1             643         N2               0
```

このコマンドは、現セッションで発行された Adabas コマンドの数の合計を表示します。MC コマンドの場合、表示される値は、MC コール回数と MC コールに含まれる単一 Adabas コマンド数を足したものです。

マルチフェッチオプションが設定された状態で発行された読み込みコマンドは単一コマンドとしてカウントされます。

ユーティリティによって行われた更新は表示に含まれません。

注意: コマンドカウントは、ADAOPR RESET=COMMANDS でリセットできます。

例：DISPLAY=CQ

```
adaopr: display=cq
                      ADANUC Version <version number>
        Database 2         Command Queue        on 14-NOV-2014 13:41:53


  No   Node Id  Login Id       ES Id  APU Cmd File  Status
  --   -------  --------       -----  --- --- ----  -----
   1  PC0001    miller          3316    1  RC   13  Ready to run
   2  PC0001    jones           1360    2  S8   13  Running
   3  PC0001    smith           6148    1  RC   13  Ready to run
   4  PC0001    miller          4208    1  S8   13  Running
   5  PC0001    jones           5224    2  S9   13  Ready to run
   6  PC0001    dba             7024    2  U1    0  Running
   7  PC0001    brown           3140    2  S1   13  Running
   8  PC0001    meyer           6180    2  S8   13  Running
   9  PC0001    smith           4756    1  S1   13  Running
  10  PC0001    king            1240    2  ET    0  Ready to run
  11  PC0001    meyer            836    2  RC   13  Ready to run
  12  PC0001    brown           6272    1  L6   13  Ready to run

Selected: 12, Used: 12, Queue Size: 13
```

この例では、現在のコマンドキューエントリが表示されています。

- ［NODE ID］には、ノード ID 文字列が表示されます。
- ［LOGIN ID］には、ログインユーザー ID 文字列が表示されます。
- ［ES ID］には、環境固有の ID（プロセス ID など）が表示されます。
- ニュークリアスパラメータ APU が設定されている場合、APU にはコマンドキューエントリの割り当てられた Adabas 処理ユニットが表示されます。APU が指定されていない場合、列 APU は表示されません。
- ［CMD］には、コマンド文字列が表示されます。
- ［FILE］には、ファイル番号が表示されます。
- ［STATUS］には、コマンドキューエントリのステータスが表示されます。

画面の最後の行は、現在有効な選択条件に従って選択されたコマンドキューエントリの数と、コマンドキューで使用されたエントリ全体の数を示します。

ステータスとして表示される値は次の表のとおりです。

ステータス	説明
Completed	コマンド処理完了。
Marked For Deletion	コマンドは削除のためにマークされます。ユーザーはアクティブではなくなります。
New	コマンドはスケジューリングキューに挿入される用意ができています。
Ready To Run	キューに配置され、スケジューリングの用意ができています。
Running	スレッドで実行中です（DISPLAY=TT を参照）。
Waiting For Complex	複合コマンドは実行待ち状態です。
Waiting For Et Sync	ET 同期を待機しています。
Waiting For Group Commit	グループ ET を待機しています。スレッドテーブルにエントリがありません。
Waiting For Isn <isn>	画面の［File］列に表示されている ISN を待機しています。スレッドテーブルにエントリがありません。
Waiting For Space	作業スペースを待機しています。エントリはスレッドテーブルにありません。
Waiting For Uqe	ユーザーキューエントリを待機しています。要求されたエントリはアクティブな内部コマンドによってロックされています。

注意: 画面には、［U0］などのコマンドコードが示されることがありますが、これは Adabas が内部的に使用するコードです（例えばユーティリティの実行中など）。ユーザーが明示的な選択条件を指定しなかった場合でも、［RUNNING］と［COMPLETED］の値は異なることがあります。

例：DISPLAY=DYNAMIC_PARAMETERS

```
adaopr: display=dynamic_parameters

                      ADANUC Version <version number>
        Database 76      Dynamic Parameters     on 19-JAN-2014 14:58:10


Resources:          NISNHQ    :           100    WRITE_LIMIT:            -

Time Slices:        TNAA      :           900    TNAX       :          900
                    TNAE      :           900    TT         :          300

Logging:            CLOG      : OFF


Read limits:        200, 10, 30

Response check with ABORT    : 84,160,164-182,243,251-252
```

上記の表示は動的なニュークリアスパラメータの現在の値を示しています。

例：DISPLAY=FILE_IO

```
adaopr: display=file_io

                      ADANUC Version <version number>
       Database 76           File I/O          on 19-JAN-2014 14:58:10


                      Reads           Hit
    File       Logical       Physical  Rate         Writes
    ----       -------       --------  ----         ------
      11       145,341            180   99%          2,869
      12        99,070            148   99%          2,149
```

この表示は、対象ファイルの I/O 統計ファイルが有効（ADAOPR SET_FILE_STATS）にされてからの論理的および物理的な読み取り、ヒット率、および各ファイルに対するバッファプールマネージャの書き込みを示します。I/O 統計が有効になっていないファイルや、I/O がまったく実行されなかったファイルは表示されません。

注意:

1. ヒット率の値の公式については、DISPLAY=ACTIVITYの記述を参照してください。
2. 書き込み処理は、ブロックが変更済みとしてマークされていなかった場合にのみカウントされます。つまり、前のバッファフラッシュですでに実行されている物理的書き込み I/O や、次回のバッファフラッシュで実行される保留中の物理的書き込み I/O がカウントされます。

例：DISPLAY=FP_STATISTICS

```
adaopr: display=fp_statistics

                      ADANUC Version <version number>
       Database 76    Format Pool Statistics   on 19-JAN-2014 14:58:10

Maximum Local  Pool Size:        251,656
Maximum Global Pool Size:        251,656

Pool Allocation                             Pool Contents
---------------                             -------------
Local  Current   ( 22%) :         57,540   Local  Format Buffers:         162
Local  Highwater ( 27%) :         70,000   Global Format Buffers:           1

Global Current   (  0%) :             84
Global Highwater (  0%) :             84


Pool Statistics                    Local          Global
---------------                    -----          ------
```

```
Scans                              11,780              3
Hits                               11,547              2
Hit Rate                               98%            66%

Replacements                            0              0
Overflows                               0              0
```

上記の表示はフォーマットプール統計を示しています。

- ［POOL ALLOCATION］には、次の情報が表示されます。
  - 現在使用されているローカルおよびグローバルフォーマットプールのサイズ（バイト数）と割合。
  - ローカルおよびグローバルフォーマットプールの最大使用量のサイズ（バイト数）と割合。
- ［POOL STATISTICS］には、次の情報が表示されます。
  - フォーマットプールの有効なフォーマットバッファのスキャン回数とヒットした回数の合計（フォーマットプールのヒット率を計算するには両方の回数が必要）。
  - フォーマットプールのヒット率（フォーマットプールのヒット率の計算式については DISPLAY=ACTIVITY の例を参照）。
  - フォーマットプールで上書きされた有効なフォーマットバッファの合計数（［Replacements］の表示）。
  - オーバーフロー。これは、フォーマットバッファがフォーマットプールサイズを超過したため、発生したレスポンス 42 の回数です。
- ［POOL CONTENTS］には、次の情報が表示されます。
  - フォーマットプールで有効なローカルフォーマットバッファの数。
  - フォーマットプールで有効なグローバルフォーマットバッファの数。

例：DISPLAY=HIGH_WATER

```
adaopr: display=high_water
                        ADANUC Version <version number>
        Database 2        High Water Marks      on 21-NOV-2014 11:44:19

Area/Entry                Size      In Use  High Water    %      Date/Time
----------                ----      ------  ----------    -      ---------
User Queue                 100          13          13   13 21-NOV-2014 11:44:00
Command Queue                -          12          13    - 21-NOV-2014 11:44:19
  APU 01                     -           2          12    - 21-NOV-2014 11:44:02
  APU 02                     -          13          15    - 21-NOV-2014 11:44:00
Hold Queue                   -           2           2    - 21-NOV-2014 11:44:00
Client Queue               100          13          13   13 21-NOV-2014 11:44:00
HQ User Limit                -           -           1    - 21-NOV-2014 11:44:00
Threads                      6           4           6  100 21-NOV-2014 11:44:00
Workpool           524,288,000           0 131,072,016   25 21-NOV-2014 11:42:16
```

```
  ISN Sort          65,536,000            -      380,000     0 21-NOV-2014 11:44:04
  Complex Search    65,536,000            -            0     0
Attached Buffer      1,048,576      219,136      219,136    20 21-NOV-2014 11:44:02
ATBX       (MB)             20            0            0     0
Buffer Pool(KB)      2,048,000      957,962    1,009,978    49 21-NOV-2014 11:42:16
Protection Area        127,990
  Active Area           38,397            -            4     0 21-NOV-2014 11:44:04
Group Commit                50            1            1     2 21-NOV-2014 11:42:17
Transaction Time         3,000            -            0     0
```

上記の表示は、現セッションに対する最大使用量を示しています。

- ［SIZE］には、プールおよびバッファのサイズ（バイト数）が表示されます。キュー、スレッド、およびホールドキューユーザーリミットの場合、エントリの数が表示されます。
- ［IN USE］には、現在使用中のサイズ（バイト数）または個数が表示されます。
- ［HIGH WATER］には、それぞれのエリア／エントリで同時に必要になった最大の量を示します。
- ［%］には上限とサイズの関係が表示されます。最大使用量がサイズを超過すると、この列の値は 100 %より大きくなることがあります。例えば、値が ADAOPR によって減少した場合や、元の領域が動的に増加した場合に発生する可能性があります。これは通常の Adabas 動作です。Adabas パラメータの変更は必要ありません。
- DATE/TIME には、初めて最大使用量になった日付／時刻が表示されます。最大使用量が 0 である場合、この列には何も出力されません。

［AREA/ENTRY］列に表示されるエントリは、ADANUC パラメータ NU（ユーザーキュー）、NCL（クライアントキュー）、NISNHQ（ホールドキューユーザー制限)、NT（スレッド）、APU（Adabas 処理ユニット、ニュークリアスパラメータ APU が設定されている場合にのみ表示）、LWP（ワークプール）、LBP（バッファプール）、LAB（アタッチドバッファ）、TT（トランザクション時間）に対応します。ホールドキューとコマンドキューには、事前定義されたサイズがありません。必要に応じて動的に増加します。

［ACTIVE AREA］には、単一トランザクションで使用できる WORK パート 1 の最大部分が表示されます。トランザクションのプロテクション情報が［Active Area］で許可されているスペースを超えると、レスポンス 9（LP）が返され、ニュークリアスは PLOVFL メッセージを表示し、最大使用表示の［%］列に 100 を超える値を表示します。

OP コールにユーザー固有のタイムアウト値を設定したユーザーは、［Transaction Time］の値に含まれません。

注意: 1.ADAOPR の設定が原因で、ニュークリアスから情報を取得できない場合（例えば ADAOPR パラメータ CSA を使用したとき）、［Attached Buffer］と［Command Queue］には正しい値が表示されません。
2.スレッドはラウンドロビン方式で使用されます。したがって、スレッドに対する最大使用量は、ほとんどの場合、［Size］列に表示される値と同じになります。

3.ニュークリアスの異常終了後の自動再スタート中に、その時間内にアクティブで、更新を回復する必要があるユーザーに対して、ユーザーキューエレメントが作成されます。そのため、新しいニュークリアスセッションの開始直後は、ユーザーキューの最大使用量が比較的多くなることがありますが、使用中のユーザーキューエレメントの数は少なくなります。

例：DISPLAY=HQ

```
adaopr: file=11, display=hq

                    ADANUC Version <version number>
        Database 76          Hold Queue          on 19-JAN-2014 14:58:10

        Id Node Id  Login Id      ES Id User Id  File           ISN Locks  Flg
        -- -------  --------      ----- -------  ----           --- -----  ---
        15 sunxxx01 miller         6974 *adatst    11         2,222   X     M
        19 sunxxx01 smith          7056 *adatst    11             2   X

Selected: 2, Used: 8, Queue Size: 160
```

上記の表示は現在のホールドキューエントリを示しています。

- ［ID］には、ISN をホールドしているユーザーのユーザー ID が表示されます。
- ［NODE ID］には、ノード ID 文字列が表示されます。ローカルノードは空白によって表現されます。
- ［LOGIN ID］には、ログインユーザー ID 文字列が表示されます。
- ［ES ID］には、環境固有の ID（プロセス ID など）が表示されます。
- ［USER ID］には、ユーザー ID が表示されます。［USER ID］の値は、アスタリスクとユーティリティ名をつなげた文字です。
- ［FILE］には、ISN が存在する Adabas ファイルの番号が表示されます。
- ［ISN］には、ホールドされている ISN の番号が表示されます。
- LOCKS は、ISN のロックの種類を示します。ここで X は排他的ロック、S は共有ロックです。

  注意: Adabas バージョン 6.3 SP1 以降では共有ロックは S と表示されますが、以前のリリースでは R と表示されていました。

- ［FLG］に［M］と表示されている場合は、レコードが変更されたことを表します。

画面の最終行は、現在有効な選択条件にしたがって選択されたホールドキューエントリの数、および全体で使用されたエントリ数を示しています。

エントリの並び順は、ソートされて表示されるわけではありません。

例：DISPLAY=ICQ

```
adaopr: display=icq

                      ADANUC Version <version number>
       Database 76    Internal Command Queue   on 19-JAN-2014 14:58:10


       Id  Node Id   Login Id       ES Id  Command   Status
       --  -------   --------       -----  -------   ------
  00000002           *system     00000000  SHUT      Running

Selected: 1, Used: 1, Queue Size: 101
```

上記の表示は内部コマンドキューを示しています。その内容は次のとおりです。

Command	説明
AR	自動再スタート
BT	トランザクションのバックアウト
BTCL	トランザクションのバックアウトおよびユーザーのクローズ
CANCEL	ニュークリアスのキャンセル
DELUQE	ファイルリストの解放およびユーザーキューエレメントの削除
ETSYNC	グローバルトランザクションがタイムアウトを受け取った後にET-SYNCステータスチェックを開始
SHUT	ニュークリアスのシャットダウン
STOP	ADAOPRからの指示による停止
TIMEOUT	非アクティビティタイムアウト

内部コマンドのステータスは、［READY TO RUN］、［RUNNING］、［WAITING FOR ET SYNC］または［WAITING FOR UQE］です。

画面の最終行は、現在有効な選択条件に従って選択された内部コマンドキューの数と全体で使用されたエントリ数を示しています。

例：DISPLAY=IO_TIMES

```
adaopr: display=io_times

                      ADANUC Version <version number>
       Database 76        IO Statistics        on 19-NOV-2014 12:16:48


                 Number of IOs       Maximum IO time      Average IO time
                 -------------       ---------------      ---------------
```

```
ASSO Read    :              735574              14397                 1
ASSO Write   :               12136                  2                 1
DATA Read    :             2023257              13910                 1
DATA Write   :                 444                  1                 1
WORK Read    :                   4                  1                 1
WORK Write   :                 660                  2                 1
NUCSRT Read  :                4060                940                 1
NUCSRT Write :                4060                  1                 0
NUCTMP Read  :                  30                  1                 1
NUCTMP Write :                 896                  1                 1
```

［Number of IOs］には、ASSO、DATA、WORK、NUCSRT、および NUCTMP への物理的な読み取りおよび書き込み I/O アクセスの回数が表示されます。

［Maximum IO time］には、ASSO、DATA、WORK、NUCSRT、および NUCTMP への 1 回の I/O の読み取りおよび書き込みアクセスの最大期間がマイクロ秒単位で表示されます。

［Average IO time］には、ASSO、DATA、WORK、NUCSRT、および NUCTMP への 1 回の I/O アクセスの平均時間が表示されます。

I/O 時間のログ記録は、ADAOPR IO_TIME が有効になっている場合にのみ使用できます。

例：DISPLAY=PLOG_STATISTICS

```
adaopr: display=plog_statistics

                      ADANUC Version <version number>
        Database 76       PLOG Statistics       on 19-JAN-2014 14:59:41


PLOG Environment
----------------
NUCPLG     (active) : /FS/fsxxxx/sag/ada6180102/ada/db076/NUCPLG


Active PLOG
-----------
Session Number      :  37
Extent              :  2

Active Since        :  19-JAN-2014 14:59:41
Duration            :              00:00:01

Allocated Space     :         24,683 KB
Used Space   (  0%) :             32 KB
Average Growth Rate :        115,200 KB/h
```

例：DISPLAY=REPLICATIONS

```
adaopr: display=replications
                      ADANUC Version <version number>
        Database 34         Replications        on 19-JAN-2014 09:47:48


       ID  From FNR  To DB  To FNR  Status         Remark
       --  --------  -----  ------  ------         ------
        1     111      37     111   Inactive
       86      86      37      86   Active


           2 transactions pending:
        --------------------------

        To DB   Transactions
        -----   ------------
           37              2


            5 commands pending:
      -------------------------

      From FNR     Commands
      --------     --------
           86             5
          111             0
```

この画面には、現在定義されている Adabas-Adabas レプリケーションが表示されます。このパラメータは、Adabas-Adabas レプリケーションで Adabas Event Replicator を使用している顧客にのみ関連します。

注意: SQL データベースなどの他のレプリケーションターゲットへのレプリケーションは表示されません。このようなレプリケーションは、イベントレプリケーションの管理ツールでのみ表示できます。

ディスプレイには次の情報が表示されます。

- ［ID］は、レプリケーション管理にも使用されるレプリケーションの ID です。
- ［From FNR］は、別の Adabas ファイルにレプリケートされるファイルのファイル番号です。
- ［To DB］と［To FNR］は、レプリケーションのターゲットファイルのデータベース ID とファイル番号です。
- ［Status］には、次の値が表示されます。

ステータス	説明
Inactive	現在、ターゲットファイルにはデータがレプリケートされておらず、レプリケーションを開始するためのアクティビティも実行されていません。
準備	このステータスは、レプリケーションの初期状態処理を実行する予定であることを示します。このステータスは、パラメータ REPLICATION を指定した ADABCK を介して、レプリケートするファイルのバックアップを作成するための前提条件です。
初期化	このステータスは、パラメータ REPLICATION を指定した ADABCK が実行中であり、レプリケートされるファイルの初期状態を含むバックアップが作成されることを示します。
記録	Adabas は、現在レプリケーションコマンドファイルおよびレプリケーショントランザクションファイル内の更新トランザクションを記録していますが、現在はターゲットデータベースに更新処理をレプリケートしていません。
Active	レプリケーションがアクティブです。ソースファイルのすべての変更は、ターゲットファイルにレプリケートされます。
エラー	レプリケーション中に、不明なエラーが発生しました。レプリケーションを続けるには、新しい初期状態処理が必要です。

- ［Pending Transactions］は、ターゲットファイルにまだレプリケートされていないトランザクションの数です。

  注意:

  1. この数には、すでにコミットされていてもターゲットデータベースにまだレプリケートされていないトランザクションと、まだオープンでトランザクション終了後にのみレプリケートできるトランザクションの両方が含まれます。
  2. 複数のターゲットデータベースにレプリケートされるコマンドがトランザクションに含まれている場合、トランザクションはターゲットデータベースの数に関係なく、1回だけカウントされます。そのため、保留中のトランザクションの総数は、異なるターゲットデータベースに対するトランザクションの合計よりも少なくなる可能性があります。
- ［Pending Commands］とは、ターゲットファイルにまだレプリケートされていないコマンドの数です。

  注意:

  1. この数には、すでにコミットされていてもターゲットデータベースにまだレプリケートされていないトランザクションに属するコマンドと、まだオープンでトランザクション終了後にのみレプリケートできるトランザクションに属するコマンドの両方が含まれます。
  2. ファイルが複数のターゲットファイルにレプリケートされている場合、ソースファイルのデータベース更新コマンドは、コマンドをレプリケートする必要があるターゲットファイルの数に関係なく、1 回だけカウントされます。

ADAOPR DISPLAY=REPLICATIONS が非対話モードで実行されると、ADAOPR は次の終了ステータス値のいずれかを返します。

値	説明
0	少なくとも 1 つのレプリケーションが定義されていて、ステータスエラーになっているレプリケーションはありません。
12	ステータスエラーになっているレプリケーションがあります。
15	レプリケーションがアクティブ化されていないか、レプリケーションが定義されていません。

例：DISPLAY=RPL_STATS

```
adaopr: start_rpl_stats
adaopr: display=rpl_stats

                        ADANUC Version <version number>
        Database 6      Replication Statistics   on 18-JUL-2016 11:24:47



Replication Statistics Summary - All Times in usec
--------------------------------------------------------------------------------
Transact not yet Repl (Cur/Max)              0               2
Replicated Transactions                    281
Transact Repl Time (Avg/Min/Max)         3,055               9         171,013
Transact Latency (Avg/Min/Max)           3,173              14         171,021

Replicated Commands                      4,984
Command Repl Time (Avg/Min/Max)              1               1              18
Replicated A1 Commands                   1,536
A1 Repl Time (Avg/Min/Max)                   1               1              11
Replicated E1 Commands                   1,711
E1 Repl Time (Avg/Min/Max)                   1               1              12
Replicated NX Commands                   1,737
NX Repl Time (Avg/Min/Max)                   1               1              18
Command Wait Counter                         2
Command Wait Time (Avg/Min/Max)         15,518              41          30,995
```

注意:

1. レプリケーション統計を表示するには、START_RPL_STATISTICS コマンドでレプリケーション統計を有効化する必要があります。
2. Windows 7 で現在時刻の取得に使用されている関数の精度は 1 ミリ秒しかありません。前のコール以降にミリ秒が変化していない場合、時間は 1 マイクロ秒増加します。つまり、表示される時間値はあまり正確ではありません。値が 1000 よりもかなり小さい場合は、時間が 1 ミリ秒未満であることを示しているだけで、表示される値よりもかなり大きい可能性があります。

ディスプレイには次の情報が表示されます。

値	説明
Transact not yet Repl	コミットされてもまだレプリケートされていないレプリケーションの数。この値が大きい場合は、システムが過負荷状態であることを意味し、Adabasは更新処理を時間内にレプリケートできません。例外として大きな値でも正常な場合があります。これは、ターゲットデータベースがダウンしている場合です。この状態では、トランザクションはレプリケートできず、まだレプリケートされていないトランザクションの数が増加します。
Replicated Transactions	レプリケーション統計がアクティブ化されてからの、レプリケートされたトランザクションの数。
Transact Repl Time	1 つのトランザクションのレプリケートに必要な時間。
Transact Latency	ソースデータベース内のトランザクションがコミットされてから、レプリケートされたトランザクションがターゲットデータベースでコミットされるまでの時間。 **注意:** ターゲットデータベースがダウンしている場合、データベースが再び稼働するまで、トランザクションはレプリケーションを待機する必要があります。これは、トランザクションレイテンシの値が大きくなることを意味します。
Replicated Commands	レプリケーション統計がアクティブ化されてからの、レプリケートされたコマンドの数。
Command Repl Time	1 つのコマンドのレプリケートに要した時間。
Replicated A1 Commands	レプリケーション統計がアクティブ化されてからの、レプリケートされたA1コマンドの数。
Command A1 Repl Time	A1 コマンドのレプリケートに要した時間。
Replicated E1 Commands	レプリケーション統計がアクティブ化されてからの、レプリケートされたE1コマンドの数。
Command E1 Repl Time	E1 コマンドのレプリケートに要した時間。
Replicated NX Commands	レプリケーション統計がアクティブ化されてからの、レプリケートされたN1またはN2コマンドの数。
Command NX Repl Time	N1 または N2 コマンドのレプリケートに要した時間。
Command Wait Counter	複数のトランザクションが同時にレプリケートされた場合は、レプリケーションの一貫性を保証するために、別のトランザクションに属する別のコマンドのレプリケーションが終了するまで、コマンドのレプリケーションの待機が必要になる場合があります。カウンタには、レプリケーション統計がアクティブ化されてから、この状況がどのくらいの頻度で発生したかが表示されます。
Command Wait Time	別のトランザクションに属する別のコマンドのレプリケーションの終了を、コマンドのレプリケーションが待機する必要があった場合に、コマンドのレプリケーションを続行できるようになるまでの時間。

例：DISPLAY=STATIC_PARAMETERS

```
adaopr: display=static_parameters
                       ADANUC Version <version number>
        Database 22      Static Parameters      on 21-NOV-2014 11:13:25


Resources:          LAB       :     1,048,576    NT        :             6
                    LBP       :   104,857,600    NU        :            50
                    LWP       :     1,000,000    NCL       :            50
                    LABX      :    20,971,520
                    APU       :  ( 2,  3,  2)

Logging:            PLOG, BI
Options:            AUTO_EXPAND
```

上記の表示は、静的なニュークリアスパラメータを示しています。

注意: ニュークリアスパラメータ APU は、それが指定されている場合にのみ表示されます。

例：DISPLAY=TT

```
adaopr: display=tt
                       ADANUC Version <version number>
        Database 2           Thread Table        on 21-NOV-2014 11:49:38


 No  APU     Cmd Count  File  Cmd  Status
 --  ---     ---------  ----  ---  ------
  1    2       120,715    13   S9  Complex, waiting for DATA / 2785
  2    1       120,146    13   S8  Complex, waiting for TEMP / 35794
  3    2       124,364     0       Free
  4    1       122,300    13   S8  Complex, waiting for TEMP / 168654
  5    2       120,325    13   S8  Complex, active
  6    1       123,210    13   S1  Simple , active
```

上記の表示はスレッドテーブルのエントリを示しています。表示されるエントリ数は、同時に発生したスレッドの最大使用量です。

- ［APU］には、ニュークリアスパラメータ APUが設定されている場合に、スレッドの割り当てられた Adabas 処理ユニットが表示されます。APU が指定されていない場合は、［APU］列は表示されません。
- ［CMD COUNT］に表示される内容は、対応するスレッド環境から処理された Adabas コマンド数の合計です。これらのカウントの合計は、内部コマンドもカウントされるので、通常 DISPLAY=COMMANDS で示される合計とは異なります。

- ［FILE］に表示される内容は、対応するスレッド環境に基づいて現在処理されているAdabasコマンドのファイル番号です。対応するスレッド環境が有効でない場合、または、コマンドがどのファイルにも関連付けられていないグローバルコマンドの場合、ファイル番号は0になります。
- ［CMD］に表示される内容は、対応するスレッド環境に基づいて現在処理されているAdabasコマンドのコマンド文字列です。対応するスレッド環境がアクティブでない場合、この列には何も出力されません。
- ［STATUS］に表示される内容は、対応するスレッド環境のコマンドタイプおよびステータスです。

コマンドタイプとしては、次のものが表示されます。

- Update
- Simple
- Complex

スレッドのステータスとしては、次のものが表示されます。

ステータス	説明
free	割り当てに利用可能
ready	実行可能な状態
active	実行中
waiting for io <rabn>/<block type>	ブロック <rabn> の I/O 完了の待機
waiting for <rabn>/<block type>	ブロック <rabn> の同期のアクセス／更新の待機
waiting for space <size> bytes	ワークプールスペースの <size> バイトの待機
PLOG processing	PLOG および WORK のログエントリが作成されます。
Waiting for PLOG processing	スレッドは PLOG 処理を実行しようとしていますが、別のスレッドがすでに PLOG 処理を実行しています。複数のスレッドで同時にログエントリを作成することはできません。 **注意:** スレッドステータスのエントリは1つずつ表示されます。そのため、スレッドステータス「PLOG processing」が複数表示されたり、他のスレッドでステータス「PLOG processing」が表示されていないにもかかわらず、スレッドのステータス「Waiting for PLOG processing」が表示されたりすることがあります。

注意: スレッドステータスは、スレッドごとに１つずつ順番に表示されます。そのため、複数のスレッドでステータス「"PLOG processing"」が表示されたり、他のスレッドでステータス「"PLOG processing"」が表示されていないにもかかわらず、ステータス「"Waiting for PLOG processing"」が表示されたりすることがあります。

"block type" の値は、ASSO、DATA、WORK、FILE、または PLOG のいずれかです。

例：DISPLAY=UCB

```
adaopr: display=ucb

                      ADANUC Version <version number>
       Database 76             UCB             on 19-JAN-2014 14:59:45


      Date/Time        Entry Id  Utility  Mode Files
      ---------        --------  -------  ---- -----
19-JAN-2014 14:59:41          42   adaopr  UTO  13
```

上記の表示はユーティリティコミュニケーションブロックを示しています。

- ［DATE/TIME］には、ファイルがロックされた日付と時刻が表示されます。
- ［ENTRY ID］には、エントリに割り当てられた ID が表示されます。
- ［UTILITY］には、ユーティリティの名前が表示されます。
- ［MODE］には、ファイルがアクセスされているモードが表示されます。値は次のいずれかになります。
  - ACC：アクセス可能
  - UPD：更新可能
  - EXU：排他更新可能（同時アクセス許可）
  - UTO：ユーティリティによる使用のみ可能
  - UTI：排他アクセス可能（同時アクセスまたは更新は不許可）
- ［Files］には、ロックされているファイルの数が表示されます。

例：DISPLAY=UQ

```
adaopr: display=uq

                      ADANUC Version <version number>
       Database 76         User Queue          on 19-JAN-2014 14:58:10


       Id  Node Id   Login Id       ES Id   User Id    Type    Status
       --  -------   --------       -----   -------    ----    ------
       26  sunxxx01  dba             4473   *adaopr    UT
       23  sunxxx01  smith           3075              ET          E
       20  sunxxx01  jones           3178              ET         I
       19  sunxxx01  jones           1946              ET         IE
```

```
        18  sunxxx01  smith           4689              ET
        16  sunxxx01  smith           4661              ET
        17  sunxxx01  jones           4638   ########             T
        14  sunxxx01  miller          4379              ET      R
        13  sunxxx01  dba             3967   *adatst    AC
        12  sunxxx01  dba             3651   *adatst    EX,ET      E
        11  sunxxx01  dba             4025   DBADMIN    EX      RU

Selected: 11, Used: 11, Queue Size: 100
```

上記の表示は現在のユーザーキューエントリを示しています。

- ［ID］には内部ユーザー ID が表示されます。
- ［NODE ID］には、ノード ID 文字列が表示されます。
- ［LOGIN ID］には、ログイン ID が表示されます。
- ［ES ID］は、クライアントプロセスのプロセス ID です。

  注意: ES ID は「環境固有 ID」を意味します。この用語が使用されたのは、Windows での以前の Adabas バージョンで、同じ Adabas セッション ID の二重使用を回避するために、プロセス ID の代わりにランダムな番号が ES ID として使用されていたためです。これは、Windows の場合、プロセス ID は短時間で再利用できたことが理由です。Adabas セッション ID にタイムスタンプを追加した後は、同じ Adabas セッション ID を再利用できなくなりました。そのため、プロセス ID を Windows 上で ES ID としても使用できます。タイムスタンプは、ADAOPR DISPLAY = UQ_FULL でのみ表示されます。

- ［USER ID］には、現在の Adabas セッションのオープンコマンドのアディション 1 で指定されたユーザー ID が表示されます。

  注意: ニュークリアスオプション OPEN_REQUIRED を使用していない場合は、非アクティビティタイムアウト後にユーザー ID 情報が削除されます。このような場合、ユーザー ID は「###########」と表示されます。ニュークリアスオプション OPEN_REQUIRED を使用している場合は、ユーザー情報だけでなく、すべてのユーザーキューエレメントも削除されます。これは、DISPLAY=UQ ではそのようなユーザーキューエレメントは表示されないことを意味します。

- ［TYPE］には、ユーザータイプが表示されます。表示される値は次のとおりです。
  - AC：アクセスオンリーユーザー
  - ET：ET ユーザー
  - EX：排他更新ユーザー
  - EX,ET：ET ロジックを備えた排他更新ユーザー。
  - UT：ユーティリティユーザー。
- ［STATUS］には、ユーザーステータスが表示されます。表示される値はあ次のとおりです。

- E：ET ステータスのユーザー
- G：グローバルタイムアウト（XA）
- I：暗示的な OPEN で開始したユーザーセッション
- P：保留になっている ET（XA）
- R：制限されたファイルリスト
- T：ユーザーはタイムアウトを受け取った
- U：ユーザー固有のタイムアウト間隔の値

注意: Adabas セッション ID（ノード ID、ログイン ID、ES ID、および ADAOPR DISPLAY=UQ で表示されないタイムスタンプ）のコンポーネントの説明は、機能 lnk_set_adabas_id を使用していない場合にのみ正確です（「コマンドリファレンス」を参照）。この機能を使用すると、独自の Adabas セッション ID を定義できます。

最後の行には、現在アクティブな選択条件に基づく選択済みのユーザーキューエントリ数と、使用中の全エントリ数が表示されます。

例：DISPLAY=UQ_FILES

```
adaopr: display=uq_files

                     ADANUC Version <version number>
        Database 76         User Files          on 19-JAN-2014 14:58:10


        Id  Type   Mode Files
        --  ----   ---- -----
        26  UT
        23  ET     UPD  11-12
        20  ET     UPD  11-12
        19  ET     UPD  11-12
        18  ET     UPD  11-12
        16  ET     UPD  11-12
        14  ET     UPD  11-12
        13  AC
        12  EX,ET EXU   14
        11  EX     ACC  11
                   EXU  13

Selected: 10, Used: 11, Queue Size: 100
```

上記の表示は、アクティブなユーザーに対するファイルリストを示しています。

- ［ID］には内部ユーザー ID が表示されます。
- ［TYPE］には、ユーザータイプ（詳細については DISPLAY=UQ の例を参照）が表示されます。

- ［MODE］には、ファイルがアクセスされているモードが表示されます。表示される値は次のとおりです。
  - ACC：アクセス可能
  - EXF：排他アクセスのためのオープン（並列アクセスまたは更新は不可）
  - EXU：排他更新可能（同時アクセス許可）
  - UPD：更新可能
  - UTI：排他アクセス可能（同時アクセスまたは更新は不許可）
  - UTO：ユーティリティによる使用のみ可能
- ［FILES］には、ユーザーエントリの Adabas ファイルリストが表示されます。リストが大きすぎて1行に表示できない場合、複数行にまたがって表示されます。ファイル番号は省略されません。

最後の行には、現在アクティブな選択条件に基づく選択済みのユーザーキューエントリ数と、使用中の全エントリ数が表示されます。

例：DISPLAY=UQ_FULL

```
adaopr: disp=uq_full
                       ADANUC Version <version number>
        Database 36     Full User Queue Entry   on  3-SEP-2014 17:12:24

User Entry:  Id            :  8              ES Id        :  17937
             Node Id       :  sunada05       Login Id     :  smith
             User Id       :  *adaopr
             Timestamp Id  :   3-SEP-2014 17:12:18:182,671

             User Type     :  UT             User Status  :

Time Stamps: Session Start :  3-SEP-2014 17:12:17
             Trans. Start  :
             Last Activity :

Time Limits: TT            :             0   TNA          :             0

Resources:   ISN Lists     :             0   ISNs Held    :             0
             Open Files    :             0

Activity:    ADABAS Calls  :             1   Transactions :             0

Settings:    User Encoding :  UTF-8
-------------------------------------------------------------------------------

User Entry:  Id            :  6              ES Id        :  15808
             Node Id       :  sunada05       Login Id     :  jones
             User Id       :  JONES001
             Timestamp Id  :   3-SEP-2014 17:11:32:113,750
```

```
             User Type     :  ET             User Status  :

Time Stamps: Session Start :  3-SEP-2014 17:11:31
             Trans. Start  :  3-SEP-2014 17:11:56
             Last Activity :  3-SEP-2014 17:11:56

Time Limits: TT            :           300   TNA          :           300

Resources:   ISN Lists     :             0   ISNs Held    :             1
             Open Files    :             1

Activity:    ADABAS Calls  :             3   Transactions :             1

Settings:    User Encoding :  UTF-8
```

上記の表示は、ユーザーキューエレメントについての詳細情報を示しています。

ADAOPR DISPLAY=UQ で表示される情報に加えて、次の情報が表示されます。

- ［TIMESTAMP ID］は、Adabas セッション ID の一意性を保証するために Adabas セッション ID に追加されたタイムスタンプを表示します。
- ［Time Stamps］は、現在の Adabas ユーザーセッションが開始された時刻、セッションの最後のトランザクションが開始された時刻、およびセッションの最後のアクティビティが実行された時刻を示します。
- ［Time Limits］は、Adabas ユーザーセッションに定義された、トランザクションタイムリミットと、非アクティビティタイムリミットを示します。

  注意: 通常、このタイムリミットは ADANUC パラメータで定義されたデフォルト値ですが、Adabas ユーザーセッションのオープンコマンドでこれらのデフォルト値を上書きできます。

- ［Resources］には、Adabas ユーザーセッションで現在アクティブな ISN リストの数、セッションのホールドキュー内の ISN の数、およびセッションで使用中の Adabas ファイルの数が表示されます。
- ［Activity］には、Adabas ユーザーセッションで実行された、Adabas コールの数とトランザクションの数が表示されます。
- [Settings］には、セッションのオープンコマンドで指定された、現在の Adabas セッションで使用される W フィールドのデフォルトユーザーエンコードが表示されます。何も指定されていなかった場合は、デフォルトの UTF8 が使用されます。

例：DISPLAY=UQ_TIME_LIMITS

```
adaopr: display=uq_time_limits

                      ADANUC Version <version number>
        Database 76       User Time Limits      on 19-JAN-2010 14:58:10

TNAA Interval   :             00:15:00  TNAX Interval   :             00:15:00
TNAE Interval   :             00:15:00  TT   Interval   :             00:05:00

        Id St Limit   Timeout Interval     Remaining Time     Start Date/Time
        -- -- -----   ----------------     --------------     ---------------
        23    TNAE            00:15:00           00:15:00  19-JAN-2014 14:58:10
              TT              00:05:00
        22    TNAE            00:15:00           00:15:00  19-JAN-2014 14:58:10
              TT              00:05:00
        21    TNAE            00:15:00           00:15:00  19-JAN-2014 14:58:10
              TT              00:05:00           00:05:00  19-JAN-2014 14:58:10
        20    TNAE            00:15:00           00:15:00  19-JAN-2014 14:58:10
              TT              00:05:00           00:05:00  19-JAN-2014 14:58:10
        19    TNAE            00:15:00           00:15:00  19-JAN-2014 14:58:10
              TT              00:05:00
        18    TNAE            00:15:00           00:15:00  19-JAN-2014 14:58:10
              TT              00:05:00           00:04:50  19-JAN-2014 14:58:00
        17    TNAA            00:15:00           00:15:00  19-JAN-2014 14:58:10
        16    TNAE            00:15:00           00:15:00  19-JAN-2014 14:58:10
              TT              00:05:00           00:05:00  19-JAN-2014 14:58:10
        14    TNAE            00:15:00           00:15:00  19-JAN-2014 14:58:10
              TT              00:05:00           00:05:00  19-JAN-2014 14:58:10
        13    TNAA            00:15:00           00:10:01  19-JAN-2014 14:53:11
        12    TNAE            00:15:00           00:10:01  19-JAN-2014 14:53:11
              TT              00:05:00
        11  U TNAX            00:40:00           00:34:57  19-JAN-2014 14:53:07

Selected: 12, Used: 14, Queue Size: 100
```

上記の表示は、ユーザーキューエントリごとの現在のタイムアウト制限を示しています。

- ［ID］には内部ユーザー ID が表示されます。
- ［ST］には、エントリのステータスが表示されます。表示される値は次のとおりです。
  - U：ユーザー固有のタイムアウト値
  - T：タイムアウトが保留状態。レスポンス 9 はまだクライアントに集められていません。
- ［LIMIT］には、タイムアウトのタイプが表示されます。
- ［TIMEOUT INTERVAL］には、現在有効なタイムアウト間隔が表示されます。
- ［REMAINING TIME］には、次のタイムアウトマークまでの残り時間が表示されます。
- ［START DATE/TIME］には、エントリの開始日付および時刻が表示されます。

最後の行には、現在アクティブな選択条件に基づく選択済みのユーザーキューエントリ数と、使用中の全エントリ数が表示されます。

ES_ID

```
ES_ID = number
```

この機能は、DISPLAY オプション CQ、HQ、ICQ、UQ、UQ_FILES、UQ_FULL、および UQ_TIME_LIMITS の出力に影響します。指定された環境特定の ID を持つエントリだけが表示されます。

[NO]ET_SYNC

```
[NO]ET_SYNC
```

このオプションは、FEOF=PLOG機能の動作を制御します。FEOF=PLOGを指定する前に指定してください。詳細については、FEOF=PLOG 機能の説明を参照してください。

デフォルトは NOET_SYNC です。

[NO]EVENTING

```
[NO]EVENTING
```

実行中の adanuc プロセスの Adabas Event Analytics を開始および停止します。データベース INI ファイルの Adabas Event Analytics 設定に基づいて、adanuc プロセスがイベントの生成を開始します。Adabas Event Analytics がイベントを Analytics Server に送信するように設定している場合は、Analytics Server が起動していることを確認してください。

注意: Adabas Event Analytics を設定していない場合は、データベースディレクトリにある NUCELG ファイルに、デフォルトのイベントが書き込まれます。

デフォルトは NOEVENTING です。

EXT_BACKUP

```
EXT_BACKUP = [PREPARE | CONTINUE | ABORT]
```

この機能は、外部バックアップシステムを使用してデータベースをバックアップする場合に使用し、非常に大きなデータベースに対してADABCKを使用するよりもかなり速くバックアップを行える可能性があります。

キーワード PREPARE は、データベースのバックアップを作成するための準備をします。このフェーズ中では次の制限が適用されます。

- 新規トランザクションは停止されます。
- 更新を伴わないユーティリティの機能（ADADBM など）のみ使用可能です。

- EXT_BACKUP = PREPARE コールが処理を終了すると、SHUTDOWN、CANCEL、LOCK、STOPUSER、UNLOCK および FEOF=PLOG は使用できません。
- すべての非アクティビティタイムアウトのチェックは停止されます。

キーワードCONTINUEは、外部バックアップの完了の後に、データベースに対する通常のオペレーションを再開させます。次のような処理が実行されます。

- 新しいセッション番号で、新しい PLOG をオープンします。
- 非アクティビティタイムアウトのチェックを再開します。
- 更新を伴うユーティリティを使用可能にします。
- 待機中のすべてのユーザーをアクティブにします（新規トランザクションの開始）。

キーワード ABORT は、PREPARE がすでに発行された外部バックアップを異常終了させる場合に使用します。この場合、PLOGは切り替わりません。また、チェックポイントも書き込まれません。

注意: 外部バックアップ後に作成されたプロテクションログが外部リストアで上書きされないよう注意してください。プロテクションログがないと、ADAREC REGENERATE での外部バックアップ後に実行された変更内容を再適用できません。

例

次の例では、サードパーティのバックアップツールを使用してバックアップおよびリストアを行います（実際には tar は使用できません。説明のためにのみ使用しています）。

データベースのダンプ

```
% adaopr db=37 ext_backup=prepare
%ADAOPR-I-STARTED,      23-JAN-2015 11:49:08, Version 6.3.4.01 (Solaris 64Bit)

Database 37, startup at 22-JAN-2015 13:54:47
ADANUC Version 6.3.4.01, PID 18302

%ADAOPR-I-EXTBPREP, preparing for external backup, 23-JAN-2015 11:49:09

%ADAOPR-I-TERMINATED,   23-JAN-2015 11:49:09, elapsed time: 00:00:01
% adaopr db=37 ext_backup=continue
%ADAOPR-I-STARTED,      23-JAN-2015 11:49:20, Version 6.3.4.01 (Solaris 64Bit)

Database 37, startup at 22-JAN-2015 13:54:47
ADANUC Version 6.3.4.01, PID 18302
During ET Sync (phase 2), for external backup

%ADAOPR-I-EXTBCONT, continue from external backup, 23-JAN-2015 11:49:21

%ADAOPR-I-TERMINATED,   23-JAN-2015 11:49:21, elapsed time: 00:00:01
% adarep
```

```
adarep: checkpoints=(23-jan-2015,24-jan-2015)


Name        Date/Time        Session   User Id / Function
----        ---------        -------   ------------------
SYNX   23-JAN-2015 11:49:09       95   ADAOPR EXT_BACKUP STARTED
SYNX   23-JAN-2015 11:49:21       95   ADAOPR EXT_BACKUP
SYNC   23-JAN-2015 11:49:21       95   ADAOPR FEOF=PLOG
SYNC   23-JAN-2015 11:49:21       96   ADANUC 6.3.4.01
```

データベースのリストアとリカバリ

```
% tar xvf $BACKUPDIR/backup.tar # external restore
% mv $ADADIR/db037/plog.0096 .  # Assume current directory is not $ADADIR/db037
% adastart 37
% adarep
adarep: checkpoints=(23-jan-2015,24-jan-2015)


Name        Date/Time        Session   User Id / Function
----        ---------        -------   ------------------
SYNX   23-JAN-2015 11:49:09       95   ADAOPR EXT_BACKUP STARTED
SYNC   23-JAN-2015 13:06:41       96   ADANUC 6.3.4.01

adarep: q
%ADAREP-I-TERMINATED,   23-JAN-2015 13:07:47, elapsed time: 00:00:03
% setenv RECPLG plog.0096 # Set RECPLG for ADAREC (C shell)
% adarec dbid=37 regenerate=\* plog=96
```

リストア後、チェックポイントファイルには EXT_BACKUP=PREPARE によって書き込まれた EXT_BACKUP STARTED チェックポイントが含まれますが、EXT_BACKUP=CONTINUE によって書き込まれたチェックポイントは含まれません。最新のニュークリアスセッションに表示されるセッション番号は、外部バックアップ後に行われた変更を再適用する際に ADAREC REGENERATE で使用する必要がある最初の PLOG の番号です。

外部バックアップは、ADANUC ログファイルに記録されます

```
%ADANUC-I-DBSTART, Database 37, session 16 started, 14-NOV-2012 16:17:10
%ADANUC-I-EXTBPREP, preparing for external backup, 14-NOV-2012 16:18:30
%ADANUC-I-DBSTART, Database 37, session 17 started, 14-NOV-2012 16:18:45
%ADANUC-I-PLOGCRE, plog NUCPLG, file 'plogs/plog.0017' created
%ADANUC-I-EXTBCONT, continue from external backup, 14-NOV-2012 16:18:45
```

FEOF

```
FEOF = (keyword [,keyword])
```

指定されたキーワードに応じて、ログファイルがクローズされ、新規ログファイルが作成されます。

キーワード	説明
CLOG	コマンドログファイルのクローズ。
PLOG	プロテクションログファイルのクローズ。 [NO]ET_SYNC オプションに応じて、処理内容が異なります。 ET_SYNC を指定した場合： 現在アクティブなすべてのETロジックユーザーがETステータスになったときに、現在のプロテクションログファイル（PLOG）はクローズされ、次に大きいPLOG番号で新しいPLOGが作成されます。
ELOG	イベントログファイルのクローズ。 ELOG キーワードは、Adabas Analytics for LUW（EAL）がインストールされている場合にのみ適用できます。

ADAREC REGENERATE=* が実行されているときにキーワード PLOG を使用すると、その FEOF コマンドは拒否されます（詳細については「*ADAREC*」を参照）。

FILE

```
FILE = number
```

これは、DISPLAY オプション HQ、ICQ、UQ、UQ_FILES、UQ_FULL、および UQ_TIME_LIMITSの出力に影響します。指定のファイル番号に関連するエントリだけが表示されます。

FREE_CLQ

```
FREE_CLQ
```

通常、クライアントキューがいっぱいになると、クライアントキュー内の廃止されたエントリは自動的に解放されます。ADAOPR FREE_CLQを使用すれば、クライアントキューがいっぱいになる前に、クライアントキューを強制的にクリーンアップできます。

ID

```
ID = number
```

この機能は、DISPLAY オプション CQ、HQ、ICQ、UQ、UQ_FILES、UQ_FULL、および UQ_TIME_LIMITS の出力に影響します。指定の内部 ID に関連するエントリだけが表示されます。

[NO]IO_TIME

```
[NO]IO_TIME
```

パラメータ IO_TIME を使用して、ASSO、DATA、WORK、NUCSRT、および NUCTMP コンテナの I/O 時間をログに記録できます。時間はマイクロ秒単位で示されます。

I/O 時間のロギングがすでに有効になっているときに再び有効にすると、すべての I/O 時間と I/O カウンタ統計がリセットされます。

デフォルトは NOIO_TIME です。

ISN

```
ISN = ( number [- number] [,number [- number] ] ... )
```

この機能は、DISPLAY オプション HQ の出力に影響します。指定 ISN に関連するエントリだけが表示されます。

[UN]LOCK

```
[UN]LOCK = (number [,number]...)
```

ファイル番号で指定したファイルのロックまたはロック解除を行います。指定したファイルは、ユーティリティ以外の使用目的に対してロックされるため、Adabas ユーティリティは、ファイルを通常どおりに使用することができます。0 を指定すると、データベース全体がロック／アンロックされます。

オープンしたファイルリストの中に 1 つでもロックするファイルがあるユーザーに対しては、STOP <ユーザー ID> コマンドが内部的に発行されます。詳細については、ADAOPR STOP パラメータを参照してください。

注意:

1. 存在しないファイル番号をロックすることもできます。その後に、これらの番号を使用してファイルを作成すると、そのファイルはロックされます。
2. LOB ファイルをロックしても、ユーザーが LOB ファイルに LOB データを保存できなくなることはありません。LOB ファイル内の LOB データへのアクセスを無効にする処理は、対応す

る基本ファイルのロックの一部となっています。LOB ファイルのロックは、今後基本ファイルでこのファイル番号を使用する予定がある場合にのみ役立ちます。

3. LOCK=0 は、OPTIONS=UTILITIES_ONLY に加えてすべてのユーザーを停止することと同義です。UNLOCK=0 は OPTIONS=NOUTILITIES_ONLY と同義です。
4. ファイルがファイルレベルでロックされていた場合は、ファイルレベルでもロック解除する必要があります。UNLOCK=0 では、このようなファイルはロック解除*されません*。

LOGGING

```
LOGGING = (keyword [,keyword]...)
```

このパラメータは、キーワードリストに指定されたバッファに対するコマンドのロギングを開始するためのものです。

次のキーワードを使用できます。

キーワード	説明
CB	コントロールブロックのロギングを有効にします。
FB	フォーマットバッファのロギングを有効にします。
RB	レコードバッファのロギングを有効にします。
SB	サーチバッファのロギングを有効にします。
VB	バリューバッファのロギングを有効にします。
IB	ISN バッファのロギングを有効にします。
ABD	Adabas バッファ記述のロギングを有効にします。
IO	I/O リストのロギングを可能にします。
OFF	全バッファのロギングを停止しますが、コマンドログファイルはオープン状態を保持します。

ニュークリアスが LOGGING=OFF で起動されていて、バッファロギングの受けた場合、CLOG が作成されます。

LOGIN_ID

```
LOGIN_ID = string
```

この機能は、DISPLAY オプション CQ、HQ、ICQ、UQ、UQ_FILES、UQ_FULL、および UQ_TIME_LIMITS の出力に影響します。指定された文字列で始まるログイン ID を持つエントリのみが選択されます。文字列の仕様は大文字小文字を区別することに注意してください。8 文字未満のログイン ID を明示的に選択し、このログイン ID で始まる他のログイン ID を選択しないようにするには、ログイン ID に「^」（Windows プラットフォーム）または「\」（Windows プラットフォーム以外）を追加する必要があります。

NISNHQ

```
NISNHQ = number
```

このパラメータは、単一のユーザーによっていつでもホールド状態にできる最大レコード数を指定します。

指定された値が対応する最大値より小さい場合、警告が発行されます。

最小値は 0 で、制限がないことを意味します。

NODE_ID

```
NODE_ID = string
```

この機能は、DISPLAY オプション CQ、HQ、ICQ、UQ、UQ_FILES、UQ_FULL、および UQ_TIME_LIMITS の出力に影響します。指定した文字列で始まるノード ID を持つエントリのみが選択されます。文字列の仕様は大文字小文字を区別することに注意してください。8 文字未満のノード ID を明示的に選択し、このノード ID で始まる他のノード ID を選択しないようにするには、ノード ID に「^」（Windows プラットフォーム）または「\」（Windows プラットフォーム以外）を追加する必要があります。

OPTIONS

```
OPTIONS = (keyword[,keyword])
```

次にあげるキーワードを使用できます。

キーワード	説明
[NO]LOCAL_UTILITIES	LOCAL_UTILITIES を選択すると、すべてのリモートユーティリティコールが拒否されます。つまり、ネットワーク上にあるリモートノードからは Adabas ユーティリティを実行できなくなります。
[NO]UTILITIES_ONLY	UTILITIES_ONLY を選択すると、ユーティリティに対するコール以外のコールはすべて拒否されます。ただし、この制限は新規ユーザーにのみ適用され、OPTIONS=UTILITIES_ONLY を指定したときにすでにアクティブであったユーザーは通常の処理を続行できます。ファイルまたはデータベース全体に対する排他的なユーティリティ制御が必要な場合は、ADAOPR の LOCK 機能を使用してください。

上記のオプションは、キーワードの先頭に NO を付ける（例えば OPTIONS=NOUTILITIES_ONLY）と無効になります。

READ_PARALLEL_LIMITS

```
READ_PARALLEL_LIMITS = (records,blocks,total)
```

ニュークリアスパラメータ READ_PARALLEL_LIMITS を変更するには、このパラメータを使用します。詳細については、ADANUC の説明を参照してください。

ニュークリアスパラメータ READ_PARALLEL_LIMITS を変更するには、このパラメータを使用します。詳細については、ADANUC の説明を参照してください。

RESET

```
RESET = keyword
```

RESET=HIGH_WATER は最大使用量の値を現在使用中の値にリセットします。

RESET=COMMANDS を使用すると、ADAOPR DISPLAY=COMMANDS で表示されるコマンド数がリセットされます。

RESET=RPL_STATS は、すべてのレプリケータスレッドのレプリケーション統計カウンタをリセットします。THREAD パラメータと組み合わせて、特定のスレッドを指定することもできます。このキーワードは、Adabas-Adabas レプリケーションで Adabas Event Replicator を使用している顧客にのみ関連します。

[NO]RESPONSE_ABORT

```
[NO]RESPONSE_ABORT
```

ADAOPR の RESPONSE_CHECK パラメータでレスポンスのチェックが可能な場合、RESPONSE_ABORT オプションは、指定したレスポンスの 1 つが発生するときにニュークリアスを異常終了するかどうか（RESPONSE_ABORT）、またはニュークリアスの操作を続行してデータベースセクションファイルをディスクに書き込むかどうか（NORESPONSE_ABORT）を決定します。

[NO]RESPONSE_ABORT オプションの指定は、RESPONSE_CHECK パラメータの前にあるときだけ変更できます。XA_RESPONSE_CHECK の場合も同様です（OpenVMS には該当しません）。

デフォルトは NORESPONSE_ABORT です。

詳細については、RESPONSE_CHECK パラメータを参照してください。

RESPONSE_CHECK

```
RESPONSE_CHECK = [(number[-number][,number[-number]]...)]
```

この機能は、Adabasレスポンスコードの1つが発生したときに、DBAが情報を収集できるようにするものです。書き込まれた情報は、データベースの処理で発生する可能性のある問題を分析するために使用できます。Adabasレスポンスコードに対してレスポンスのチェックが可能な場合、このレスポンスコードが発生するとデータベースセクションファイルがディスクに書き出されます。

RESPONSE_ABORTオプションの設定に応じて、ニュークリアスは処理を終了または続行します。

- RESPONSE_ABORTオプションを設定すると、データベースセクションファイル（Adabas.xxx.hh:mm:ss（UNIX）、Adabas.xxx.hh-mm-ss（Windows）、またはAdabas-xxx-hh-mm-ss（OpenVMS））がデータベースのデフォルトディレクトリに書き込まれます。データベースセクションファイルは、CSAダンプファイルとも呼ばれます。詳細については、ADANUCおよび環境変数ADA_CSA_DUMPを参照してください。

  CSAダンプファイルが書き込まれるときに、SMPダンプファイルも書き込まれます（UNIXプラットフォームのみ）。SMPダンプファイルの名前はSMPPOS.APP:hh:mm:ssです。
- NORESPONSE_ABORTオプションを設定すると（デフォルト設定）、ニュークリアスの処理は続行され、データベースセクションファイル（Adabas.xxx.RSPyyy.hh:mm:ss（UNIX）、Adabas.xxx.RSPyyy.hh-mm-ss（Windows）、またはAdabas-xxx-RSPyyy-hh-mm-ss（OpenVMS））がデータベースのデフォルトディレクトリに書き込まれます。詳細については、ADANUCおよび環境変数ADA_CSA_DUMPを参照してください。1つのレスポンスコードに対してダンプは1つのみ生成されます。レスポンスコードが発生すると、そのレスポンスコードに対してRESPONSE_CHECKオプションは非アクティブになります。ただし、このオプションが他のレスポンスコードに対してアクティブになっている場合は、他のレスポンスコードに対してアクティブのままになります。

詳細については、RESPONSE_ABORT処理を参照してください。

デフォルトでは、レスポンスはトラップされず、ニュークリアスは処理を続行します。

レスポンスのトラップを無効にするには、引数なしでRESPONSE_CHECK =を使用してください。

SET_FILE_STATS

```
SET_FILE_STATS = [(number[-number][,number[-number]]...)]
```

この機能は、指定ファイルのファイルレベルのI/O統計を有効にします。指定したファイルだけを表示するには、DISPLAY = FILE_IO を使用します。

SHUTDOWN

```
SHUTDOWN
```

この機能は、Adabas セッションを正常終了させるものです。新規ユーザーは受け付けられません。ETユーザーの更新は、ユーザーごとの現行トランザクションが終了するまで継続されます。上記の更新処理がすべて完了すると、Adabas セッションは終了します。

データベースへのコミュニケーションリンクは切断されますが、共有メモリは維持されます。この場合、ADAOPR を使用することで表示機能は実行可能ですが、パラメータの変更はできなくなります。

STATUS

```
STATUS = (keyword [,keyword] ,... )
```

このパラメータは、DISPLAY パラメータのオプション HQ、ICQ、UQ、UQ_FILES、UQ_TIME_LIMITS、UQ_FULL の出力に影響します。指定した状態に該当するエントリだけが表示されます。

指定可能なキーワードは次のとおりです。

キーワード	説明
[NO]TIMEOUT	T ステータスを持つユーザーまたは持たないユーザー。
[NO]ET_STATUS	オープントランザクションを持つ ET ステータスのユーザー
[NO]PENDING_ET	P ステータスを持つユーザーまたは持たないユーザー

STOP

```
STOP = (number[-number][,number[-number]]...)
```

このパラメータは、指定した ID（内部 ID）のユーザーを終了させます。ID は、DISPLAY = UQ で参照できます。

メッセージ "Stop handling started for n users" が表示されます。"n" は停止されるユーザー数です。

注意: ユーティリティをこの方法で停止することはできません。

ユーザーを終了させるときに、Adabas が行う操作は、次の表に示すように、ユーザーのタイプ、ユーザーセッション開始時に明示的な OP（オープン）コマンドをニュークリアスが必要とするかによって異なります。

表中に出現する省略語 "SUQE" は "Stop user queue element"（ユーザーキューエレメントの停止）の意味で、その処理の内容は、全コマンド ID の解放、ファイルリストの削除、ユーザー ID の削除、ユーザータイプの削除、次回のコールに対するレスポンス 9 の設定となっています。

ユーザータイプ	Adabas 操作（ADANUC OPTIONS=OPEN_REQUIRED なし）	Adabas 操作（ADANUC OPTIONS=OPEN_REQUIRED あり）
ACC	ID ユーザーの場合：SUQE 非 ID ユーザーの場合：セッションのクローズ	セッションのクローズ
ET、ET ステータス	ID ユーザーの場合：SUQE 非 ID ユーザーの場合：セッションのクローズ	セッションのクローズ
ET、非 ET ステータス	トランザクションのバックアウト、SUQE	バックアウトトランザクション、セッションのクローズ
EX	SUQE、CLSE チェックポイント	セッションのクローズ
EX、ET ステータスの ET	SUQE、CLSE チェックポイント	セッションのクローズ
EX、ET、非 ET ステータス	トランザクションのバックアウト、SUQE、CLSE チェックポイント	バックアウトトランザクション、セッションのクローズ
UT	セッションのクローズ	セッションのクローズ

次の機能の実行中に、ユーザーに STOP コマンドを発行すると

```
ADAREC REGENERATE = *
```

STOP コマンドは拒否されます。

THREAD

```
THREAD = number
```

このパラメータは、Adabas を搭載した Event Replicator（Adabas レプリケーション）を使用している顧客にのみ関連します。

このパラメータを DISPLAY = RPL_STATS の前に指定すると、指定したレプリケータスレッドのレプリケーション統計のみが表示されます。スレッドの番号付けは 1 から始まります。番号なしで「THREAD=」を指定すると、後続の DISPLAY = RPL_STATS では、すべてのスレッドの統計とすべてのスレッドのサマリが表示されます。

TNAA

```
TNAA = number
```

このパラメータは、OP コマンドに TNAA 値を明示的に指定していないアクセスオンリーユーザーに対して、非アクティビティタイムリミット（秒）を設定します（詳細については『*コマンドリファレンス*』の「*OP コマンド*」を参照）。

このパラメータに指定する数値は概算値であるという点に注意してください。実際の時間は、この値から最大10秒異なっていることがあります。

最小値は 20、最大値は 2592000 です。

TNAE

```
TNAE = number
```

このパラメータは、OP コマンドに TNAE 値を明示的に指定していない ET ロジックユーザーに対して、非アクティビティタイムリミット（秒）を設定します（詳細については『*コマンドリファレンス*』の「*OP コマンド*」を参照）。

このパラメータに指定する数値は概算値であるという点に注意してください。実際の時間は、この値から最大10秒異なっていることがあります。

最小値は 20、最大値は 2592000 です。

TNAX

```
TNAX = number
```

このパラメータは、OP コマンドに TNAX 値を明示的に指定していない EXU および EXF ユーザーに対して、非アクティビティタイムリミット（秒）を設定します（詳細については『*コマンドリファレンスマニュアル*』の「*OP コマンド*」を参照）。

このパラメータに指定する数値は概算値であるという点に注意してください。実際の時間は、この値から最大10秒異なっていることがあります。

最小値は 20、最大値は 2592000 です。

TT

```
TT = number
```

このパラメータは、OP コマンドに TT 値を明示的に指定していない ET ロジックユーザーに対して、トランザクションタイムリミット（秒）を設定します（詳細については『*コマンドリファレンス*』の「*OP コマンド*」を参照）。

指定された値が対応する最大値より小さい場合、警告が発行されます。

このパラメータに指定する数値は概算値であるという点に注意してください。実際の時間は、この値から最大10秒異なっていることがあります。

最小値は 20、最大値は 2592000 です。

USER_ID

```
USER_ID = string
```

この機能は、DISPLAY パラメータのオプション CQ、HQ、ICQ、UQ、UQ_FILES、UQ_TIME_LIMITS、UQ_FULL の出力に影響します。指定した文字列で始まるユーザー ID を持つエントリのみが選択されます。文字列の仕様は大文字小文字を区別することに注意してください。8 文字未満のユーザー ID を明示的に選択し、このユーザー ID で始まる他のユーザー ID を選択しないようにするには、ユーザー ID に「^」（Windows プラットフォーム）または「\」（Windows プラットフォーム以外）を追加する必要があります。

WCHARSET

```
WCHARSET = <ICU encoding>
```

このパラメータは、ユーザーセッションの場合の W フィールドに対するデフォルトエンコードを指定します。このエンコーディングは、OP コールのレコードバッファ、または L または A/N コールのフォーマットバッファでエンコードが指定されていない場合に使用されます。

例

```
adanuc: wcharset=utf-16be
```

WRITE_LIMIT

```
WRITE_LIMIT = [number]
```

このパラメータは、暗黙のバッファフラッシュが実行されるまでに、バッファプール内で変更可能なブロックの割合を指定します。

"WRITE_LIMIT="（等号はそのまま残してその次の数値を省略）と指定しても、"WRITE_LIMIT=0" と指定しても、機能的は同じです。

最小値は 0、最大値は 70 です。0 は、Adabas がダイナミックに適切な値を選択することを意味しています。

XA_RESPONSE_CHECK

```
XA_RESPONSE_CHECK = (keyword [,keyword] ,... )
```

この機能を使用すると、XA レスポンスコードのいずれかが発生した場合に、DBA が情報を集めることができるようになります（OpenVMS には該当しません）。書き込まれた情報は、データベースの処理で発生する可能性のある問題を分析するために使用できます。XA レスポンスコードに対するレスポンスチェックが有効な場合、このレスポンスコードが発生すると、データベースセクションファイルはディスクに書き込まれます。

RESPONSE_ABORT オプションの設定に応じて、ニュークリアスは処理を終了または続行します。

- RESPONSE_ABORT オプションを設定すると、データベースセクションファイル（Adabas.xxx.hh:mm:ss）がデータベースのデフォルトディレクトリに書き込まれます。
- NORESPONSE_ABORT を設定する（デフォルト設定）と、ニュークリアスは実行を継続し、データベースセクションファイル（Adabas.xxx.XAyyyy.hh:mm:ss）がディスクに書き込まれます（詳細については ADAOPR FILE パラメータを参照）。

デフォルトでは、レスポンスはトラップされず、ニュークリアスは処理を続行します。

詳細については、RESPONSE_ABORT オプションを参照してください。

レスポンスのトラップを無効にするには、引数なしで "XA_RESPONSE_CHECK =" を使用してください。

次のキーワードがサポートされています。

XA_RBROLLBACK
XA_RBCOMMFAIL
XA_RBDEADLOCK
XA_RBINTEGRITY
XA_RBOTHER
XA_RBPROTO
XA_RBTIMEOUT

XA_RBTRANSIENT
XA_NOMIGRATE
XA_HEURHAZ
XA_HEURCOM
XA_HEURRB
XA_HEURMIX
XA_RETRY
XAER_ASYNC
XAER_RMERR
XAER_NOTA
XAER_INVAL
XAER_PROTO
XAER_RMFAIL
XAER_DUPID
XAER_OUTSIDE
XA_RBROLLBACK

詳細については、「*XA サポート*」を参照してください。

23 ADAORD（データベースまたはファイルのリオーダ、ファイルのエクスポート／インポート）

この章では ADAORD ユーティリティについて説明します。

機能概要

リオーダユーティリティ ADAORD は、データベース全体を再編成する機能（REORDER）とデータベース間でファイルを移行する機能（EXPORT/IMPORT）を提供します。

選択する機能によって、シーケンシャルファイル（ORDEXP）は、ADAORDが作成するファイルになったり、ADAORD の処理に必要なファイルになったりします。

ADAORD を実行する主な目的は、次のとおりです。

- 完全なデータベースのレイアウトを変更する。使用可能な最大ファイル番号を増やしたり、減らしたりすることも含まれます。
- スペース割り当てやファイルの配置を変更する。インデックス、アドレスコンバータ、またはデータストレージに割り当てられている論理エクステント数を減らすことも含まれます。また、パディングファクタを変更したり、再確立したりします。
- すべて同じデータを持つ1つ以上のテストファイルを作成する。このようなファイルを作成する場合には、ファイルをエクスポートしてから、別々のファイル番号を使用してインポートする必要があります。
- ファイルの元の場所がどこにあったかや、データベースに使用したデバイスの種類が何であるかに関係なく、ファイルをアーカイブし、その後でファイルを再度復元する。

データベースからファイルをエクスポートするとき、Adabas ニュークリアスは不要です。システムファイルを処理する場合、ニュークリアスは非アクティブにする必要があります。詳細については、ニュークリアス条件の表を参照してください。

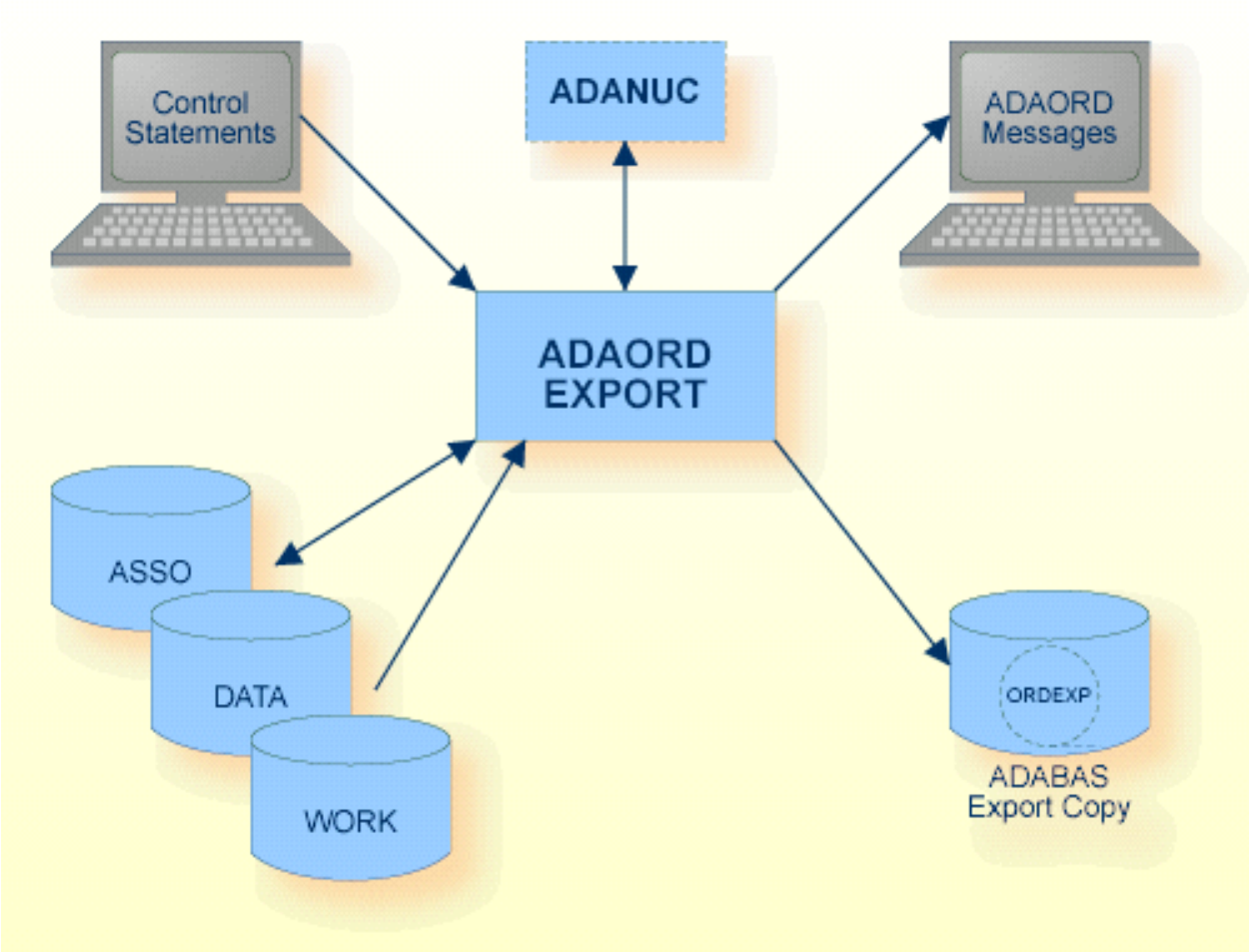
データベースにファイルをインポートするときに、Adabas ニュークリアスはアクティブである必要はありません。したがって、この手順を進めているときに、ニュークリアスを起動しても、シャットダウンしてもかまいません。

データベースをリオーダするときに、Adabas ニュークリアスがアクティブであってはいけません。

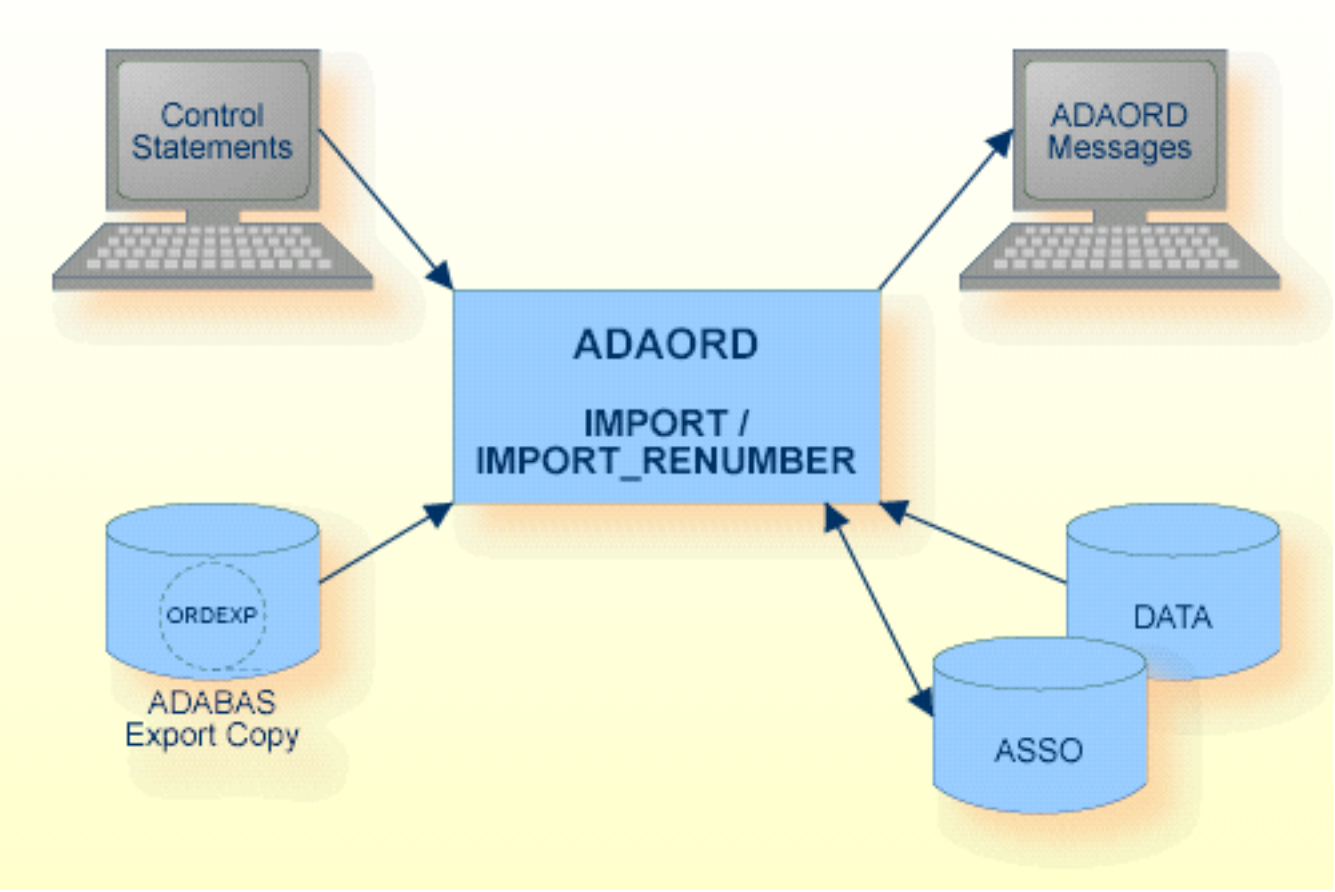
注意: IMPORT および IMPORT_RENUMBER 機能では、以前の Adabas バージョンで作成されたエクスポートファイルを処理できますが、より新しい Adabas バージョンで作成されたエクスポートファイルは処理できません。

このユーティリティは単一機能ユーティリティです。

処理フロー



シーケンシャルファイル ORDEXP は複数のエクステントを持つことができますが、RAW デバイスを使用している場合に限られます。複数のエクステントを持つシーケンシャルファイルの詳細については、『*Adabas Basics*』の「*ユーティリティの使用*」を参照してください。



シーケンシャルファイル ORDEXP は複数のエクステントを持つことができますが、RAW デバイスを使用している場合に限られます。複数のエクステントを持つシーケンシャルファイルの詳細については、『*Adabas Basics*』の「*ユーティリティの使用*」を参照してください。

データセット	環境変数／論理名	記憶媒体	追加情報
アソシエータ	ASSOx	ディスク	
データストレージ	DATAx	ディスク	
エクスポートコピー	ORDEXP	ディスク、テープ（* 注参照）	エクスポート（出力）、リオーダ（入出力）、その他の機能（入力）
コントロールステートメント	stdin/ SYS$INPUT		ユーティリティマニュアル
ADAORD メッセージ	stdout/ SYS$OUTPUT		メッセージおよびコード
ワークストレージ	WORK1	ディスク	

注意: （*）このシーケンシャルファイルには、名前付きパイプは使用できません（詳細については『*管理マニュアル*』の「*ユーティリティの使用*」を参照）。

チェックポイント

次の表は、各機能に対するニュークリアス条件と記録チェックポイントを示しています。

機能	ニュークリアスのアクティブ化が必要	ニュークリアスの非アクティブ化が必要	ニュークリアスは不要	記録チェックポイント
CONTENTS			X	-
EXPORT		X（* 注参照）	X	SYNX
IMPORT		X（* 注参照）	X	SYNP
IMPORT_ RENUMBER		X（* 注参照）	X	SYNP
REORDER		X	X	SYNP

注意: （*）Adabas システムファイルの処理時。

EXPORT 機能の場合、ADAORD は 1 つのチェックポイントを書き出し、指定したファイルすべてがエクスポートされてシーケンシャル出力ファイル（ORDEXP）がクローズすると、UCB エントリを削除します。

IMPORT 機能の場合、ADAORD はチェックポイントを書き出し、ファイルが正常にインポートされるたびにファイルがロードされたことをニュークリアスに知らせます。

UCB エントリは、指定したファイルすべてがインポートすると削除されます。ユーティリティをオフラインで実行すると、複数のチェックポイントが書き出されるので、チェックポイントブロック（CPB）オーバーフローの可能性が大きくなります。Adabas ニュークリアスを起動して CPB を空にできるように、チェックポイントファイルは常に存在していなければなりません。

データベースレベルでの REORDER 機能の場合、ADAORD は 1 つのチェックポイントを書き出し、この機能が終了すると UCB エントリを削除します。

制御パラメータ

次のコントロールパラメータを使用できます。

```
       CONTENTS

       DBID = number

       EXPORT = (number[-number][,number[-number]]...)
                [,FDT]
D               [,SORTSEQ = ({descriptor_name|ISN|PHYSICAL},...)]

       FILES = (number[[-number], number[-number]] ...)

       IMPORT = (number[-number][,number[-number]]...)
                [,ACRABN = number]
                [,ASSOPFAC = number]
                [,DATAPFAC = number]
                [,DSRABN = number]  [,DSSIZE = number[B|M] ]
                [,LOBACRABN = number]
                [,LOBDSRABN = number]
                [,LOBNIRABN = number]
                [,LOBSIZE = numberM]
                [,LOBUIRABN = number]
                [,MAXISN = number]
                [,NIRABN = number|(number,number)]
                [,NISIZE = number[B|M]|(number[B|M],number[B|M])]
                [,UIRABN = number|(number,number)]
                [,UISIZE = number[B|M]|(number[B|M],number[B|M])]

       IMPORT_RENUMBER = (number, number[,number])
                         [,ACRABN = number]
                         [,ASSOPFAC = number]
                         [,DATAPFAC = number]
                         [,DSRABN = number]  [,DSSIZE = number[B|M] ]
                         [,LOBACRABN = number]
                         [,LOBDSRABN = number]
                         [,LOBNIRABN = number]
                         [,LOBSIZE = numberM]
                         [,LOBUIRABN = number]
                         [,MAXISN = number]
                         [,NIRABN = number|(number,number)]
                         [,NISIZE = number[B|M]|(number[B|M],number[B|M])]
                         [,UIRABN = number|(number,number)]
                         [,UISIZE = number[B|M]|(number[B|M],number[B|M])]

       REORDER = *
```

CONTENTS

```
CONTENTS
```

この機能は、前回の EXPORT パラメータの実行で作成されたシーケンシャル出力ファイル（ORDEXP）内のファイルのリストを表示するものです。

DBID

```
DBID = number
```

このパラメータは、使用対象となるデータベースを選択するためのものです。

EXPORT

```
EXPORT = (number[-number][,number[-number]]...)
         [,FDT]
         [,SORTSEQ = ({descriptor_name|ISN|PHYSICAL},...)]
```

この機能は、データベースから1つ以上のファイルをシーケンシャル出力ファイル（ORDEXP）にエクスポート（コピー）するものです。エクスポートコピーで参照整合性を維持するために、指定されたファイルに参照制約を介して接続されているすべてのファイルもエクスポートされます。指定されたファイル番号は、基本ファイルのファイル番号である場合にのみ考慮されます。選択したファイルに対応する LOB ファイルは、基本ファイルとともに自動的にエクスポートされるので、指定する必要はありません。EXPORT は、各ファイルのデータストレージをそのインデックスの再構築に必要な情報とともにコピーします。処理対象のすべてのファイルは、指定された順に ORDEXP に書き出されます。重複する範囲と番号は削除されます。

注意: チェックポイントファイルがファイルリストに含まれている場合は、最後に書き出されます。

FDT

このパラメータは、処理対象のファイルの FDT を表示します。

SORTSEQ = ({descriptor_name|ISN|PHYSICAL} ,...)

このパラメータは、データストレージが処理される順序を制御します。ディスクリプタ、サブディスクリプタまたはスーパーディスクリプタのフィールド名か、キーワード "ISN" または "PHYSICAL" のいずれかを指定できます。

デフォルトは物理順です。

次の値を指定することができます。

値	シーケンス
descriptor_name	ディスクリプタ、サブディスクリプタまたはスーパーディスクリプタのフィールド名を指定する場合、データレコードはそのフィールド名が参照するディスクリプタ値の昇順の論理順に処理されます。 MU、MC、または NU オプションを持つフィールドや、このようなフィールドから派生したピリオディックグループ、サブディスクリプタ、スーパーディスクリプタに含まれるフィールドは、指定することができません。 論理順は、単一のファイルを選択した場合のみ使用できます。
ISN	ISN を指定すると、データレコードは ISN の昇順に処理されます。
PHYSICAL	PHYSICAL を指定する場合または SORTSEQ パラメータを指定しない場合、データレコードはデータストレージに格納されている物理順に処理されます。

データベースがオンラインであれば（さらにバッファプールの大きさが十分に大きい場合）、論理順および ISN 順で処理するときのパフォーマンスは向上します。

1 つの値を SORTSEQ に対して指定する場合、その値が全ファイルに有効です。複数の値を指定する場合、値の個数は EXPORT パラメータに指定されたファイル範囲の数と同じでなければなりません。この場合、最初のファイル範囲は最初に指定されたソート順でエクスポートされ、2 番目の範囲は 2 番目に指定されたソート順でエクスポートされ、次々と同様にエクスポートされます。

例

```
EXPORT = (1, 20-30, 40)
SORTSEQ = (AA, PHYSICAL, ISN)
```

ファイル 1 はディスクリプタ AA の順番でエクスポートされます。ファイル 20～30 は物理順で、ファイル 40 は ISN 順でエクスポートされます。

FILES

```
FILES = (number[[-number], number[-number]] ...)
```

このパラメータは、シーケンシャル入力ファイル（ORDEXP）に含まれているファイルを指定し、そのステータスに関する情報を表示させるために使用します。

IMPORT

```
IMPORT = (number[-number][,number[-number]]...)
         [,ACRABN = number]
         [,ASSOPFAC = number]
         [,DATAPFAC = number]
         [,DSRABN = number] [,DSSIZE = number[B|M] ]
         [,LOBACRABN = number]
         [,LOBDSRABN = number]
         [,LOBNIRABN = number]
         [,LOBSIZE = numberM]
         [,LOBUIRABN = number]
         [,MAXISN = number]
         [,NIRABN = number|(number,number)]
         [,NISIZE = number[B|M]|(number[B|M],number[B|M])]
         [,UIRABN = number|(number,number)]
         [,UISIZE = number[B|M]|(number[B|M],number[B|M])]
```

この機能は、前回のADAORDの実行で作成されたシーケンシャルファイル（ORDEXP）のデータを使用して、データベースに1つ以上のファイルをインポートします。参照整合性を維持するために、指定されたファイルに参照制約を介して接続されているすべてのファイルもインポートされます。指定されたファイル番号は、基本ファイルのファイル番号である場合にのみ考慮されます。選択したファイルに対応するLOBファイルは、基本ファイルとともに自動的にエクスポートされるので、指定する必要はありません。指定されたファイル番号は昇順でソートされます。重複する範囲と番号は削除されます。

指定されたファイル番号はデータベースにロードしないでください。

デフォルトとして、ADAORDはファイル配置や割り当て量を制御します。デフォルトを上書きするためのパラメータは、単一ファイルが選択されたときだけ使用できます。

パラメータの詳細については、**IMPORT_RENUMBER** 機能の説明を参照してください。

IMPORT_RENUMBER

```
IMPORT_RENUMBER = (number, number[,number])
                  [,ACRABN = number]
                  [,ASSOPFAC = number]
                  [,DATAPFAC = number]
                  [,DSRABN = number] [,DSSIZE = number[B|M] ]
                  [,LOBACRABN = number]
                  [,LOBDSRABN = number]
                  [,LOBNIRABN = number]
                  [,LOBSIZE = numberM]
                  [,LOBUIRABN = number]
                  [,MAXISN = number]
                  [,NIRABN = number|(number,number)]
                  [,NISIZE = number[B|M]|(number[B|M],number[B|M])]
```

```
                [,UIRABN = number|(number,number)]
                [,UISIZE = number[B|M]|(number[B|M],number[B|M])]
```

この機能は、前の ADAORD 実行で出力されたシーケンシャルファイル（ORDEXP）のデータを使用して、ファイルをデータベースにインポートします。別のファイルに参照整合性を介して接続されているファイルは、インポートして番号を変更することはできません。ファイルをエクスポートする前に制約をドロップするか、ファイルを番号変更せずにインポートして後で番号を変更する必要があります（ADADBM RENUMBER）。最初に指定された番号は、インポートする基本ファイルを定義します。2番目の番号は、そのファイルに割り当てる新しいファイル番号になります。第 3 の任意指定の数値は、LOB ファイルの新しいファイル番号です。第 3 の数値を指定しない場合、LOB ファイル番号（存在する場合）は変更されません。

新しいファイル番号をデータベースにロードしないでください。

特にそうしないという明示的な指定がない限り、ADAORD はファイル配置と割り当て量を制御します。

ACRABN = number

このパラメータは、アドレスコンバータ（AC）のスペース割り当てを開始する RABN を指定するものです。

このパラメータ指定がなければ、ADAORD が開始 RABN を割り当てます。

ASSOPFAC = number

このパラメータは、ファイルのインデックスに使用する新しいパディングファクタを指定します。ADAORD やその後で実行する一括更新ユーティリティ ADAMUP で使用されない各インデックスブロックの割合を数値で指定します。このパディングエリアは、Adabas ニュークリアスによって、将来ブロックにエントリを追加する必要が発生した場合に使用できるように予約されています。これにより、オーバーフローエントリを他のブロックに再配置する必要がなくなります。

指定可能な値の範囲は 0～95 です。

ディスクリプタの更新頻度が少ないかまったくない場合には、小さいパディングファクタ（0～10）を指定します。大きいパディングファクタ（10～50）は、新規にディスクリプタ値が作成されて、大量のディスクリプタの更新が予想される場合に指定する必要があります。

このパラメータを省略すると、ファイルのインデックスで現在有効なパディングファクタが使用されます。

DATAPFAC = number

このパラメータは、ファイルのデータストレージに使用する新しいパディングファクタを指定します。各データブロックのうち ADAORD によって使用されない部分の割合を数値として指定します。このパディングエリアは、将来Adabasニュークリアスによるレコード更新の結果として、ブロック内のレコードに追加スペースが必要になった場合に使用するため予約されます。これにより、オーバーフローエントリを他のブロックに再配置する必要がなくなります。

指定可能な値の範囲は 0〜95 です。

レコード拡張の頻度が少ないかまったくない場合は、小さいパディングファクタ（0〜10）を指定します。拡張が発生するような大量のレコード更新がある場合は、大きなパディングファクタ（10〜50）を指定します。

このパラメータを省略すると、ファイルのデータストレージで現在有効なパディングファクタが使用されます。

DSRABN = number

このパラメータは、ファイルのデータストレージ（DS）のスペース割り当てを開始する RABN を指定するものです。

このパラメータを省略した場合、ADAORD が開始 RABN を割り当てます。

DSSIZE = number[B|M]

このパラメータは、ファイルのデータストレージ（DS）に最初に割り当てられるブロック数またはメガバイト数を指定するものです。デフォルトでは、サイズはメガバイト単位で認識されます。

このパラメータを省略した場合、ADAORDは、割り当てられているブロック数の古い数値と新旧パディングファクタの差に基づいてサイズを計算します。

LOBACRABN=number

このパラメータの値は、LOBファイルのアドレスコンバータ（AC）のスペース割り当てがどこから始まるのかを示す RABN です。

このパラメータを省略した場合、ADAORD が開始 RABN を割り当てます。

LOBDSRABN=number

このパラメータの値は、LOB ファイルのデータストレージ（DS）のスペース割り当てがどこから始まるのかを示す RABN です。

このパラメータを省略した場合、ADAORD が開始 RABN を割り当てます。

LOBNIRABN=number

このパラメータの値は、LOB ファイルのノーマルインデックス（NI）のスペース割り当てがどこから始まるのかを示す RABN です。

このパラメータを省略した場合、ADAORD が開始 RABN を割り当てます。

LOBSIZE=numberM

このパラメータの値は、LOB ファイルのデータストレージ（DS）に最初に割り当てられるメガバイトの数です。LOB ファイルの AC サイズ、NI サイズ、UI サイズは、このサイズから派生します。

このパラメータを省略した場合、ADAORD は、割り当てられているブロック数の古い数値と新旧パディングファクタの差に基づいてサイズを計算します。

LOBUIRABN=number

このパラメータの値は、LOB ファイルのアッパーインデックス（UI）のスペース割り当てがどこから始まるのかを示す RABN です。

このパラメータを省略した場合、ADAORD が開始 RABN を割り当てます。

MAXISN = number

このパラメータは、ファイルに対する最大許容 ISN を指定するものです。ADAORD は、このパラメータを使用して、ファイルのアドレスコンバータ（AC）に割り当てられるスペース量を決定します。

初期割り当ての自動拡張機能がないため、ファイルの最初のフリー ISN よりも値が小さいと、アドレスコンバータに入らない ISN が存在する場合、ADAORD は実行を終了し、エラーステータスを返します。

このパラメータを指定しないと、ファイルのアドレスコンバータに対して現在有効な MAXISN の値が使用されます。

できるだけ連続領域をとるように試行されます。

NIRABN = number|(number,number)

このパラメータは、ファイルのノーマルインデックス（NI）のスペース割り当てを開始するRABN（複数可）を指定するものです。通常、Adabas は小さいインデックスブロック（ブロックサイズが 16 KB 未満の場合）には小さいディスクリプタ値（253 バイト以下）を、大きいインデックスブロック（ブロックサイズが 16 KB 以上の場合）には大きいディスクリプタ値を格納します。このため、2 つの RABN を指定することが可能です。RABN を 2 つ指定する場合、一方は 16 KB 未満のブロックサイズ、もう一方は 16 KB 以上のブロックサイズにする必要があります。

このパラメータを省略した場合、ADAORD が開始 RABN を割り当てます。

NISIZE = number[B|M]|(number[B|M],number[B|M])

このパラメータは、ファイルのノーマルインデックス（NI）に最初に割り当てられるブロック数またはメガバイト数を指定するものです。デフォルトでは、サイズはメガバイト単位で認識されます。2 つの値が指定され、NIRABN パラメータも指定された場合、最初の値は NIRABN パラメータの最初の値に対応し、2 番目の値は NIRABN パラメータの 2 番目に対応します。2 つの値が指定され、NIRABN パラメータが指定されていない場合、最初の値は小さいノーマルインデックスブロック（16 KB 未満）のサイズを指定し、2 番目の値は大きい NI ブロック（16 KB 以上）のサイズを指定します。

このパラメータを省略した場合、ADAORD は、割り当てられているブロック数の古い数値と新旧パディングファクタの差に基づいてサイズを計算します。

UIRABN = number|(number,number)

このパラメータは、ファイルのアッパーインデックス（UI）のスペース割り当てを開始するRABN（複数可）を指定するものです。通常、Adabas は小さいインデックスブロック（ブロックサイズが 16 KB 未満の場合）には小さいディスクリプタ値（253 バイト以下）を、大きいインデックスブロック（ブロックサイズが 16 KB 以上の場合）には大きいディスクリプタ値を格納します。このため、2 つの RABN を指定することが可能です。RABN を 2 つ指定する場合、一方は 16 KB 未満のブロックサイズ、もう一方は 16 KB 以上のブロックサイズにする必要があります。

このパラメータを省略した場合、ADAORD が開始 RABN を割り当てます。

UISIZE = number[B|M]|(number[B|M],number[B|M])

このパラメータは、ファイルのアッパーインデックス（UI）に最初に割り当てられるブロック数（B）またはメガバイト数（M）を指定するものです。デフォルトでは、サイズはメガバイト単位で認識されます。2つの値が指定され、UIRABNパラメータも指定された場合、最初の値はUIRABN パラメータの最初の値に対応し、2 番目の値は UIRABN パラメータの 2 番目に対応します。2つの値が指定されて、UIRABNパラメータが指定されていない場合、最初の値は小さいアッパーインデックスブロック（16KB未満）のサイズを指定し、2番目の値は大きいUIブロック（16 KB 以上）のサイズを指定します。

このパラメータを省略した場合、ADAORDは、割り当てられているブロック数の古い数値と新旧パディングファクタの差に基づいてサイズを計算します。

REORDER

```
REORDER = *
```

この機能は、データベース全体のレイアウトを変更するために使用します。データベースのグローバルエリアを再配置し、DSSTおよびファイルのアドレスコンバータ、データストレージ、ノーマルインデックス、アッパーインデックスエクステントの物理的な位置を変更することにより、それらに含まれるフラグメントを除去します。また、ファイルのパディングファクタを再設定します。データベースコンテナファイルに対する排他制御が必要です。

データベースに対する REORDER は、ファイルを暗黙的にエクスポートし、データベースから削除し、再度インポートします。REORDER の実行中に作成されたシーケンシャルファイル（ORDEXP）はそのまま残ります。

注意: ADAORD は、ファイルのディスクスペースを割り当てる最適なアルゴリズムを使用します。そのため、特定のタイプの最初のコンテナに続いて、最初のコンテナよりも小さく適切なブロックサイズを持つ別のコンテナが存在する場合、最初のコンテナが空のままになることがあります。

再スタートに関する考慮事項

ADAORD は、再スタート機能を備えていません。

EXPORT が異常終了した場合は、最初から再実行しなければなりません。

1 つ以上のファイルの IMPORT が異常終了した場合、最後にインポートしていたファイルの RABN が失われます。これらの RABN は、ADADBM の RECOVER 機能を実行することで回復できます。割り込み発生時に処理中のファイルより前のファイルは、データベース内で利用できます。このため、割り込みの発生時点のファイル番号から IMPORT 機能を再実行する必要があります。

IMPORT_RENUMBER が異常終了すると、インポートしているファイルの RABN が失われます。これらの RABN は、ADADBM の RECOVER 機能を実行することで回復できます。IMPORT_RENUMBER 機能は最初からやり直す必要があります。

データベースレベルでのREORDERが異常終了した場合、データベースのグローバル領域（GCB、FST、DSST など）の再構築時に割り込みが発生すると、データベースにアクセスできなくなります。この場合、ADAFRM を使用して新規に空のデータベースを作成するか、ADABCK の RESTORE 機能を使用して、Adabas のバックアップコピーから古いデータベースを再設定しなければなりません。再インポート時に割り込みが発生すると、最後にインポートしていたファイルの RABN が失われます。これらの RABN は、ADADBM の RECOVER 機能を実行することで回復できます。割り込み発生時に処理中のファイルより前のファイルは、データベース内で利用できます。残りのファイルは、ADAORD の IMPORT 機能を使用してシーケンシャル WORK ファイル（ORDEXP）から取得できます。

例

次の例では、データベース 1 にファイル 1、2、4、6、7、8、10、11、12、および 25 がロードされています。データベース 2 には、ファイル 3、6、11 が含まれています。

例 **1**

```
adaord: dbid = 1
adaord: export = (1-4,7,10-25)
```

ファイル 1、2、4、7、10、11、12、および 25 がデータベース 1 からエクスポートされます。

例 **2**

```
adaord: dbid = 2
adaord: import = (1-10,12)
```

ファイル1、2、4、7、10、および12がデータベース2にインポートされます。「import=(1-12)」は指定できません。ADAORDは、インポートするいずれかのファイルがすでにロードされているかどうかを先に確認し、ロードされている場合はインポート全体を拒否することが理由です。ここでは、ファイル 11 はすでにロードされています。

例 **3**

```
adaord: dbid = 2
adaord: import_renumber = (11,19), acrabn = 131, datapfac = 20
```

ファイル11は新規ファイル番号19（11はすでに使用されているため）を使用してデータベース2にインポートされます。ファイルのアドレスコンバータ（AC）は、ASSO RABN 131に割り当てられます。データストレージ（DS）の新規パディングファクタは20 %です。

例 **4**

```
adaord: dbid = 1
adaord: reorder = *
```

データベース全体がリオーダされます。

24 ADAPLP（プロテクションログプリント出力）

この章では ADAPLP ユーティリティについて説明します。

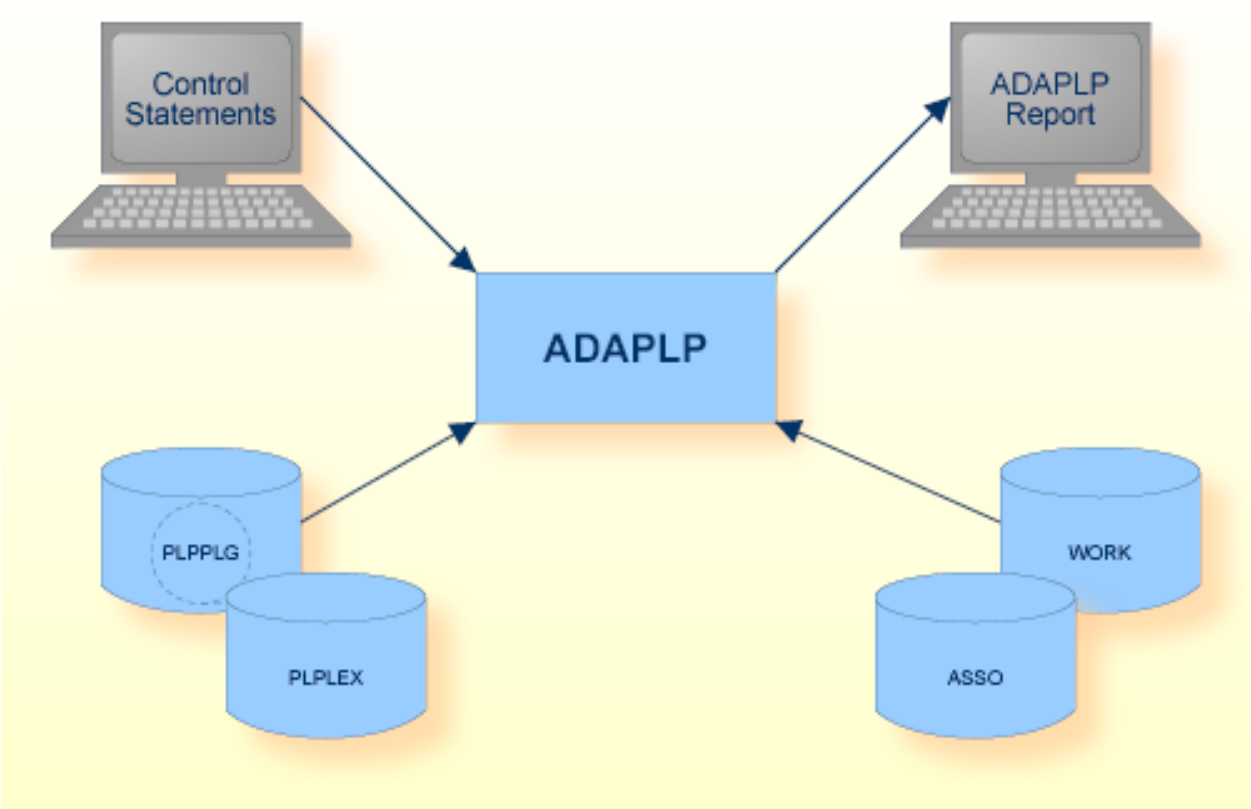
機能概要

ADAPLP ユーティリティは、プロテクションログまたは WORK のプリント出力を行います。

このユーティリティは多機能ユーティリティです。

注意: LOB 値は LOB ファイルの複数のレコードに分割されます。LOB 値をデータベースに保存するとき、プロテクションログには LOB ファイルの変更を記録したレコードが格納されます。ADAPLP では LOB 値が1つの値として表示されず、その代わりに LOB ファイルで対応するレコードの変更が表示されます。LOB ファイルレコードを圧縮解除することはできません。LOB レコードは非常に大きいために 1 ブロックに収まらないからです。連続したプロテクションログのレコードも圧縮解除できません。

処理フロー



データセット	環境変数／論理名	記憶媒体	追加情報
アソシエータ	ASSOx	ディスク	
プロテクションログ	PLPPLG	ディスク、テープ	ユーティリティマニュアル
プロテクションログ（最後のエクステント）	PLPLEX	ディスク	エクステントの数が 1 より大きい PLOG を処理する場合と、DECOMPRESS または DELTA オプションを使用する場合にのみ必要となります。処理する PLOG エクステントの 1 つ前の PLOG エクステントを指定する必要があります。
コントロールステートメント	stdin/ SYS$INPUT		ユーティリティマニュアル
ADAPLP レポート	stdout/ SYS$OUTPUT		メッセージおよびコード

データセット	環境 変数／ 論理名	記憶 媒体	追加情報
ワークストレージ	WORK1	ディスク	

シーケンシャルファイル PLPPLG は複数のエクステントを持つことができます。複数のエクステントを持つシーケンシャルファイルの詳細については、『*Adabas Basics*』の「*ユーティリティの使用*」を参照してください。

チェックポイント

このユーティリティはチェックポイントを書き込みません。

制御パラメータ

次のコントロールパラメータを使用できます。

```
M    DATASET = keyword

     DBID = number

D    [NO]DECOMPRESSED

     DELTA

D    [NO]DUMP

     FILES = (number [-number][,number [-number]]...)

D    [NO]HEADER

     INTERNAL_ID = number

     ISN = (number [,number] ... )

     MODIFIED_RABN = number

     NOFILETYPE

     NONULL

     PLOG = (number [,number])

M    RABN = {*|number[-number]}
```

```
     RECORD ={*| (number [- number]  [, number  [- number]]...) }

     SEQ = number

D    [NO]SHORT

     THREAD = number

     TSN = number

D    TYPE = (keyword [,keyword]...)

     USER_ID = string

D    [NO]WXA
```

DATASET

```
DATASET = keyword
```

このパラメータは、処理対象のプロテクションログ情報を含むファイルを選択します。キーワードに使用できる値は PLOG または WORK です。DATASET=PLOG を指定してプロテクションログが RAW デバイスに配置されている場合は、パラメータ PLOG を先に指定する必要があります。

DBID

```
DBID = number
```

このパラメータは、使用対象となるデータベースを選択するためのものです。

このパラメータは、DATASET=WORK が要求されるとき、または DATASET=PLOG でプロテクションログが RAW セクション内に書き込まれる場合に使用する必要があります。このパラメータは、最初に指定する必要があります。それ以外の場合は、このパラメータを指定しないと、PLOG に格納された DBID が使用されます。

[NO]DECOMPRESSED

```
[NO]DECOMPRESSED
```

このオプションは、プロテクションログから選択された個々のDATA レコードに対して、フィールドごとにフィールド名と圧縮解除された値を 16 進数で 1 行出力する（DECOMPRESSED）か、または出力しない（NODECOMPRESSED）かを指定するものです。

挿入レコードの場合には、挿入後のレコードのフィールド値を含むアフターイメージが表示されます。

更新レコードの場合には、更新前のレコードのフィールド値を含むビフォーイメージと、更新後のレコードのフィールド値を含むアフターイメージが表示されます。

削除レコードの場合には、削除前のレコードのフィールド値を含むビフォーイメージが表示されます。

DECOMPRESSED および NONULL が指定されている場合、次の項目は出力されません。

- NU または NC オプションが定義されているフィールドで、空値のもの
- NU または NC オプションが定義されている MU フィールドで、値がすべて空値のもの
- NU または NC オプションが定義されているフィールドのみで構成されている PE グループで、値がすべて空値のもの
- PE グループ以外のグループの名前

デフォルトは NODECOMPRESSED です。

注意: CONTINUED レコードの圧縮解除（DECOMPRESSED 出力）はできません。PLOG レコードにブロックヘッダーを加えた値が PLOG 内で 32 KB より大きい場合、または WORK に使用するブロックサイズより大きい場合、CONTINUED レコードが作成されます。

DECOMPRESSED の出力例（**NONULL** オプションなし）

```
>>> After Image <<<


Length = 20, ISN = 2

Field    : AA: ^30372E31322E3034
Group    : AB
Field    : AC: ^2020202020202020202020202020202020202020
Field    : AE: ^2020202020202020202020202020202020202020
Field    : AD: ^2020202020202020202020202020202020202020
Field    : AF: ^20
Field    : AG: ^20
Field    : AH: ^0000000C
Group    : A1
MU-field : AI, count = ^01
           AI(  1): ^2020202020202020202020202020202020202020
Field    : AJ: ^2020202020202020202020202020202020202020
Field    : AK: ^2020202020202020
Field    : AL: ^202020
Group    : A2
Field    : AN: ^202020202020
Field    : AM: ^2020202020202020202020202020
Field    : AO: ^202020202020
Field    : AP: ^2020202020202020202020202020202020202020202020202020
PE-group : AQ, count = ^01
```

```
  PE index (1)
Field    : AR: ^202020
Field    : AS: ^000000000C
MU-field : AT, count = ^01
           AT(  1): ^000000000C
End of PE-group : AQ
Group    : A3
Field    : AU: ^3030
Field    : AV: ^3030
PE-group : AW, count = ^01
  PE index (1)
Field    : AX: ^3030303030303030
Field    : AY: ^3030303030303030
End of PE-group : AW
MU-field : AZ, count = ^01
           AZ(  1): ^202020
```

DECOMPRESSED の出力例（**NONULL** オプションあり）

```
>>> After Image <<<


Length = 15, ISN = 2

Field    : AA: ^30372E31322E3034
Field    : AF: ^20
Field    : AG: ^20
```

DELTA

```
DELTA
```

このパラメータは、更新後に変更されたフィールドのみ表示します。

挿入レコードの場合には、DECOMPRESSED および NONULL オプションの出力と同じ出力になります。

更新レコードの場合には、変更されたフィールド値を含むデルタが表示されます。MU/PE フィールドの場合には、MU/PE のオカレンス数が減少した場合、表示された値の数は表示された MU/PE インデックスより少なくなることがあります。これは、すべてのフィールド値に空値が設定されたために起こります。

レコードが削除された場合、何も出力されません。しかし、CE タイプのエントリをプロテクションログに表示させる場合、削除情報を確認できます。

注意: CONTINUED レコードの圧縮解除（DELTA 出力）はできません。PLOG レコードにブロックヘッダーを加えた値が PLOG 内で 32 KB より大きい場合、または WORK に使用するブロックサイズより大きい場合、CONTINUED レコードが作成されます。

[NO]DUMP

```
[NO]DUMP
```

このオプションは、プロテクションログレコードの可変部分がプリント出力内に含まれる（DUMP）か、含まれない（NODUMP）かを指定するものです。

DUMP を指定した場合には、プロテクションログレコードの可変部分は 16 進形式および解釈不能 ASCII 形式の両方で表示されます。

DUMP を指定すると、SHORT はリセットされます。

デフォルトは NODUMP です。

FILES

```
FILES = (number [-number][,number [-number]]...)
```

プロテクションログレコードは、このパラメータで指定されたファイルに属している場合にのみ表示されます。

タイプ DA、DV、EXT、INDEX および FCB のレコードだけが表示されます。

プロテクションログレコードのタイプについては、このセクションの最後にある表を参照してください。

[NO]HEADER

```
[NO]HEADER
```

このオプションは、プロテクションログのブロックごとにヘッダーを表示する（HEADER）か、表示しない（NOHEADER）を指定するものです。

デフォルトは HEADER です。

INTERNAL_ID

```
INTERNAL_ID = number
```

このオプションを使用すると、指定した内部 ID を持つレコードだけが表示されます。

ISN

```
ISN = (number [,number] ... )
```

プロテクションログレコードが、このパラメータで指定する ISN に属する場合だけ表示されます。タイプDA とDVのレコードだけが表示されます。プロテクションログレコードのタイプについては、このセクションの最後にある表を参照してください。このパラメータは、FILE パラメータと組み合わせたときにだけ使用できます。

MODIFIED_RABN

```
MODIFIED_RABN = number
```

このオプションは、指定した RABN に対する修正が記録されたレコードだけを表示します。

プロテクションログレコードのタイプについては、このセクションの最後にある表を参照してください。

NOFILETYPE

```
NOFILETYPE
```

このキーワードは、ファイル番号と関連付けられているレコードタイプだけでなく、ファイル番号とは無関係なレコードタイプ（ET や BT など）を表示する場合に使用します。

例

```
adaplp dbid=6 file=25 type=(da,dv,et,bt) nofiletype
```

ファイル 25 の DA および DVT レコードに加えて、ET および BT レコードも表示させます。

NONULL

```
NONULL
```

DECOMPRESSED パラメータと一緒に指定した場合にだけ、NONULL パラメータを指定する意味があります。詳細については、**DECOMPRESSED** パラメータを参照してください。

PLOG

```
PLOG = (number[,number])
```

DATASET=PLOGが指定され、プロテクションログがRAWセクション内に書き込まれる場合、このパラメータは必須です。プロテクションログがファイルシステム内に書き込まれる場合には、指定するかどうかは任意です。PLOG番号とエクステンションカウントを指定することができます。エクステンションカウントを指定すると、指定エクステントだけが処理されます。エクステンションカウントを指定しない場合、Adabasは必要に応じて後続のエクステントをオープンします。パラメータ PLOG は DATASET=PLOG を指定する前に指定しなければなりません。

例：

```
Section layout

          .
          .
          .
250000 260000  10001  30  PLG.36 created

377000 378000   1001  30  PLG.36(3) created

adaplp: plog=36
adaplp: dataset=plog
```

PLG.36 がオープンされます。

```
adaplp: plog=(36,3)
adaplp: dataset=plog
```

PLG.36(3) がオープンされます。

RABN

```
RABN ={*| number [- number] }
```

このパラメータは、WORK またはプロテクションログファイルのブロックを 1 つずつ指定するか、または連続した範囲で指定します。指定したブロック内の内容が表示されます。

アスタリスク（*）を指定すると、全ブロックが表示されます。

注意:

1. RABNを指定せずにADAPLPを起動した場合、ユーティリティは実行しますが、何も出力されません。

2. RABN パラメータを指定すると、要求された出力が直ちに生成されます。そのため、出力生成に必要な他のすべてのパラメータ（RABN パラメータの前の DATASET パラメータなど）を指定する必要があります。

例

```
adaplp: rabn = 123

adaplp: rabn = 123 - 1246
```

RECORD

```
RECORD ={*| (number [- number]  [, number  [- number]]...) }
```

このパラメータは、出力対象となるレコードまたはレコードの範囲を選択するものです。アスタリスク（*）が指定されるか、または何も指定されない場合には全レコードが出力の対象となります。

例：

```
adaplp: record = (2-5,9,11)
```

レコード 2、3、4、5、9 および 11 が 1 つまたは複数 PLOG ブロックの出力時に書き出されます。

SEQUENCE

```
SEQUENCE = number
```

このパラメータは指定したシーケンス番号で書かれたレコードだけを表示します。

[NO]SHORT

```
[NO]SHORT
```

このオプションは、プロテクションログブロックのヘッダーだけを表示対象とする（SHORT）か、または全レコードを表示対象とする（NOSHORT）かを示すものです。

SHORT を指定すると、DUMP はリセットされます。

デフォルトでは、プロテクションログブロックのヘッダーの後に、そのブロック内のすべてのレコードが表示されます。

デフォルトは NOSHORT です。

THREAD

```
THREAD = number
```

このオプションは、指定スレッドのレコードだけを表示します。

TSN

```
TSN = number
```

このオプションは、指定したトランザクションシーケンス番号（TSN）を持つレコードだけを表示します。

タイプ BT、C5、CL、DA、DV、ET および XA（OpenVMS にはない）のレコードだけが表示されます。

プロテクションログレコードのタイプについては、このセクションの最後にある表を参照してください。

TYPE

```
TYPE = (keyword [,keyword]...)
```

このオプションは、所定のキーワード（複数可）で指定したプロテクションログレコードだけを表示します。各キーワードは次の表のように1つ以上のプロテクションログレコードのタイプに対応します。

キーワード	プロテクションログレコードのタイプ
AB	AB
ASSO	レコードタイプ AC、およびキーワード EXT、FCB、INDEX で選択された全レコードタイプ
AT	AT
BF	BS、BE、BF
BT	BT
C1	C1
C5	C5
CE	CE
CF	CF
CT	CT
DA	DA
DATA	キーワード BT、CE、DA、DV、ET および OP で選択された全レコードタイプ
DC	DC
DT	DT
ET	ET、CL

キーワード	プロテクションログレコードのタイプ
EXT	ACEXT、UIEXT、NIEXT、DSEXT
FCB	FCBDS、FCBIX、SPISN
INDEX	FE、INDEX、IB、INSRU、REMRU
OP	OP
XA	キーワード YB、YD、YF および YP で選択された全レコードタイプ
YB	YB（OpenVMS にはない）
YD	YD（OpenVMS にはない）
YF	YF（OpenVMS にはない）
YP	YP（OpenVMS にはない）

プロテクションログレコードのタイプについては、このセクションの最後にある表を参照してください。

デフォルトでは、すべてのプロテクションログレコードのタイプが表示されます。

USER_ID

```
USER_ID = string
```

このパラメータは、指定ユーザー ID で開始したレコードだけを表示します。

タイプ BT、C1、C5、CL、DA、DV、ET、FCBDS、FCBIX、INDEX および XA のレコードだけが表示されます。

プロテクションログレコードのタイプについては、このセクションの最後にある表を参照してください。

[NO]WXA

```
[NO]WXA
```

このオプションは、WORK パート 1 リングバッファ（NOWXA）および WORK パート 1 XA エリア（WXA）を切り替えます。

デフォルトは NOWXA です。

ADAPLP 出力

プロテクションログまたはWORKの各ブロックはヘッダーの後に続けて出力され、次の情報から構成されます。

- ブロックのシーケンス番号
- ブロックのサイズ
- ブロックが属するセッションの番号（PLOG 番号と同じ）
- ブロックの作成時刻を示すタイムスタンプ（WORK の内部タイムスタンプ）

1 レコードの出力は次のエントリから構成されます。

- 1 レコードのシーケンス番号（ブロックごとに 1 から始まる）
- レコードの内部長
- コマンドのシーケンス番号（コマンドを一意に識別する）
- PLOG レコードのタイプ（詳細については次の一覧を参照）
- コマンドを実行したスレッドの番号

これ以外にも、ほとんどのレコードで次のエントリも出力されます。

- 内部ユーザー ID（16 進表示）。1 トランザクションをオープンする各コマンドに対して一意に割り当てられます。

PLOG レコードのタイプの一覧を次に示します。

型	説明
AB	WORK ラップラウンドを記録します（WORK のみ）。
AC	バックアウトトランザクション中にレコードの再配置を記録します（WORK のみ）。
ACEXT	アドレスコンバータの拡張を記録します（WORK のみ）。
AT	フィールドの追加を記録します（ADADBM）。
BE	バッファフラッシュの終了を記録します（WORK のみ）。
BF	バッファフラッシュの開始および終了を記録します（WORK のみ）。
BS	バッファフラッシュの開始を記録します（WORK のみ）。
BT	BT 処理の開始を記録します。
C1	C1 コマンドからのレコードを記録します。チェックポイント名を含みます（PLOG のみ）。
C5	C5 コマンドからのレコードを記録します（PLOG のみ）。
CE	コマンドの最終エントリ（シーケンス番号を持つ最終エントリ）を示します。コマンドが削除操作だった場合、削除レコードのファイル番号と ISN が表示されます。 例

型	説明
	`>>> DELETE FILE 10 ISN 2 <<<`
CF	FDT の作成を記録します（ADAFDU）。
CL	ユーザーの CLOSE を記録します。
CT	ファイルの作成を記録します（ADAFDU）。
DA	データレコードの変更を記録します。ファイル、RABN、データレコードの ISN が表示されます。レコードはアフターイメージ（AI）、ビフォーイメージ（BI）、デルタイメージ（DI）のいずれかであり、DUMP が有効であるときに表示されます。TSN は内部トランザクションシーケンス番号です。あるトランザクションから派生するエントリはどれも、TSN が同じです（『コマンドリファレンスマニュアル』の「ET コマンド」も参照）。WB の出力は、DATASET=WORK が指定されている場合にだけ表示され、同じ TSN を持つ以前の PLOG レコードを見つけることのできる WORK ブロックを示します。clu の値がゼロでない場合、その値は排他ユーザーまたは特権ユーザーを示します。
DC	フィールドの削除を記録します（ADADBM）。
DSEXT	データストレージの拡張を記録します（WORK のみ）。
DT	ファイルの削除を記録します（ADADBM）。
DV	ディスクリプタの更新を記録します（必ず DA レコードの後にあります）。ファイルのエントリ、ISN、TSN、clu、および WB は DA レコードタイプのものと同様です。
ET	ET コマンドからのエントリを記録します。ET TSN と指定すると、ET コマンドで書かれた最終ユーザーデータに TSN が設定されます。
FCBDS	データストレージに対する FCB 変更を記録します（WORK のみ）。
FCBIX	ノーマルインデックスに対する FCB 変更を記録します（WORK のみ）。
FE	インデックスブロックの最初のエントリ変更を記録します（WORK のみ）。
IB	修正されたインデックスブロックを記録します（WORK のみ）。
INDEX	分割されたインデックスブロックを記録します（WORK のみ）。
INSRU	再利用チェーンへのインデックスブロックの挿入を記録します（WORK のみ）。
NIEXT	ノーマルインデックスの拡張を記録します（WORK のみ）。
OP	ユーザーの OPEN を記録します。
REMRU	再利用チェーンからのインデックスブロックの削除を記録します（WORK のみ）。
SPISN	ISN 再利用またはスペース再利用での変更を記録します。
UIEXT	アッパーインデックスの拡張を記録します（WORK のみ）。
YB	XA プロトコル内で実行された、トランザクションのバックアウトを記録します（OpenVMS にはない）。
YD	XA プロトコルでヒューリスティックに終了した、トランザクションの破棄を記録します（OpenVMS にはない）。
YF	XA プロトコルで実行された、トランザクションの最終コミットを記録します（OpenVMS にはない）。
YP	XA プロトコルで実行された、トランザクションの予備コミットを記録します（OpenVMS にはない）。

次に示すように、DA レコードまたはDV レコードの場合に、表示されることがあるフラグもあります。

フラグ	説明
AI	この PLOG レコードのデータにはデータレコードのアフターイメージが含まれます（レコードタイプ DA）。
BACKOUT	レコードは単一コマンド内のバックアウト時に書き出されたことを示します。
BI	この PLOG レコードのデータにはデータレコードのビフォーイメージが含まれます（レコードタイプ DA）。
BT	レコードは 1 トランザクションのバックアウト時に書き出されたことを示します。
DI	このPLOG レコードのデータにはデータレコードのデルタイメージが含まれます（レコードタイプ DA）。
FDATA	このコマンドの最初の DA レコードであることを示します。
FIRST_ENTRY	指定したシーケンス番号を持つ最初のレコードであることを示します。
HIMERGE	最高インデックスレベルのマージ。
HISPLIT	最高インデックスレベルの分割。
USERD	トランザクションはユーザーデータを処理します。

25 ADAPRI（Adabas ブロックの出力）

この章では ADAPRI ユーティリティについて説明します。

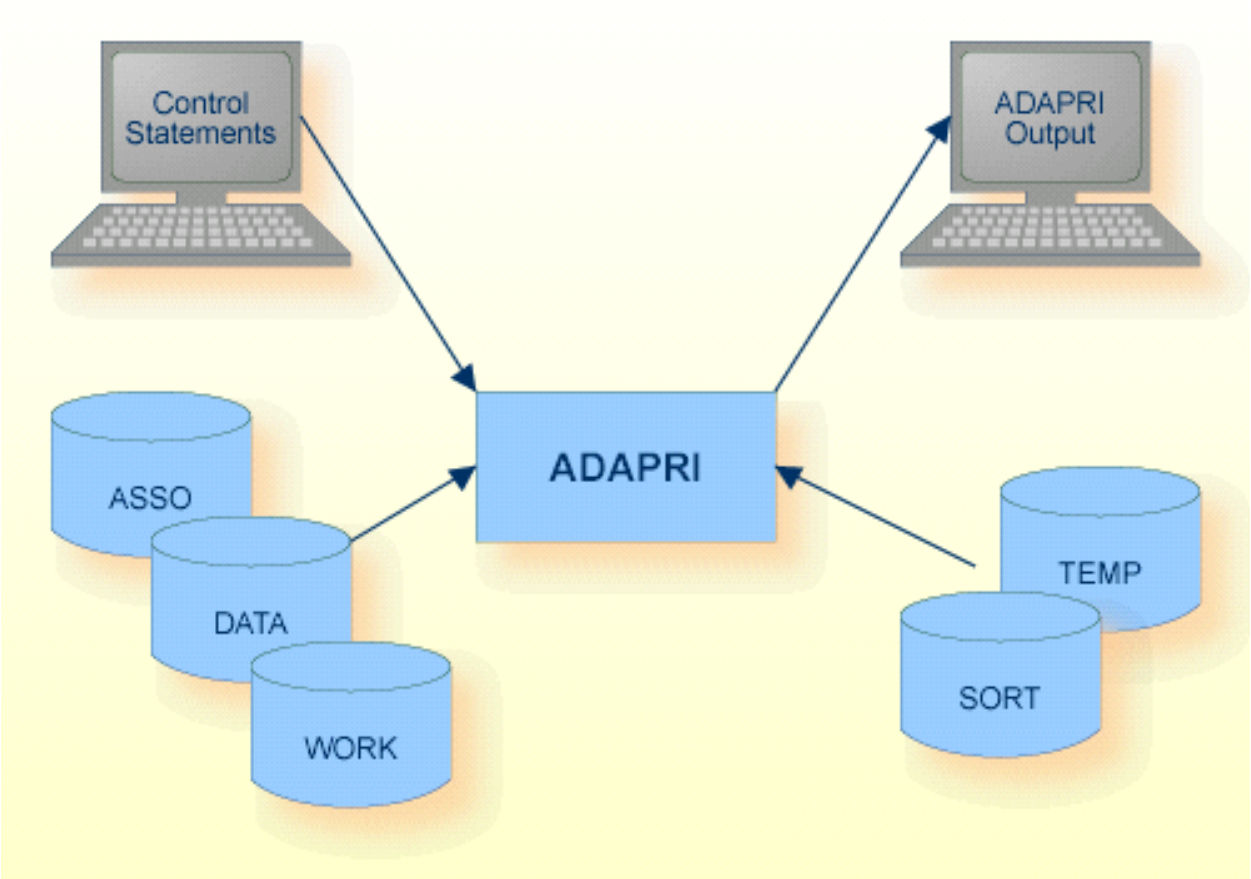
機能概要

ADAPRI ユーティリティは、メンテナンスまたは監査目的で、アソシエータ、データストレージ、WORK、TEMP、または SORT 内のブロックまたはブロック範囲の内容を出力します。

出力は、16 進形式と ASCII 形式で行われます。同じ行やブロックが続く場合、それらは省略されます。

このユーティリティは多機能ユーティリティです。

処理フロー



データセット	環境変数／論理名	記憶媒体	追加情報
アソシエータ	ASSOx	ディスク	
データストレージ	DATAx	ディスク	
ソートストレージ	SORTx	ディスク	
コントロールステートメント	stdin/ SYS$INPUT		ユーティリティマニュアル
ADAPRI 出力	stdout/ SYS$OUTPUT		
一時ストレージ	TEMPx	ディスク	
ワークストレージ	WORK1	ディスク	

DATA または WORK コンテナファイルを処理するには、ASSO コンテナファイルに対する割り当てが必要です。

チェックポイント

このユーティリティはチェックポイントを書き込みません。

制御パラメータ

次のコントロールパラメータを使用できます。

```
DATASET = keyword

DBID = number

RABN = number [- number]
```

DATASET

```
DATASET = keyword
```

このパラメータは、ダンプ出力対象となるデータベースの部分を指定するためのものです。使用できるキーワードは次のものです。

キーワード	説明
ASSO	アソシエータ
DATA	データストレージ
SORT	ソートエリア
TEMP	一時的な領域
WORK	ワークエリア

例

```
adapri: dataset = asso, rabn = 123 - 321
```

RABN 123 から RABN 321 までのアソシエータがダンプ出力されます。

DBID

```
DBID = number
```

このパラメータは、使用対象となるデータベースを選択するためのものです。

このパラメータは、DATASET=TEMP または DATASET=SORT パラメータ指定のときは必要ありません。

RABN

```
RABN = number [- number]
```

このパラメータは、ダンプ対象となる RABN または RABN の範囲を指定するものです。

例

```
adapri: dbid = 1, dataset = data, rabn = 123
```

データベース 1 の DATA RABN 123 がダンプ対象となります。

```
adapri: dataset = sort, rabn = 123 - 129
```

データセット SORT の RABN の 123 から 129 までがダンプ対象となります。

26 ADARBA（RBAC 管理）

この章では「ADARBA」ユーティリティについて説明します。

機能概要

ADARBA ユーティリティは、RBAC セキュリティ定義の管理に使用されます。RBAC セキュリティ定義は、データベースの RBAC システムファイルに保存されています。

ADARBA は、ユーザーや役割などの基本的なセキュリティオブジェクトの作成と変更、および権限の付与と取り消しに使用します。詳細については、『*管理マニュアル*』の「*Adabas ユーティリティの認可*」を参照してください。

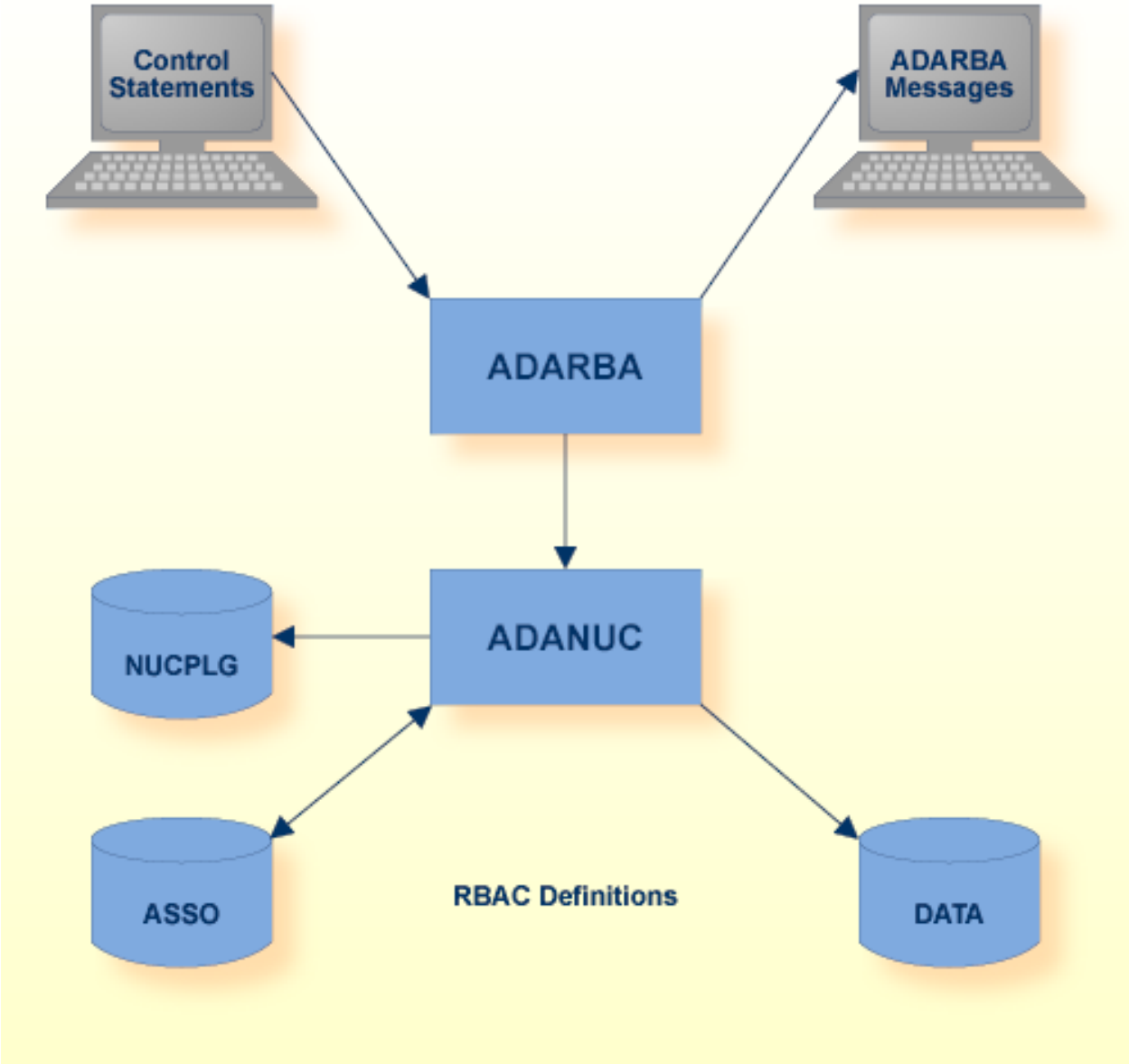
使用するデータベースはオンラインになっている必要があります。

注意: 各 ADARBA コマンドはトランザクションとなります。これは、セキュリティ定義への変更が直ちに有効になることを意味します。

重要: このユーティリティへのアクセスは、データベースセキュリティに責任を負う人員に厳密に制限すべきです。

このユーティリティは多機能ユーティリティです。

処理フロー


Control Statements
ADARBA Messages
ADARBA
ADANUC
NUCPLG
ASSO
RBAC Definitions
DATA

データセット	論理名	記憶媒体	追加情報
RBAC 定義		データベース／RBACシステムファイル	
コントロールステートメント	stdin/SYS$INPUT		ユーティリティマニュアル
ADARBA メッセージ	stdout/SYS$OUTPUT		メッセージおよびコード

チェックポイント

このユーティリティはチェックポイントを書き込みません。

制御パラメータ

次のコントロールパラメータを使用できます。

```
     CREATE  ,{OPERATION|USER|OBJECT|ROLE} = string
M    DBID = number
     DROP ,{OPERATION|USER|OBJECT|ROLE} = string
     [NO]ECHO
     GRANT ,ROLE = string [,TO] ,USER = string
     GRANT ,OPERATION = string [,OBJECT = string] [,TO]  ,ROLE = string
     LIST  ,{OPERATION|USER|OBJECT|ROLE} [= string]
     LIST  ,ASSIGNMENT, {USER|PERMISSION}
     REVOKE ,ROLE = string [,FROM] ,USER = string
     REVOKE ,OPERATION = string [,OBJECT = string] [,FROM] ,ROLE = string
     [NO]STAT
```

CREATE

```
CREATE ,{OPERATION|USER|OBJECT|ROLE} = string
```

この機能は、指定されたタイプと値で RBAC 定義を作成します。

値の大文字と小文字は区別されます。

USER タイプの項目に割り当てられる値は、有効なログオン資格情報（ユーザー ID など）である必要があります。これらの値はプラットフォーム固有です。

- Unix/Linux：user_identification
- Windows： domain\user_identification

詳細については、『*管理マニュアル*』の「*Adabas* ユーティリティの認可」を参照してください。

例：

```
adarba: create,user=domain\userid
```

Windows 用のユーザー定義 domain\userid が作成されます。

DBID

```
DBID = number
```

このパラメータは、使用対象となるデータベースを選択するためのものです。

注意: ニュークリアスは実行させる必要があります。

例：

```
adarba: dbid=200
```

現在使用されているデータベースはデータベース 200 です。

DROP

```
DROP ,{OPERATION|USER|OBJECT|ROLE} = string
```

この機能は、指定されたタイプと値を持つ RBAC 定義を削除します

例：

```
adarba: drop,user=NEWUSER
```

ユーザー定義 NEWUSER が削除されます。

[NO]ECHO

```
[NO]ECHO  ↩
```

この機能は、コマンド入力のエコーのオンとオフを切り替えます。

例：

```
adarba: echo
%ADARBA-I-INP, echo
%ADARBA-I-PAR, echo input enabled
```

この ADARBA セッションでは、エコー入力が有効になります。

GRANT（ユーザーの割り当て）

```
GRANT ,ROLE = string [,TO] ,USER = string
```

この機能は、ユーザーに役割を付与します。

例：

```
adarba: grant,role=NEWROLE,to,user=NEWUSER
```

ユーザー NEWUSER に、ロール NEWROLE が割り当てられます。

GRANT（権限の割り当て）

```
GRANT ,OPERATION = string [,OBJECT = string] [,TO]  ,ROLE = string
```

この機能は、オブジェクトに対して特定の処理を実行する権限を役割に付与します。

例：

```
adarba: grant,operation=ada.uti.opr,to,role=ANYROLE
```

役割 ANYROLE に対して、デフォルトオブジェクト（DBID.CURRENT）に ada.uti.opr 処理を実行する権限が割り当てられます。

LIST

```
LIST ,{OPERATION|USER|OBJECT|ROLE} [= string]
```

この機能は、文字列値が指定されていて、指定された定義が存在する場合に、RBAC定義を表示します。

値が何も指定されていない場合、この機能は、指定されたタイプのアクティブな RBAC 定義をすべて表示します。

例：

```
adarba: list,role=PUBLIC
PUBLIC
```

役割 PUBLIC が表示されます。

```
adarba: list,role=
PUBLIC
```

唯一のアクティブな役割定義となっている役割 PUBLIC が表示されます。

LIST ASSIGNMENT

```
LIST ,ASSIGNMENT ,{USER|PERMISSION}
```

この機能は、指定されたタイプに応じて、すべてのアクティブなユーザーまたは権限の割り当てを表示します。

例：

```
adarba: list,assignment,user
PUBLIC,PUBLIC
```

すべてのユーザー割り当てが表示されます。

REVOKE（ユーザーの割り当て）

```
REVOKE ,ROLE = string [,FROM] ,USER = string
```

この機能は、ユーザーに付与された役割を取り消します。

例：

```
adarba: revoke,role=NEWROLE,from,user=NEWUSER
```

ロール NEWROLE が、ユーザー NEWUSER から取り消されます。

REVOKE（権限の割り当て）

```
REVOKE ,OPERATION = string [,OBJECT = string] [,FROM] ,ROLE = string
```

この機能は、役割に付与されていた、オブジェクトに対して処理を実行する権限を取り消します。

例：

```
adarba: revoke,operation=ada.uti.dbm,from,role=NEWROLE
```

役割 NEWROLE に割り当てられていた、デフォルトのオブジェクト DBID.CURRENT に ada.uti.dbm 処理を実行する権限が取り消されます。

[NO]STAT

```
[NO]STAT
```

この機能は、コマンド統計の有効／無効を切り替えます。

例：

```
adarba: stat
%ADARBA-I-INP, stat
%ADARBA-I-PAR, command statistics enabled
```

この ADARBA セッションのコマンド統計が有効になります。

27 ADAREC（データベースまたはファイルのリカバリ）

この章では ADAREC ユーティリティについて説明します。

機能概要

ADAREC ユーティリティには、次のようなデータベースリカバリ機能が含まれます。

- CLOSE 機能は、ディスクセクション内の異常終了したプロテクションログファイルにエンドオブファイルを明示的に書き込みます（UNIX のみ）。
- LIST 機能は、プロテクションログに関する情報をリストします。
- REGENERATE 機能は、指定した 2 つのチェックポイント間で行われた更新をすべて再適用します。使用されるチェックポイントは、通常、チェックポイントコマンド（C1）の結果ですが、EXU ユーザーまたはユーティリティ処理からの OP コマンドによって取得された内部チェックポイントでも構いません。データベース全体を再生成する場合、EXCLUDE_FILES オプションを使用して一部のファイルを除外することができます。このオプションとともに指定したファイルは再生成されず、除外された更新がレポートされます。

REGENERATE 機能が SYNP チェックポイント時点で終了すると、ADAREC はこの PLOG から、次回の実行時の再開位置が他にもあるか見つけるために、現在の PLOG を検索します。その後、ADAREC の再開前に実行しなければならない他のユーティリティ機能のリストが表示されます。1 つまたは複数の SYNP チェックポイントが検出された場合、ADAREC は次のように終了します。

- PLOG に ADAREC の再起動によって適用される追加トランザクションが含まれている場合は、出口コード 14 で終了します。
- それ以外の場合は、出口コード 12 で終了します。

算出された再開位置は、BLOCK= または CHECKPOINT= を入力することによってリセットまたは変更することができます。使用可能なシステムのチェックポイントのタイプについては、このマニュアルのデータベースレポートユーティリティ ADAREP の説明を参照してください。

通常、REGENERATE 機能は全ログを終了し、トランザクションの確認が行われると終了します。この機能は、ADABCK ユーティリティの RESTORE 機能によってデータベース（または 1 つ以上のファイル）を以前の状態に戻した後でかなり頻繁に使用されます。

エラーファイルにレコードが書き込まれると、ユーティリティはゼロ以外のステータスで終了します。

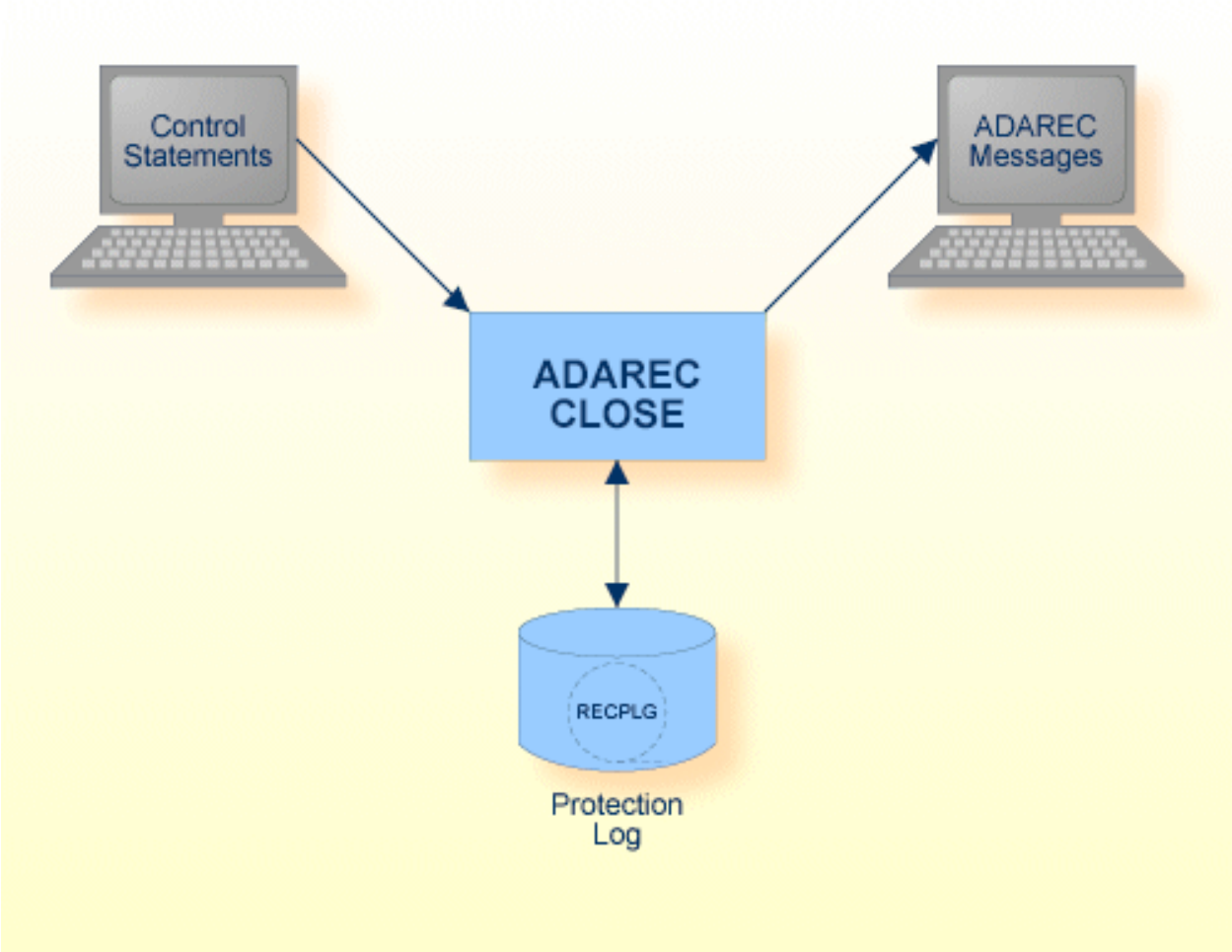
注意:

1. ADAREC を使用して複数のファイルを同時に再生成する場合、最初にニュークリアスパラメータ LBP の値を大きくしてください。これは、ADAREC がデータベースに対して大量の更新を行うためです。LBP の値が十分大きくない場合、Adabas はレスポンスコード 162 を返します。

2. 出口コード 12 は、バージョン 6.3 SP2 で導入されました。これまでの Adabas のリリースでは、SYNP チェックポイントが検出されたときは、常に出口コード 14 で終了していました。

このユーティリティは単一機能ユーティリティです。

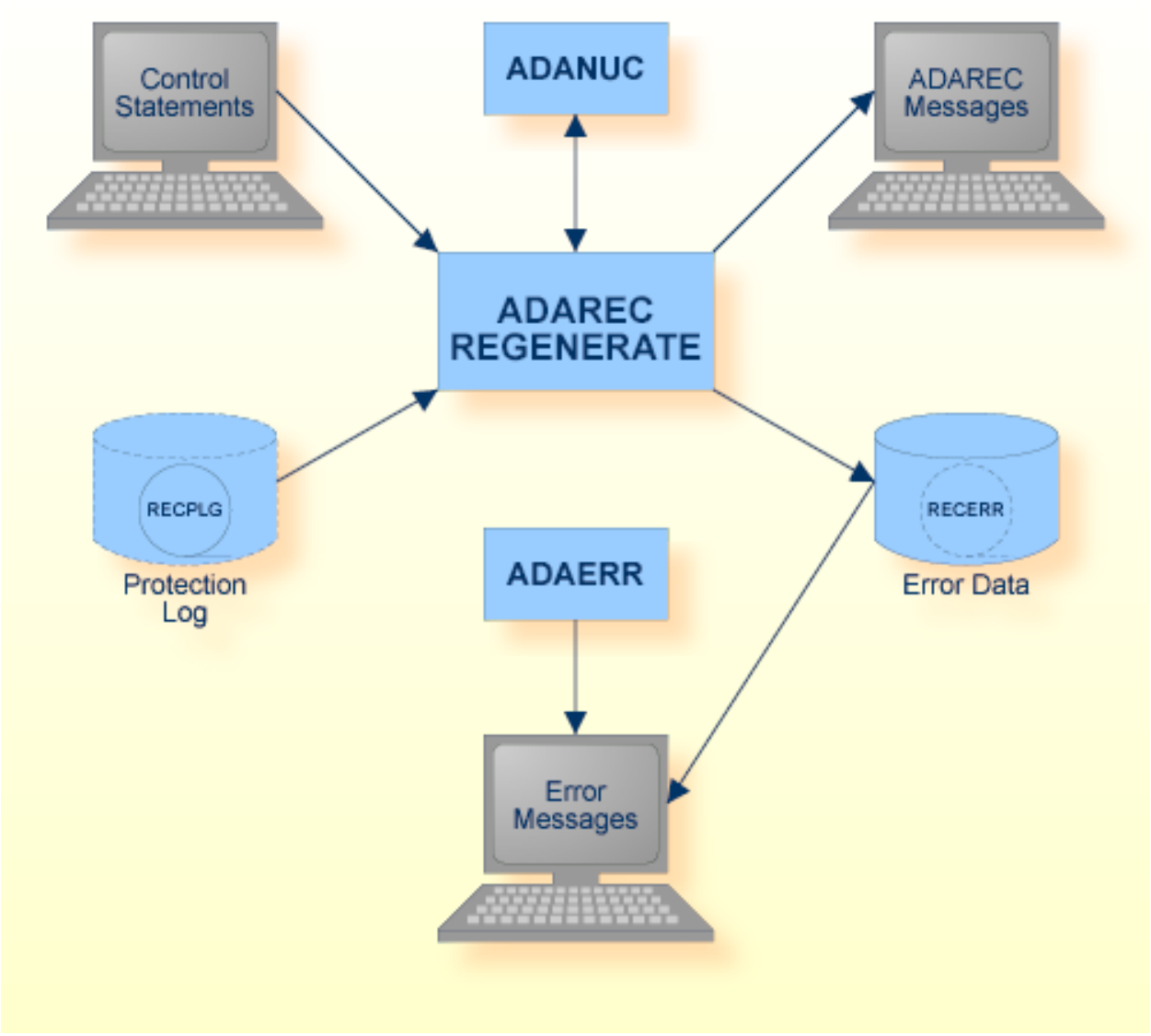
処理フロー



(UNIX platforms only)

データセット	環境 変数	記憶 媒体	追加情報
プロテクションログ	RECPLG	ディスク（* 注参照）	

注意: （*）CLOSE 機能は RAW ディスクセクションのプロテクションログファイルだけに作用します。



REGENERATE Function

データセット	環境 変数／ 論理名	記憶 媒体	追加情報
コントロールステートメント	stdin/ SYS$INPUT		ユーティリティマニュアル
ADAREC メッセージ	stdout/ SYS$OUTPUT		メッセージおよびコード
拒否データ	RECERR	ディスク、テープ（* 注参照）	ADAREC の出力
プロテクションログ	RECPLG	ディスク、テープ	

注意: （*）このシーケンシャルファイルには、名前付きパイプを使用できます（OpenVMS にはない。詳細については、『*Adabas Basics*』の「*ユーティリティの使用*」を参照）。

シーケンシャルファイル RECPLG は複数のエクステントを持つことができます。複数のエクステントを持つシーケンシャルファイルの詳細については、『*Adabas Basics*』の「*ユーティリティの使用*」を参照してください。

チェックポイント

次の表は、各機能に対するニュークリアス条件と記録チェックポイントを示しています。

機能	ニュークリアスのアクティブ化が必要	ニュークリアスの非アクティブ化が必要	ニュークリアスは不要	記録チェックポイント
LIST			X	-
REGENERATE	X			SYNX

ADAREC 入力データ

データプロテクション情報は、更新レコードのビフォーイメージとアフターイメージという形式で、各 Adabas セッション時にプロテクションログに記録されます。この情報は、更新内容の再生成に必要となります。

制御パラメータ

次のコントロールパラメータを使用できます。

```
     CLOSE = PLOG-number[(extent-number)]

M    DBID = number

     LIST = keyword

     REGENERATE = {* [,EXCLUDE_FILES =
                  (number[-number] [,number[-number] ] ... ) } |
                  (number[-number] [,number[-number] ] ... )
                  PLOG = number
D                 [,[NO]BI_CHECK]
                  [,BLOCK = ([number][,number])
                       ,CHECKPOINT = ([string][,string])]
D                 [,[NO]ERROR_LOG]
D                 [,ON_ERROR = keyword]
```

CLOSE

```
CLOSE = PLOG-number[(extent-number)]
```

CLOSE 機能は、ディスクセクション内で異常終了したプロテクションログファイルにエンドオブファイルを明示的に書き込みます。この機能を実行してからでないと、このようなプロテクションログファイルを REGENERATE 機能の入力として使用できません。

CLOSE 機能は、AUTORESTART が保留中であるか、または AUTORESTART が実行された後で実行が可能となります。

この機能は、以降の Adabas セッションで他のプロテクションログデータファイルが作成された場合でも引き続き使用できます。

PLOG-number および extent-number は、Adabas プロテクションログ番号と、クローズするプロテクションログファイルのエクステント番号を指定します。これらの番号は ADADEV の LAYOUT 機能で表示されます。

注意: この機能は、UNIX プラットフォームのみ利用できます。

例：

```
adarec: db=1
%ADAREC-I-DBON,  database 1 accessed online
adarec: close=93
adarec:

Protection log 93 - 20-JUL-2005 13:12:54 closed successfully
```

データベース 1 のプロテクションログ 93 が CLOSE 機能によってクローズされます。

DBID

```
DBID = number
```

このパラメータは、使用対象となるデータベースを選択するためのものです。

注意: ニュークリアスが動作している必要がないプログラム機能でも、コンテナファイルの環境変数／論理名の設定が必要になります。

LIST

```
[PLOG=number,] LIST = keyword
```

有効なキーワードは BRIEF、FULL、および RESTART です。BRIEF を指定すると、プロテクションログの番号と作成日が表示されます。FULL を指定すると、チェックポイントやファイルごとの変更の数など、プロテクションログのレコードに関する詳細情報が表示されます。RESTART を指定すると、処理中にチェックポイントが検出されたときに ADAREC が書き込んだ再開位置が表示されます。

注意: チェックポイントに表示されるタイムスタンプは、チェックポイントが PLOG に含まれていたときに作成されたタイムスタンプです。オフラインのチェックポイントが作成されると、まず、ASSO のチェックポイントブロックにチェックポイントが書き込まれます。次回ニュークリアスを起動したときに、実際のチェックポイント作成日とともにチェックポイントファイルに書き込まれ、現在の日付で PLOG に書き込まれます。これは、ADAREP CHECKPOINT と ADAREC LIST で表示されるオフラインチェックポイントのタイムスタンプが異なることを意味します。

また LIST=FULL 機能を指定した場合には、内部的な整合性が取れているかどうかプロテクションログ構造のチェックも実行されます。構造エラーが検出されると、そのレコードとブロック番号とともにエラータイプを示すメッセージが出力されます。

プロテクションログがディスクセクション内にある場合、LIST を指定する前に PLOG パラメータを設定する必要があります。

例

```
adarec: list=brief

Protection log 1 - 26-OCT-2006 11:39:03
```

PLOG 1 の作成日付が表示されます。

REGENERATE

この機能は、データベース全体またはデータベース内のファイルを再生成するために使用します。

データベースの再生成

```
REGENERATE = *, PLOG = number
             [,EXCLUDE_FILES = (number[-number][,number[-number]]...)]
             [,[NO]BI_CHECK]
             [,BLOCK = ([number][,number]),
                   CHECKPOINT = ([string][,string])]
             [,[NO]ERROR_LOG]
             [,ON_ERROR = keyword]
```

REGENERATE機能のこのオプションは、データベースの再生成を行うためのものです。再生成から特定のファイルを除外したい場合にはファイル除外リストを使用できます。ET ロジックがサポートされています。

REGENERATE 処理中に ADAREC は、ユーティリティ専用モードにデータベースを設定します。SYNP チェックポイントが検出されると処理は終了します。この場合、ADAREC は、代わりの再開位置を計算するためプロテクションログを調査します。その後、この再開位置は、処理を継続する前に実行する必要があるユーティリティ機能のリストとともに表示されます。REGENERATEに対する次回のコールは、自動的にこのポイントに設定されます。計算された再開位置の使用は、"BLOCK=" または "CHECKPOINT=" を指定することで（これらのキーワードに空の値を指定）無効にできます。この手順は、PLOGの最後に到達するまで繰り返されます。ADAREC が終了しても、データベースはユーティリティ専用モードになったままです。REGENERATEがさらに続けてコールされることがあるからです。データベースの再生成が終了したら、ADAOPR コマンド OPTIONS=NOUTILITIES_ONLY を実行すれば、データベースを通常どおりに処理することができます。

[NO]BI_CHECK

このオプションを BI_CHECK に設定すると、ADAREC はデータベース内のデータに対してプロテクションログのビフォーイメージの整合性をチェックします（ISNが使用中か、レコードが存在するか、ビフォーイメージの不一致が存在するかなど）。不一致が見つかった場合、ADAREC は関連情報を含むメッセージを発行し、更新は実行されません。

このオプションを NOBI_CHECK に設定した場合も整合性チェックは実行され、ERROR_LOG が暗黙的に有効になりますが、BI 不整合が見つかると更新が実行され、ERROR_LOG に不一致がレポートされます（後述）。エラーが見つかった場合、各ファイルの最初のエラーのみが表示され、以降のエラーはすべて ERROR_LOG に記録されます。この場合、インデックスが不整合になる可能性がある点に注意してください。

ただし、PLOGがニュークリアスのNOBIオプションで書き込まれている場合、ビフォーイメージは含まれず、BI_CHECK オプションを設定することはできません。

デフォルトは BI_CHECK です。

BLOCK = ([number][,number])

このパラメータは、該当するチェックポイント名を持つプロテクションログファイル内のブロックの番号を指定します。ブロック番号は、ADAREC LIST=FULL から取得できます。

CHECKPOINT = ([string][,string])

このパラメータは、開始および終了チェックポイント名を指定します。チェックポイント名は、ADAREP データベースステータスレポートまたは ADAREC LIST=FULL から取得できます。

プロテクションログファイルの先頭から処理を開始する場合、最初のパラメータを省略する必要があります。

[NO]ERROR_LOG

このオプションを ERROR_LOG に設定すると、NOBI_CHECK オプションを使用したときに検出される BI 不整合の自動ロギングが有効になります。生成されたエラーファイルの内容は、ADAERR ユーティリティを使用して判定できます。レコードの中に印刷できない文字が含まれているため、オペレーティングシステム標準の印刷ユーティリティでエラーファイルを印刷しないでください。詳細については、「*ADAERR*」を参照してください。

デフォルトは NOERROR_LOG です。

EXCLUDE_FILES = (number[-number][,number[-number]]...)

このパラメータは、データベース全体の再生成時に除外されるファイルを指定するものです。除外された更新はレポートに書き込まれます。

ON_ERROR = keyword

有効なキーワードは ABORT または EXCLUDE です。使用するキーワードによって、ADAREC が処理中に致命的ではないエラー（レスポンスコード 17、ファイルのロード失敗など）を検出したときに実行するアクションが特定されます。データストレージエラーが発生すると（ニュークリアスレスポンスコード 17、49、75、77、113 など）、ABORT を指定している場合は再生成処理が異常終了し、EXCLUDEを指定している場合は問題のファイルを再生成から除外します。

ただし、ファイルのインデックスの更新中にエラーが発生した場合（ニュークリアスレスポンスコード 75、76、77、98、165、166、167、176）、このファイルに対するデータストレージの再生成だけは続行されます。再生成処理が完了すると、このファイルのインデックスは無効としてマークされます。その後に ALL_FIELDS オプションを指定して ADAINV REINVERT 機能を実行する必要があります（詳細については、このマニュアルの ADAINV ユーティリティを参照）。インデックスエラーが発生し、再生成に複数のプロテクションログが含まれる場合は、インデックスを再インバートする前に、すべてのプロテクションログを処理する必要があります。プロテクションログにインデックスエラーが発生するたびにインデックスを再インバートすると、相当な時間とコンピュータリソースを浪費することになります。

デフォルトは ON_ERROR = EXCLUDE です。

PLOG = number

このパラメータは、REGENERATE 機能の入力として使用する Adabas プロテクションログのログ番号を指定します。この番号は、ADAREC で LIST = BRIEF 機能を使用することで見つけることができます。

ファイルの再生成

```
REGENERATE = (number[-number][,number[-number]]...), PLOG = number
             [,[NO]BI_CHECK]
             [,BLOCK = ([number][,number]),
                   CHECKPOINT = ([string][,string])]
             [,[NO]ERROR_LOG]
             [,ON_ERROR = keyword]
```

REGENERATE 機能のこのオプションは、指定したファイルまたは指定した範囲のファイルに対して、プロテクションログ内のすべての更新に再適用を行うためのものです。LOB ファイルを指定した場合には無視されますが、基本ファイルをすべて指定していれば、その基本ファイルに割り当てられた LOB ファイルはダンプされます。

再生成の処理中には、ADAREC はファイルをロックして排他的に使用します。プロテクションログの処理中に、SYNP チェックポイントが検出されると、再生成処理は終了します。この場合、ADAREC は、代わりの再開位置を計算するためプロテクションログを調査します。その後、この再開位置は、処理を継続する前に実行する必要があるユーティリティ機能のリストとともに表示されます。REGENERATE に対する次回のコールは、自動的にこのポイントに設定されます。計算された再開位置の使用は、"BLOCK=" または "CHECKPOINT=" を指定することで（これらのキーワードに空の値を指定）無効にできます。この手順は、プロテクションログの最後に到達するまで繰り返されます。

ファイルはロックされたままの状態になります。REGENERATE がさらに続けてコールされる可能性があるからです。ファイルの再生成が終わったら、ADAOPR コマンド UNLOCK でファイルをロック解除する必要があります。

ADAREC の実行中は、次の機能は利用できません。

- ADAOPR ET_SYNC FEOF = PLOG
- ADABCK DUMP
- 関連する ADAREC ユーザーが存在する間のサブユーザーに対する ADAOPR STOP

[NO]BI_CHECK

このオプションを BI_CHECK に設定すると、ADAREC はデータベース内のデータに対してプロテクションログのビフォーイメージの整合性をチェックします（ISNが使用中か、レコードが存在するか、ビフォーイメージの不一致が存在するかなど）。不一致が見つかった場合、ADAREC は関連情報を含むメッセージを発行し、更新は実行されません。

このオプションを NOBI_CHECK に設定した場合も整合性チェックは実行され、ERROR_LOG が暗黙的に有効になりますが、BI 不整合が見つかると更新が実行され、ERROR_LOG に不一致がレポートされます（後述）。エラーが見つかった場合、各ファイルの最初のエラーのみが表示され、以降のエラーはすべて ERROR_LOG に記録されます。この場合、インデックスが不整合になる可能性がある点に注意してください。

NOBI_CHECK は、データの整合性が失われる可能性と引き換えにパフォーマンスを改善します。したがって、ミッションクリティカルなデータベースに NOBI_CHECK を使用することはお勧めしません。

デフォルトは BI_CHECK です。

BLOCK = ([number] [,number])

このパラメータは、対応するチェックポイント名を含むプロテクションログファイルのブロックを指定します。ブロック番号は、ADAREC LIST=FULL から取得できます。

CHECKPOINT = ([string] [,string])

このパラメータは、開始および終了チェックポイント名を指定します。チェックポイント名は、ADAREP データベースステータスレポートから取得できます。

プロテクションログファイルの先頭から処理を開始する場合、最初のパラメータを省略する必要があります。最初のチェックポイント名を指定した場合は、そのチェックポイントが最初のプロテクションログファイルに含まれている必要があります。

最後のプロテクションログファイルの末尾で処理を終了する場合、2番目のチェックポイント名を省略する必要があります。

[NO]ERROR_LOG

このオプションを ERROR_LOG に設定すると、NOBI_CHECK オプションを使用したときに検出される BI 不整合の自動ロギングが有効になります。生成されたエラーファイルの内容は、ADAERR ユーティリティを使用して判定できます。レコードの中に印刷できない文字が含まれているため、OvenVMS標準の印刷ユーティリティでエラーファイルを印刷しないでください。

詳細については、このマニュアルの ADAERR ユーティリティに関する説明を参照してください。

デフォルトは NOERROR_LOG です。

ON_ERROR = keyword

有効なキーワードは ABORT または EXCLUDE です。使用するキーワードによって、ADAREC が処理中に致命的ではないエラー（レスポンスコード 17、ファイルのロード失敗など）を検出したときに実行するアクションが特定されます。データストレージエラーが発生すると（ニュークリアスレスポンスコード 17、49、75、77、113 など）、ABORT を指定している場合は再生成処理が異常終了し、EXCLUDE を指定している場合は問題のファイルを再生成から除外します。

ただし、ファイルのインデックスの更新中にエラーが発生した場合（ニュークリアスレスポンスコード 75、76、77、98、165、166、167、176）、このファイルに対するデータストレージの再生成だけは続行されます。再生成処理が完了すると、このファイルのインデックスは無効としてマークされます。その後に ALL_FIELDS オプションを指定して ADAINV REINVERT 機能を実行する必要があります（詳細については、このマニュアルの ADAINV ユーティリティを参照）。インデックスエラーが発生し、再生成に複数のプロテクションログが含まれる場合は、インデックスを再インバートする前に、すべてのプロテクションログを処理する必要があります。プロテクションログにインデックスエラーが発生するたびにインデックスを再インバートすると、相当な時間とコンピュータリソースを浪費することになります。

デフォルトは ON_ERROR = EXCLUDE です。

PLOG = number

このパラメータは、REGENERATE 機能の入力として使用する Adabas プロテクションログのログ番号を指定します。この番号は、ADAREC で LIST = BRIEF 機能を使用することで見つけることができます。

例

例 **1**

この例では、プロテクションログ 2 を使用してデータベース 2 を再生成します。ファイル 12 を再生成から除外します。

```
adarec: regenerate=*,plog=2
adarec: exclude_files=12
adarec:

Protection log 2 - 26-OCT-2006 11:48:59


Block     3 - checkpoint SYNC - 11:49:00 - USERID ADANUC <version>
%ADAREC-I-CHKIGN, Checkpoint ignored

The following utility functions were executed in the original session:

Block     4 - checkpoint SYNP - 11:50:02 - USERID ADADBM REFRESH=13

Block     5 - checkpoint SYNX - 11:50:03 - USERID ADADBM RESET=UCB,IDENT=7

Block     6 - checkpoint SYNP - 11:50:03 - USERID ADADBM RECOVER

Re-execute all SYNP utility functions starting from block 4.

REGENERATE summary


Calculated RESTART point - BLOCK=6,CHECKPOINT=SYNP
```

プロテクションログの処理は、ブロック4内のSYNPチェックポイントで終了します。しかし、プロテクションログを検索して更新が確認されなくても、処理は算出されたブロック6内の再開位置から継続することができます。処理継続前に実行しなければならないユーティリティ機能のリストが ADAREC より表示されます。REGENERATE=* を次回にコールすると、計算された再開位置から自動的に継続します。

```
adarec: regenerate=*,plog=2
%adarec-I-restartp, calculated restart point - block=6,checkpoint=synp
adarec: exclude_files=12
adarec:

Protection log 2 - 26-OCT-2006 11:48:59.86


Block     6 - checkpoint SYNP - 11:50:03.86 - USERID ADADBM RECOVER
%ADAREC-I-CHKSTP, starting checkpoint

     1 modifications in file  11
     1 modifications EXCLUDED from file  12

     4 ET commands issued

Block     7 - checkpoint SYNC - 11:52:38.98 - USERID ADANUC SHUTDOWN
%ADAREC-I-CHKIGN, Checkpoint ignored
```

```
REGENERATE summary


Protection log 2 processed
```

プロテクションログの処理は計算された再開位置から継続します。再生成は正常終了します。

例 **2**

この例では、プロテクションログ2を使用してデータベース2を再生成します。処理は、プロテクションログのブロック 6 内のチェックポイント SYNP から開始されます。データストレージエラーが発生すると、該当ファイルは再生成から除外されます。インデックスエラーが発生すると、該当ファイルのインデックスは再生成から除外され、無効なものとして扱われます。

```
adarec: regenerate=*,plog=2,block=6,checkpoint=synp
adarec: on_error=exclude
adarec:

Protection log 2 - 26-OCT-2006 11:48:59.86

%ADAREC-W-UTIENA, OPTIONS=UTILITIES enabled in nucleus by ADAREC
%ADAREC-W-RECUPD, Updates performed between Nucleus and REGENERATE'S startup
%ADAREC-W-RECCMD, 1 N1  command(s)

Block     6 - checkpoint SYNP - 11:50:03.86 - USERID ADADBM RECOVER
%ADAREC-I-CHKSTP, starting checkpoint

     1 modifications in file  11

     3 ET commands issued

%ADAREC-E-ISNINUSE, ISN 774 in use in file  12
%ADAREC-I-PLOGRB, from record 14 in block 7 in PLOG 2
%ADAREC-I-UPDEXC, ALL following updates in file  12 will be EXCLUDED

     1 modifications EXCLUDED from file  12

     1 ET command issued

Block     7 - checkpoint SYNC - 11:52:38.98 - USERID ADANUC SHUTDOWN
%ADAREC-I-CHKIGN, Checkpoint ignored


REGENERATE summary


Protection log 2 processed
```

ファイル 12 において ISN の矛盾が発生し、このファイルに対する後続の更新はすべて除外されました。このエラーの原因を調査しなければなりません。ただし、そのニュークリアスは OPTIONS=UTILITIES_ONLY の指定なしで起動され、再生成が開始される前に N1 コマンドが 1 回発行されています。

プロテクションログは最後まで処理され、終了システムメッセージは致命的なエラーを示すためにのみ使用されています。

例 **3**

この例は、データストレージまたはインデックスエラーが発生した際に処理が異常終了するという点以外はこの前の例と同じものです。

```
adarec: regenerate=*,plog=2,block=6,checkpoint=synp
adarec: on_error=abort
adarec:

Protection log 2 - 26-OCT-2006 11:48:59.86

%ADAREC-W-UTIENA, OPTIONS=UTILITIES enabled in nucleus by ADAREC
%ADAREC-W-RECUPD, Updates performed between Nucleus and REGENERATE'S startup
%ADAREC-W-RECCMD, 1 N1  command(s)

Block     6 - checkpoint SYNP - 11:50:03.86 - USERID ADADBM RECOVER
%ADAREC-I-CHKSTP, starting checkpoint

     1 modifications in file  11

     3 ET commands issued

%ADAREC-E-ISNINUSE, ISN 774 in use in file  12
%ADAREC-I-PLOGRB, from record 14 in block 7 in PLOG 2
```

ISN の競合がファイル 12 で発生しました。以降の処理は異常終了しました。

例 **4**

この例では、プロテクションログ3を使用してデータベース2を再生成します。プロテクションログ内のビフォーイメージがデータベース内のデータに対してチェックされ、不一致内容が端末画面上に表示されます。

```
adarec: regenerate=*,plog=3
adarec:

Protection log 3 - 26-OCT-2006 12:10:25.12

%ADAREC-W-UTIENA, OPTIONS=UTILITIES enabled in nucleus by ADAREC

Block     1 - checkpoint SYNC - 12:10:25.12 - USERID ADANUC 3.2/0 PL 0
%ADAREC-I-CHKIGN, Checkpoint ignored

     1 ET command issued

%ADAREC-E-RECMIS, Before image mismatch for ISN 3 in file  11
%ADAREC-I-PLOGRB, from record 7 in block 2 in PLOG 3
%ADAREC-I-UPDEXC, ALL following updates in file  11 will be EXCLUDED

     1 modifications EXCLUDED from file  11
     1 modifications in file  12

     3 ET commands issued

Block     2 - checkpoint SYNC - 12:11:44.30 - USERID ADANUC SHUTDOWN
%ADAREC-I-CHKIGN, Checkpoint ignored


REGENERATE summary


Protection log 3 processed
```

処理時に1つのビフォーイメージの不一致が発生しました。結果として、1つの更新内容がファイル 11 から除外されます。

ファイルの再生成後、同じ例を今度はビフォーイメージの整合性チェックなしで、またBIエラーロギングを可能にして実行できます。

プロテクションログは最後まで処理され、終了システムメッセージは致命的なエラーを示すためにのみ使用されています。

```
adarec: regenerate=*,plog=3,nobi_check
adarec:

Protection log 3 - 26-OCT-2006 12:10:25.12

%ADAREC-W-UTIENA, OPTIONS=UTILITIES enabled in nucleus by ADAREC

Block     1 - checkpoint SYNC - 12:10:25.12 - USERID ADANUC 3.2/0 PL 0
%ADAREC-I-CHKIGN, Checkpoint ignored
```

```
%ADAREC-W-RECMIS, Before image mismatch for ISN 3 in file  11
%ADAREC-I-PLOGRB, from record 7 in block 2 in PLOG 3

     1 modifications in file  11
     1 modifications in file  12

     4 ET commands issued

     1 BI_CHECK error in file  11

Block     2 - checkpoint SYNC - 12:11:44.30 - USERID ADANUC SHUTDOWN
%ADAREC-I-CHKIGN, Checkpoint ignored


REGENERATE summary

     1 BI_CHECK error in file  11


Protection log 3 processed
```

処理の際に、1 つの BI_CHECK エラーが発生しました。

プロテクションログは最後まで処理され、終了システムメッセージは致命的なエラーを示すためにのみ使用されています。

エラーの発生源は、ADAERR ユーティリティで表示できるように、エラーファイルに書き出されます。最初に検出されたエラーが記録され、エラーファイル内にも書き出されます。このエラーの後の後続エラーは、ERROR_LOG に書き出されます。

次のエラーファイルが作成されています。

```
%ADAERR-E-RECMIS, Before image mismatch for ISN 3 in file  11
%ADAERR-I-PLOGRB, from record 7 in block 2 in PLOG 3
```

例 **5**

この例では、プロテクションログ 3 を使用してデータベース 2 を再生成します。

```
adarec: regenerate=*,plog=3
adarec:

Protection log 3 - 26-OCT-2006 12:10:25.12

%ADAREC-W-UTIENA, OPTIONS=UTILITIES enabled in nucleus by ADAREC

Block     1 - checkpoint SYNC - 12:10:25.12 - USERID ADANUC 3.2/0 PL 0
%ADAREC-I-CHKIGN, Checkpoint ignored
```

```
%ADAREC-E-ERRIUP, Error response 165 during index update
%ADAREC-E-Adabas_165, * Invalid descriptor name in DVT
%ADAREC-I-DESNAM, Descriptor name XA
%ADAREC-I-ISNFILE, from ISN 3 in file  11
%ADAREC-I-PLOGRB, from record 7 in block 2 in PLOG 3
%ADAREC-I-REINVERT, REINVERT all descriptors to re-establish INDEX
%ADAREC-I-REGDAT, Regenerating ONLY data-storage for file  11

     1 modifications in file  11
     1 modifications in file  12

     4 ET commands issued

Block     2 - checkpoint SYNC - 12:11:44.30 - USERID ADANUC SHUTDOWN
%ADAREC-I-CHKIGN, Checkpoint ignored


 REGENERATE summary


Protection log 3 processed
```

処理の際に無効なディスクリプタ名が検出されました。その結果、ファイル11のデータストレージのみが再生成されました。インデックスを再構築するには、全ディスクリプタを再インバートする必要があります。

プロテクションログは最後まで処理され、終了システムメッセージは致命的なエラーを示すためにのみ使用されています。

インデックスエラーが発生し、再生成に複数のプロテクションログが含まれる場合は、インデックスを再インバートする前に、すべてのプロテクションログを処理する必要があります。プロテクションログにインデックスエラーが発生するたびにインデックスを再インバートすると、相当な時間とコンピュータリソースを浪費することになります。

例 **6**

この例では、プロテクションログ3の再生成処理がインデックスエラーになった後で、プロテクションログ 4 を使用して、データベース 2 を再生成します。

```
adarec: regenerate=*,plog=4
adarec:

Protection log 4 - 26-OCT-2006 12:12:00.15


Block     1 - checkpoint SYNC - 12:12:00.15 - USERID ADANUC <version>
%ADAREC-I-CHKIGN, Checkpoint ignored

%ADAREC-E-FCBNAC, file  11's index not accessible
```

```
%ADAREC-I-REGDAT, Regenerating ONLY data-storage for file  11

     1 modifications in file  11
     1 modifications in file  12

     4 ET commands issued

Block     2 - checkpoint SYNC - 12:12:19.35 - USERID ADANUC SHUTDOWN
%ADAREC-I-CHKIGN, Checkpoint ignored


REGENERATE summary


Protection log 4 processed
```

プロテクションログ 3（上記例 5 を参照）の処理時にインデックスエラーが発生したことは、ファイル 11 のインデックスがもうアクセス不可能であるということを意味します。ファイル 11 はデータストレージだけが再生成されますが、ファイル 12 はデータストレージもインデックスも再生成されます。

プロテクションログは最後まで処理され、終了システムメッセージは致命的なエラーを示すためにのみ使用されています。

ADAREC の再スタートに関する考慮

ADAREC の実行が中断された結果、UCB エントリがなくなった場合には、ADAREC を最初からやり直す必要があります。修正がすでに行われているので、データベースまたはファイルを対象に RESTORE を実行してから、ADAREC を再スタートする必要があります。ただし、UCB エントリが 1 つも存在しない場合、データベースは修正されておらず、ADAREC を再スタートすることができます。

ADAREC（RESTORE/RECOVER）が異常終了した場合のデータベースの状態は、整合性が維持されることはわかっていますが、実際にどうなるかはっきりとはわかりません。ADAREC は、どれがリカバリ済みのトランザクションなのかを判断できません。そのため、RESTORE オペレーションを繰り返して、すべてを確実にリカバリするために最初から ADAREC をやり直す必要があります。

最初の更新が実行されると、started チェックポイントが ADAREC によってチェックポイントファイルに記録されます。

```
  SYNX   22-MAR-2007 16:49:46       192   ADAREC REG STARTED
```

28 ADAREP（データベースレポート）

この章では ADAREP ユーティリティについて説明します。

機能概要

ADAREP ユーティリティは、データベースステータスレポートを生成します。これには、データベースの現在の物理レイアウト、論理内容に関する情報が含まれます。特に断りのない限り、ニュークリアスがアクティブかどうかに関係なく、この機能を実行することができます。

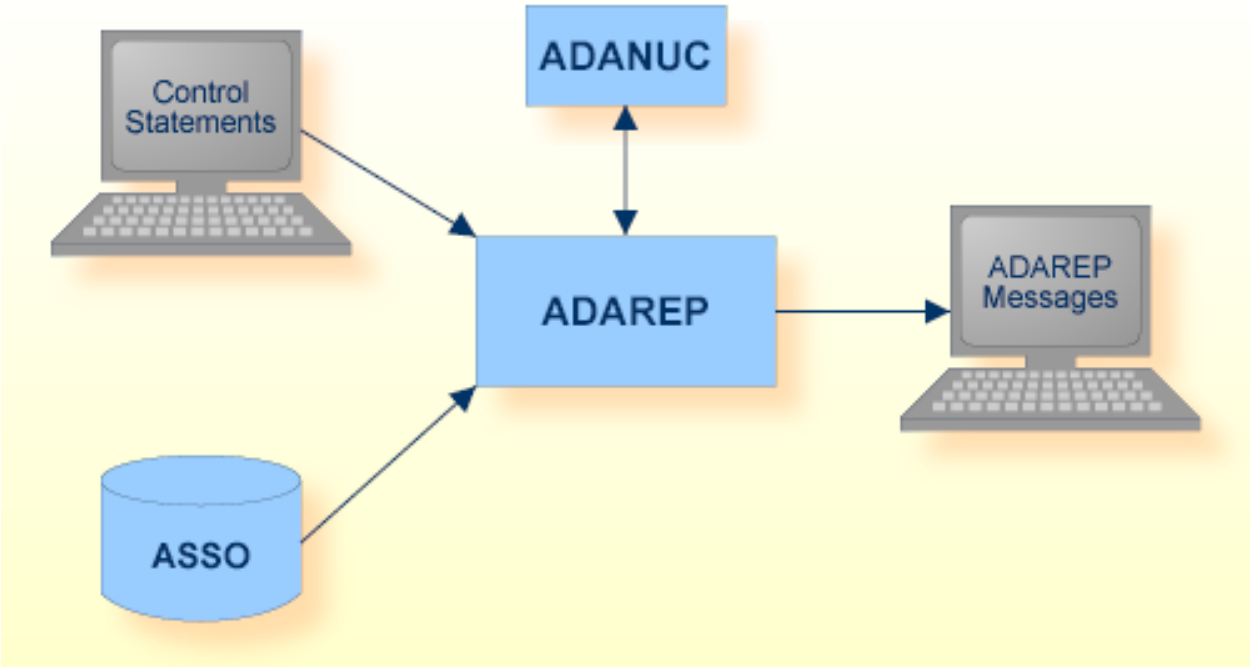
このレポートに出力される情報は次のものです。

- アソシエータとデータストレージの現在の割り当てスペース量と位置。
- アソシエータとデータストレージの使用可能な未使用スペース量と位置。
- データベースファイルのサマリ。
- チェックポイント情報。
- セキュリティ情報。
- データベース内の各ファイルの情報（スペース割り当て、使用可能スペース、ロードされたレコード数、MAXISN 値、フィールド定義など）。

CHECKPOINTS 制御パラメータ（後述）のみ、ニュークリアスがアクティブであることが条件になります。

このユーティリティは多機能ユーティリティです。

処理フロー



データセット	環境変数／論理名	記憶媒体	追加情報
アソシエータ	ASSOx	ディスク	
コントロールステートメント	stdin/ SYS$INPUT		ユーティリティマニュアル
ADAREP レポート	stdout/ SYS$OUTPUT		メッセージおよびコード、ユーティリティマニュアル

チェックポイント

このユーティリティはチェックポイントを書き込みません。

制御パラメータ

次のコントロールパラメータを使用できます。

```
 CHECKPOINTS = { * |  ( [absolute-date] [,[absolute-date] ] ) }

 CONSTRAINTS

 CONTENTS

 COUNT

M DBID = number

D [NO]FDT

 FILES = { * | (number [-number][,number[-number]]...) }

D [NO]FULL

 FREE_SPACE

 LAYOUT

 SUMMARY
```

CHECKPOINTS

```
CHECKPOINTS = { * | ( [absolute-date] [,[absolute-date]] ) }
```

この機能は、チェックポイントリストからの選択情報を出力します。ニュークリアスがアクティブなときに使用可能です。

5 種類のシステムチェックポイント（SYNP、SYNC、SYNX、OPEN、CLSE）がチェックポイントファイルとプロテクションログに書き込まれますが、C1 コマンドが書き込むユーザーチェックポイントもここに書き込まれます。

SYNC は、ニュークリアスの初期化時、終了時、キャンセル時、ADAOPR 機能の FEOF=PLOG の最中、ADAOPR EXT_BACKUP=CONTINUE 機能の実行時、または ADABCK NEW_PLOG 機能の最中に作成されたチェックポイントを示します。

SYNP は、特権的な制御が必要な Adabas ユーティリティによってチェックポイントが作成されたことを示します。すなわち、モジュールがニュークリアスを使用しないで更新ができる場合です。SYNP は、例えば ADAMUP UPDATE 処理の終了時に書き込まれます。

SYNX は、1 つまたは複数ファイルの排他制御が必要なユーティリティによって作成されたことを示します。例えば、SYNX は ADAULD によって書き込まれます。

OPEN チェックポイントは、EXU/EXF ユーザーの OP コマンドによって書き込まれます。

CLSE チェックポイントは、EXU/EXF ユーザーの CL コマンドによって書き込まれます。

注意: プロテクションログを使用して ADAREC REGENERATE 機能を実行する場合、DBA の操作が必要になるため、SYNP チェックポイントごとに、このユーティリティは停止します。

アスタリスク（*）を入力すると、すべてのチェックポイントが表示されます

日付文字列は、次の絶対日付フォーマットと時刻フォーマットに対応していなければなりません。

```
dd-mmm-yyyy[:hh[:mm[:ss]]]
```

日付と時刻の仕様の上位桁のゼロは、省略される可能性があります。指定されていない数字は、0 に設定されます（例：28-jul-2006 は 28-jul-2006:00:00:00 と同じです）。

次の表は、パラメータ CHECKPOINTS に使用できる値と、この値に対応して表示されるチェックポイントを示しています。

パラメータ CHECKPOINTS に指定された値	この指定で表示されるチェックポイント
* または（,）	すべてのチェックポイント
absolute-date	指定された日時と書き込み時間が正確に一致するチェックポイントのみ
(absolute-date,)	指定された日時以降に書き込まれたチェックポイント
(,absolute-date)	指定された日付と時刻までに書き込まれたチェックポイント
(absolute-date,absolute-date)	指定された最初の日付と時刻の値から、指定された 2 番目の日付と時刻の値までに書き込まれたチェックポイント

例

```
adarep: checkpoints=*


Name        Date/Time        Session   User Id / Function
----        ---------        -------   ------------------
SYNP   28-JUL-2006 12:50:34        8   ADADBM DELCP
SYNX   28-JUL-2006 12:50:36        8   ADABCK DUMP=* STARTED
SYNX   28-JUL-2006 12:50:37        8   ADABCK DUMP=*
OPEN   28-JUL-2006 17:23:53        8   otto
OPEN   28-JUL-2006 17:24:15        8   otto
CLSE   28-JUL-2006 17:24:24        8   otto
```

全チェックポイントが表示されます。

［User ID / Function］列には、次のデータが含まれています。

- EXU/EXF ユーザー用の OP/CL コマンドまたは C1 コマンドを使用して作成されたユーザーチェックポイントの場合：関連する OP コマンドのアディション 1 フィールドで指定されたユーザー。
- ユーティリティチェックポイントの場合：実行されたユーティリティ機能

上記の例（checkpoints=*）の出力では、以下に示すように、選択条件を使用してチェックポイントを絞り込むことができます。

次を指定するとします。

```
checkpoints=28-jul-2006:12:50:36
```

出力結果は次のようになります。

```
Name        Date/Time        Session   User Id / Function
----        ---------        -------   ------------------
SYNX   28-JUL-2006 12:50:36        8   ADABCK DUMP=* STARTED
```

次を指定するとします。

```
checkpoints=(28-jul-2006:12:50:36,)
```

出力結果は次のようになります。

```
Name        Date/Time        Session   User Id / Function
----        ---------        -------   ------------------
SYNX   28-JUL-2006 12:50:36        8   ADABCK DUMP=* STARTED
SYNX   28-JUL-2006 12:50:37        8   ADABCK DUMP=*
OPEN   28-JUL-2006 17:23:53        8   otto
OPEN   28-JUL-2006 17:24:15        8   otto
CLSE   28-JUL-2006 17:24:24        8   otto
```

次を指定するとします。

```
checkpoints=(,28-jul-2006:12:50:36)
```

出力結果は次のようになります。

```
Name        Date/Time        Session   User Id / Function
----        ---------        -------   ------------------
SYNP   28-JUL-2006 12:50:34        8   ADADBM DELCP
SYNX   28-JUL-2006 12:50:36        8   ADABCK DUMP=* STARTED
```

次を指定するとします。

```
checkpoints=(28-jul-2006:17, 28-jul-2006:17:24)
```

出力結果は次のようになります。

```
Name        Date/Time        Session   User Id / Function
----        ---------        -------   ------------------
OPEN   28-JUL-2006 17:23:53        8   otto
```

CONSTRAINTS

```
CONSTRAINTS
```

この機能は、DBIDパラメータで指定したデータベース内の、すべての参照制約に関する情報を表示します。

例

```
adarep: constraints

     Primary file          Foreign file    Name   Action
     ------------------------------------------------
           9 (AA)    <---       12 (AC)     HO     DX UX
```

この参照制約 HO は、プライマリファイル 9 のプライマリキーフィールド AA を、外部ファイル 12 の外部キーフィールド AC にリンクします。関連付けられたアクションは、削除時にアクションなし（DX）および更新時にアクションなし（UX）です。

CONTENTS

```
CONTENTS
```

この機能は、DBIDパラメータに指定するデータベースのファイルに関する情報を表示します。

例

```
adarep: contents



Content of Database 163          30-MAY-2017 11:36:07


                                                 Index   Extents   Pad % flags
 File Filename          loaded on        Top ISN level  N  U  A  D  A  D ALRCPM
-------------------------------------------------------------------------------
    1 EMPLOYEES        30-MAY-2017         1,107     3  1  1  1  1  5  5   R  M
    2 VEHICLES         30-MAY-2017           773     3  1  1  1  1  5  5   R  M
    3 MISCELLANEOUS    30-MAY-2017         1,779     3  1  1  1  1  5  5   R  M
   60 EMPL-REF         30-MAY-2017         1,107     3  1  1  1  1  5  5   R  M
  251 SECURITY-FILE    30-MAY-2017             0     3  1  1  1  1  5  5   R  M
  254 USER-DATA-FILE   30-MAY-2017             0     3  1  1  1  1  5  5   R  M
  255 CHECKPOINT-FILE  30-MAY-2017             0     3  1  1  1  1  5  5   S  M


                                Allocated blocks
 File Filename                  NI            UI            AC            DS
-------------------------------------------------------------------------------
    1 EMPLOYEES                 90            15            10            75
    2 VEHICLES                  40            20             2            40
    3 MISCELLANEOUS             50            10            10            50
   60 EMPL-REF                  90            15            10            75
  251 SECURITY-FILE              2             2             1             5
  254 USER-DATA-FILE            24             2             6            57
  255 CHECKPOINT-FILE            1             1             6            32
-------------------------------------------------------------------------------
      Total                    297            65            45           334


                                Unused blocks
 File Filename                  NI            UI            DS
-------------------------------------------------------------------------------
    1 EMPLOYEES                 4             2            18
    2 VEHICLES                 16            12            26
    3 MISCELLANEOUS            17             5            23
   60 EMPL-REF                  4             2            18
  251 SECURITY-FILE             2             1             4
  254 USER-DATA-FILE           24             1            56
  255 CHECKPOINT-FILE           1             0            31
-------------------------------------------------------------------------------
      Total                    68            23           176
```

［Extents］列には、ノーマルインデックス（N）、メイン／アッパーインデックス（U）、アドレスコンバータ（A）、およびデータストレージ（D）に現在割り当てられている論理エクステントの数が表示されます。

［Pad］列にはアソシエータ（A）およびデータストレージ（D）用に定義されたブロックパディングファクタの割合が表示されます。詳細については、ADAFDU、ADAMUP、またはADAORDの ASSOPFAC および DATAPFAC パラメータを参照してください。

［Flags］列には次の情報が表示されます。

サブカラム	フラグ	説明
A	A	Adam ファイルを示します。
L	L	ファイルは LOB ファイルです。
	B	ファイルは対応する LOB ファイルを含む基本ファイルです。
R	R	ファイルの ISN とスペースの再利用が可能です。
	I	ファイルの ISN の再利用は可能ですが、スペースの再利用は不可能です。
	S	ファイルのスペースの再利用は可能ですが、ISN の再利用は不可能です。
C	C	ファイルの暗号化が有効です。
P	P	ファイルに対するプログラムリフレッシュが有効です。
M	M	このファイルは最後にバックアップされてから変更されています。

ISNが再利用される場合、削除されたレコードのISNは新規レコードに再度割り当てられます。スペース再利用の場合、レコードを削除してできたブロック内の解放スペースは新規レコードに再利用されます（詳細については ADADBM または ADAFDU の REUSE パラメータを参照）。

2番目と3番目の表は、各ファイルについて、ノーマルインデックス（NI）、メイン／アッパーインデックス（UI）、アドレスコンバータ（AC）およびデータストレージ（DS）に割り当てられたブロック数を示しています。残りの列はメイン／アッパーインデックス（UI）およびデータストレージ（DS）の未使用ブロック数を表示します。

COUNT

```
COUNT
```

このパラメータは、DBIDパラメータで指定したデータベース内のファイルのレコード数を表示します。

例

```
adarep: count
Record Count of Database 2           20-APR-2017 16:08:52


 File Filename                     Records loaded
-------------------------------------------------------------------------------
    1 CHECKPOINT-FILE                         0
    2 SECURITY-FILE                           0
    3 USER-DATA-FILE                          0
    9 EMPLOYEES                           1,272
   11 EMPLOYEES-NAT                       1,107
   12 VEHICLES                              773
   13 MISCELLANEOUS                       1,779
   14 LOBFILE of 9                          210
```

DBID

```
DBID = number
```

このパラメータは、使用対象となるデータベースを選択するためのものです。1つのセッションで複数の DBID がサポートされます。

DBID パラメータは最初の ADAREP パラメータとして指定しなければなりません。

例

```
adarep: dbid = 1, contents
        .
        .
adarep: dbid = 2, contents
        .
        .
adarep: dbid = 3, contents
```

[NO]FDT

```
[NO]FDT
```

このオプションを FDT に設定すると、FILES 機能が続けて表示するステータス情報に、フィールド定義テーブル（FDT）も表示されます。

デフォルトは NOFDT です。

FILES

```
FILES = { * | (number [-number][,number [-number]]...) }
```

この機能は、選択したファイルステータス情報を表示します。

例

```
adarep: fdt
adarep: file=9
Database 216, File     9  (EMPLOYEES       )          30-MAY-2017 11:50:36


Highest Index Level:          3     Padding Factors:       ASSO   5%, DATA   5%
Top ISN:                  1,272     Maximum ISN expected:                 8,191
SYFMAX :                      9
Records loaded:           1,272     Corresponding LOB file:                  14
Last FDT Modification:               4-SEP-2014 14:49:36.510905
Last ADABCK dump     :           30-MAY-2017 11:49:32

ISN reusage:     Enabled, inactive  Space reusage:    Enabled
Program refresh: Disabled           Ciphering:        Disabled
Modified since last backup


Container  Block  Extent    Extents in Blocks       Allocated         Unused
  File     Size   Type       from          to      Blocks   MB      Blocks   MB
-------------------------------------------------------------------------------

ASSO:
        2  32KB   AC        7,682       7,682           1    0           0    0
        1   8KB   NI           55          99          45    0          15    0
        1   8KB   UI          100         107           8    0           0    0
        1   8KB   UI          110         119          10    0           3    0

DATA:
        1  32KB   DS           14          45          32    1          21    0

-------------------------------------------------------------------------------

Field Definition Table:

   Level  I Name I Length I Format I   Options         I Flags   I Encoding
-------------------------------------------------------------------------------
  1       I  AO  I        I        I                   I         I
   2      I  AA  I    8   I    A   I DE,UQ,NC,NN       I RP      I
   2      I  AB  I        I        I                   I         I
    3     I  AC  I    4   I    F   I DE                I         I
    3     I  AD  I    8   I    B   I NU,HF             I         I
    3     I  AE  I    0   I    A   I NU,NV,NB,LA       I         I
  1       I  BO  I        I        I                   I         I
```

```
  2       I  BA  I   40   I    W   I NU                 I          I
  2       I  BB  I   40   I    W   I NU                 I          I
  2       I  BC  I   50   I    W   I DE,NU              I SP       I
 1        I  CA  I    1   I    A   I FI                 I          I
 1        I  DA  I    1   I    A   I FI                 I          I
 1        I  EA  I    4   I    P   I DE,NC              I          I
 1        I  FO  I        I        I PE                 I          I
  2       I  FA  I   60   I    W   I NU,MU              I          I
  2       I  FB  I   40   I    W   I DE,NU              I          I
  2       I  FC  I   10   I    A   I NU                 I          I
  2       I  FD  I    3   I    A   I NU                 I          I
  2       I  F1  I        I        I                    I          I
   3      I  FE  I    6   I    A   I NU                 I          I
   3      I  FF  I   15   I    A   I NU                 I          I
   3      I  FG  I   15   I    A   I NU                 I          I
   3      I  FH  I   15   I    A   I NU                 I          I
   3      I  FI  I   80   I    A   I DE,NU,MU           I          I
 1        I  IO  I        I        I PE                 I          I
  2       I  IA  I   40   I    W   I NU,MU              I          I
  2       I  IB  I   40   I    W   I DE,NU              I          I
  2       I  IC  I   10   I    A   I NU                 I          I
  2       I  ID  I    3   I    A   I NU                 I          I
  2       I  IE  I    5   I    A   I NU                 I          I
  2       I  I1  I        I        I                    I          I
   3      I  IF  I    6   I    A   I NU                 I          I
   3      I  IG  I   15   I    A   I NU                 I          I
   3      I  IH  I   15   I    A   I NU                 I          I
   3      I  II  I   15   I    A   I NU                 I          I
   3      I  IJ  I   80   I    A   I DE,NU,MU           I          I
 1        I  JA  I    6   I    A   I DE                 I SB,SP    I
 1        I  KA  I   66   I    W   I DE,NU              I          I
 1        I  LO  I        I        I PE                 I          I
  2       I  LA  I    3   I    A   I NU                 I SP       I
  2       I  LB  I    6   I    P   I NU                 I SP       I
  2       I  LC  I    6   I    P   I DE,NU,MU           I          I
 1        I  MA  I    4   I    G   I NU                 I          I
 1        I  NO  I        I        I                    I          I
  2       I  NA  I    2   I    U   I                    I SP       I
  2       I  NB  I    3   I    U   I NU                 I SP       I
 1        I  OO  I        I        I PE                 I          I
  2       I  OA  I    8   I    U   I NU                 I          I
          I      I        I        I DT(DATE)           I          I
  2       I  OB  I    8   I    U   I NU                 I          I
          I      I        I        I DT(DATE)           I          I
 1        I  PA  I    3   I    A   I DE,NU,MU           I          I
 1        I  QA  I    7   I    P   I                    I          I
 1        I  RA  I    0   I    A   I NU,NV,NB,LB        I          I
 1        I  SO  I        I        I PE                 I          I
  2       I  SA  I   80   I    W   I NU                 I          I
  2       I  SB  I    3   I    A   I NU                 I          I
  2       I  SC  I    0   I    A   I NU,MU,NV,NB,LB     I          I
 1        I  TC  I   20   I    U   I DT(TIMESTAMP)      I          I
```

```
         I      I        I        I SY=TIME,CR        I         I
  1      I  TU  I   20   I    U   I MU                I         I
         I      I        I        I DT(TIMESTAMP)     I         I
         I      I        I        I SY=TIME           I         I
-------------------------------------------------------------------------------
  Type   I Name I Length I Format I   Options         I Parent field(s)   Fmt
-------------------------------------------------------------------------------
 COLL    I  CN  I1,144   I        I NU,HE             I BC  de@collation=phonebook
         I      I        I        I                   I     PRIMARY
-------------------------------------------------------------------------------
 SUPER   I  H1  I    5   I    B   I NU                I NA (   1 -   2 )   U
         I      I        I        I                   I NB (   1 -   3 )   U
-------------------------------------------------------------------------------
 SUB     I  S1  I    2   I    A   I                   I JA (   1 -   2 )   A
-------------------------------------------------------------------------------
 SUPER   I  S2  I   46   I    A   I NU                I JA (   1 -   6 )   A
         I      I        I        I                   I BC (   1 -  40 )   W
-------------------------------------------------------------------------------
 SUPER   I  S3  I    9   I    A   I NU,PE             I LA (   1 -   3 )   A
         I      I        I        I                   I LB (   1 -   6 )   P
-------------------------------------------------------------------------------
                          Referential Integrity
-------------------------------------------------------------------------------
  Type   I Name I Refer. I Primary I Foreign I Rules
         I      I file   I  field  I  field  I
-------------------------------------------------------------------------------
 PRIMARY I  HO  I    12  I   AA    I   AC    I DELETE_NOACTION  UPDATE_NOACTION
-------------------------------------------------------------------------------
```

FILES パラメータは、ファイル番号とファイル名、最高インデックスレベル、ASSO と DATA のパディングファクタ、最高 ISN と最大 ISN、ロードされているレコード数、基本ファイルの番号や対応する LOB ファイルがある場合はその番号、および ISN 再利用、スペース再利用、プログラムリフレッシュ、暗号化のスイッチを表示します。最後の FDT 変更の日時も表示されます。

ファイルの ASSO および DATA エレメントのレイアウトが表示されます。表示される内容は、各種リスト要素が格納されるブロックサイズ、これらのエクステントの配置および対応する割り当て済みまたは未使用のブロック数／メガバイト数などです。

さらに、FDT 機能はファイルのフィールド定義テーブルを表示します。

フィールド定義テーブルに表示されるフラグは次のとおりです。

フラグ	説明
HY	フィールドはハイパーディスクリプタの一部です。
P	フィールドはフォネティック化されています。
SB	このフィールドの一部はサブディスクリプタです。
SP	スーパーディスクリプタの一部です。

FREE_SPACE

```
FREE_SPACE
```

この機能は、ASSOまたはDATA内の空きブロックの要約情報を出力します。これは、LAYOUT機能の表示情報のサブセットです。

例

```
adarep: free_space

Free space of Database 76       28-NOV-2006 12:51:24



Container       Extents in Blocks   Number of  Block
  File           from          to    Blocks    Size
-------------------------------------------------------------------------------

ASSO:
      1-2         611       1,546         936  2,048

DATA:
        1         245         768         524  4,096
        2         769         868         100  3,072
      3-4         869         888          20  6,144
```

[NO]FULL

```
[NO]FULL
```

FDT と一緒に FULL を指定すると、FDT に関する以下の追加情報が表示されます。

- ドロップされたフィールドは、ファイルのフィールドの表示に含まれます（フィールド名は含まれません）。
- ICU バージョンは、照合ディスクリプタの表示に含まれます。

デフォルトは NOFULL です。

LAYOUT

```
LAYOUT
```

この機能は、ASSO および DATA 内のブロックの要約情報を出力し、消失ブロックをレポートするものです。消失ブロックとは、フリースペーステーブル（FST）内にリスト化されなくなったにもかかわらず、ファイル、DSST、データベースのグローバルエリアのいずれにも割り当てることができないブロックを指します。また、この機能は、二重に割り当てられたブロックもレポートします。

例

```
adarep: layout

Layout of Database 76           28-NOV-2006 12:51:24


Container       Extents in Blocks   Number of  Block  Extent  File
  File           from          to    Blocks    Size   Type   Number
-------------------------------------------------------------------------------

ASSO:
        1           1          30          30  4,096  CB
        1          31          31           1  4,096  FCB       1
        1          32          32           1  4,096  FDT       1
        1          33          35           3  4,096  AC        1
        1          36          36           1  4,096  UI        1
        1          37          37           1  4,096  NI        1
        1          38          38           1  4,096  FCB       2
        1          39          39           1  4,096  FDT       2
        1          40          40           1  4,096  AC        2
        1          41          42           2  4,096  UI        2
        1          43          43           1  4,096  NI        2
        1          44          44           1  4,096  FCB       3
        1          45          45           1  4,096  FDT       3
        1          46          48           3  4,096  AC        3
        1          49          50           2  4,096  UI        3
        1          51          62          12  4,096  NI        3
        1          63         152          90  4,096  NI        9
        1         153         167          15  4,096  UI        9
        1         168         168           1  4,096  FCB       9
        1         169         169           1  4,096  FDT       9
        1         170         170           1  4,096  NI       14
        1         171         172           2  4,096  UI       14
        1         173         173           1  4,096  FCB      14
        1         174         174           1  4,096  FDT      14
        1         175         264          90  4,096  NI       11
        1         265         279          15  4,096  UI       11
        1         280         280           1  4,096  FCB      11
        1         281         281           1  4,096  FDT      11
```

```
        1         282         321          40  4,096  NI        12
        1         322         341          20  4,096  UI        12
        1         342         342           1  4,096  FCB       12
        1         343         343           1  4,096  FDT       12
        1         344         393          50  4,096  NI        13
        1         394         403          10  4,096  UI        13
        1         404         404           1  4,096  FCB       13
        1         405         406           2  4,096  FDT       13
        1         407       2,560       2,154  4,096  FREE
        2       2,561       2,561           1 32,768  DSST
        2       2,562       2,562           1 32,768  AC         9
        2       2,563       2,563           1 32,768  AC        14
        2       2,564       2,572           9 32,768  AC        11
        2       2,573       2,581           9 32,768  AC        12
        2       2,582       2,582           1 32,768  AC        13
        2       2,583       2,880         298 32,768  FREE

DATA:
        1           1           4           4 32,768  DS         1
        1           5           5           1 32,768  DS         2
        1           6          13           8 32,768  DS         3
        1          14          45          32 32,768  DS         9
        1          46         170         125 32,768  DS        14
        1         171         202          32 32,768  DS        11
        1         203         212          10 32,768  DS        12
        1         213         222          10 32,768  DS        13
        1         223         640         418 32,768  FREE
```

LAYOUT を使用すると、ASSO および DATA の全ブロックの要約情報が出力されます。連続ブロックセクションの配置と長さ、ブロックサイズ、対応するファイルの使用タイプと番号が表示されます。このブロックはフリー（FREE）の場合も、グローバルブロック（CB）、ファイルコントロールブロック（FCB）、FCB 拡張（FCBE）、FCB ルートブロック（FCBR）、フィールド定義テーブル（FDT）、フリースペーステーブル（FST）、データスペースストレージテーブル（DSST）、ノーマルインデックス（NI）、アッパー／メインインデックス（UI）、アドレスコンバータ（AC）、データストレージ（DS）として使用されている場合もあります。

注意: 最初の FCBR ブロックと最初の FST ブロックはグローバルブロックの一部です。このため、データベースにこのようなブロックが複数含まれる場合、レイアウトでは FCBR ブロックと FST ブロックのみが表示されます。

SUMMARY

```
SUMMARY
```

SUMMARY は、データベースおよび ASSO、DATA および WORK の物理レイアウトに関する情報を提供するものです。

例

```
adarep: summary

Summary of Database 76          28-JUL-2015 12:51:24


DATABASE NAME                    DOKU-DATABASE
DATABASE ID                         76
MAXIMUM NUMBER OF FILES             30
SYSTEM FILES                         1 (CHK),     2 (SEC),     3 (USR)
                                   150 (RBAC)
ACTUAL FILES LOADED                  6
AC SIZE                              3
CURRENT PLOG NUMBER                  8
CURRENT CLOG NUMBER                  0
SECURITY                           ACTIVE


Container  Device      Extents in Blocks    Number of      Block    Total Size
  File     Type          from          to     Blocks        Size     (Megabytes)
-------------------------------------------------------------------------------

 ASSO1     file             1       1,536        1,536      2,048           3.00
 ASSO2     raw          1,537       1,546           10      2,048           0.02

 DATA1     file             1         768          768      4,096           3.00
 DATA2     raw            769         868          100      3,072           0.29
 DATA3     file           869         878           10      6,144           0.06
 DATA4     file           879         888           10      6,144           0.06

 WORK1     file             1       1,365        1,365      3,072           4.00

-------------------------------------------------------------------------------
                                                                           10.43
                                                                           =====
```

デバイスタイプには、raw（RAW セクション）、file（ファイルシステム）、または worm（光学デバイスのように何回でも読み込みは可能だが書き込みは1回に制限されるデバイス）があります。

セキュリティ情報は、データベースセキュリティがアクティブ化されている場合にのみ表示されます。それ以外の場合、情報は表示されません。

RBACシステムファイルは、それが定義されている場合のみ表示されます。それ以外の場合、情報は表示されません。

注意: データベースが READONLY モードで実行されている場合、WORK1 は表示されません。

29 ADASCR（セキュリティ機能）

この章では ADASCR ユーティリティについて説明します。

機能概要

セキュリティユーティリティ ADASCR は、ファイルプロテクションレベルとユーザーパスワードを作成、変更、削除し、個々のパスワードのセキュリティ機能を有効にします。さらに、このユーティリティを使用して、ファイルとパスワードセキュリティ情報を表示することもできます。ADASCR のエクスポート機能の出力を使用して、データベースのすべてのセキュリティ定義から、一部を別のデータベースに適用できます。

このユーティリティへのアクセスは、データベースセキュリティに責任を負う人（DBA）に厳しく制限すべきです。

1 回の ADASCR 実行に複数の機能を指定することができます。指定できる機能の数に制限はありません。

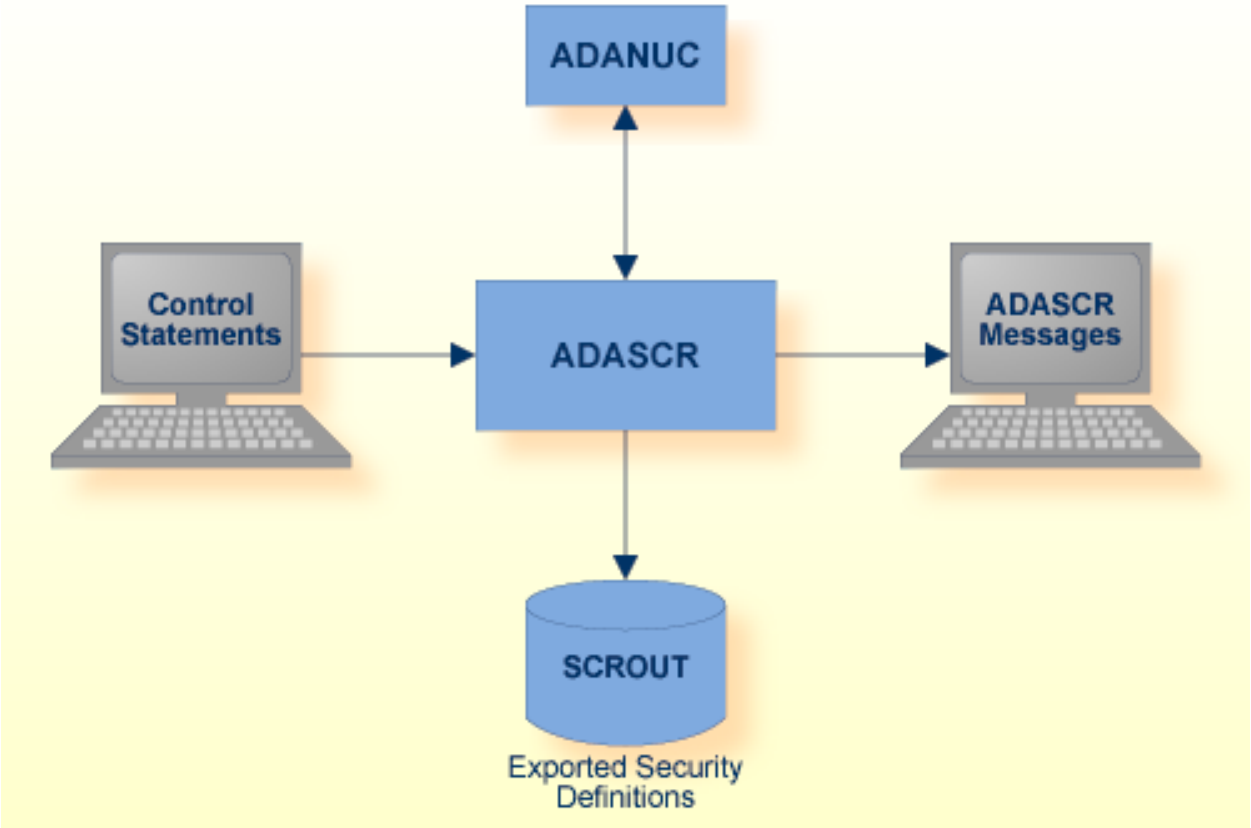
影響を受けるデータベースはオンラインにする必要があります。

ADASCR によって更新された内容はすべてすぐに適用されます。

このユーティリティは多機能ユーティリティです。

注意: あるハードウェアアーキテクチャまたはオペレーティングシステムから別のものに既存のセキュリティ定義をコピーするには、EXPORT 制御パラメータを使用する必要があります。ADAULD/ADADCU および ADACMP/ADAMUP を使って、あるハードウェアアーキテクチャから別のアーキテクチャにセキュリティ定義をコピーすることはできません。これは、セキュリティファイルに格納されるデータが、プラットフォーム依存の方法で格納され、暗号化されていることが理由です。

処理フロー



データセット	論理名／環境変数	記憶媒体	追加情報
コントロールステートメント	stdin/ SYS$INPUT		ユーティリティマニュアル
ADASCR メッセージ	stdout/ SYS$OUTPUT		メッセージおよびコード
エクスポートされたセキュリティ定義	SCROUT	ディスク	ユーティリティマニュアル

チェックポイント

このユーティリティはチェックポイントを書き込みません。

制御パラメータ

次のコントロールパラメータを使用できます。

```
    CHANGE {=|:} (string, string)

M   DBID = number

    DELETE = string

    DISPLAY = [PASSWORDS |
               PERMISSIONS, PASSWORD  {=|:} {* | string} |
               PROTECTIONS,
               FILE = {* | (number[-number][,number[-number]]...)}  |
               VALUE_CRITERIA, PASSWORD  {=|:} {* | string}]

    EXPORT = {PASSWORDS,
D            [TARGET_ARCHITECTURE = KEYWORD,]
               PASSWORD  {=|:} {* | string} |
               PROTECTIONS,
D            [TARGET_ARCHITECTURE = KEYWORD,]
               FILE = {* | (number[-number][,number[-number]]...)}  |
               VALUE_CRITERIA,
D            [TARGET_ARCHITECTURE = KEYWORD,]
               FILE = {* | (number[-number][,number[-number]]...)}
               PASSWORD = {* | string}}

    INSERT {=|:} string, FILE = (number[-number][,number[-number]]...)
             ,ACCESS = (number[,number]...)
             ,UPDATE = (number[,number]...)

    PROTECT = (number[-number][,number[-number]]...)
              ,ACCESS = (number[,number]...)
              ,UPDATE = (number[,number]...)

    SECURITY_BY_VALUE {=|:} string, FILE = number
                         ,ACCESS_CRITERION
                         ,SEARCH_BUFFER = string *
                         ,VALUE_BUFFER = string *
                         ,UPDATE_CRITERION
                         ,SEARCH_BUFFER = string *
                         ,VALUE_BUFFER = string *
```

* サーチ／アクセスバッファ文字列パラメータの後には、途中にコンマを付けずに、すぐに <新規行> を続ける必要があります。

例で使用されている ADASCR 定義

以下の例では、次の ADASCR 定義があらかじめ作成されていることを前提としています。

```
PROTECT=9,ACCESS=9,UPDATE=9
PROTECT=9,NAME=(**,AC,AD,BC),ACCESS=(0,14,14,14),UPDATE=(0,14,14,14)
PROTECT=9,NAME=(FC,FD,IC,ID),ACCESS=(14,14,14,14),UPDATE=(0,0,0,0)
PROTECT=11,ACCESS=1,UPDATE=2
PROTECT=11,NAME=(**,AA,AC,AE),ACCESS=(0,6,3,5),UPDATE=(0,9,9,7)
PROTECT=11,NAME=(AD,AI,AJ,AK),ACCESS=(4,10,10,10),UPDATE=(8,12,12,12)
PROTECT=11,NAME=(AL,AR,AS,AT),ACCESS=(10,7,7,7),UPDATE=(12,9,9,9)
PROTECT=11,NAME=(AX,AY,AZ),ACCESS=(11,11,8),UPDATE=(13,13,10)
PROTECT=(12,13),ACCESS=(12,13),UPDATE=(12,13)
INSERT=000009CD,FILE=(9,12,13),ACCESS=(0,0,0),UPDATE=(13,13,13)
INSERT=00110203,FILE=(11),ACCESS=(2),UPDATE=(3)
INSERT=00110304,FILE=(11),ACCESS=(3),UPDATE=(4)
INSERT=00110506,FILE=(11),ACCESS=(5),UPDATE=(6)
INSERT=00110809,FILE=(11),ACCESS=(8),UPDATE=(9)
INSERT=09CD09CD,FILE=(9,12,13),ACCESS=(13,13,13),UPDATE=(13,13,13)
SECURITY_BY_VALUE=09CD09CD,FILE=9,ACCESS_CRITERION,SEARCH_BUFFER=CA,1,EQ,0,CA,1,EQ.
VALUE_BUFFER=AU
UPDATE_CRITERION,SEARCH_BUFFER=CA,1,EQ.
VALUE_BUFFER=U
```

CHANGE

```
CHANGE {=|:} (string, string)
```

この機能は、既存のパスワードを変更します。

最初の文字列で指定するパスワードは存在しているパスワードにする必要があります。

2 番目の文字列で指定する値は、存在しているパスワードと同じであってはなりません。パスワードは 1～8 文字です。指定された文字列が 8 文字未満の場合、末尾に空白が追加されます。パスワードに特殊文字や空白を含めることはできません。

等号を指定する場合、「文字列」に指定された値が大文字に変換されます。コロンを指定した場合、大文字の変換は実行されません。

最初の文字列で指定したパスワードに有効なすべてのエントリは、新規パスワードに対しても有効です。

例

```
adascr: change:(000009CD,000009cd)
%ADASCR-I-PWCHA, password "000009CD" changed to "000009cd"
```

パスワード OLDPW1 を NEWPW1 に変更します。

DBID

```
DBID = number
```

このパラメータは、現在のデータベースを選択します。

注意: ニュークリアスは実行させる必要があります。

例

```
adascr: dbid=155
```

現在使用されているデータベースはデータベース 155 です。

DELETE

```
DELETE = string
```

このパラメータは、文字列で指定した既存のパスワードを関連する承認レベルおよびセキュリティバイバリュー条件とともに削除します。

例

```
adascr: delete=000009CD
%ADASCR-I-PWDEL, password "000009CD" deleted
```

パスワード USERPW1 を削除します。

DISPLAY

```
DISPLAY = { PASSWORDS |
  PERMISSIONS, PASSWORD {=|:} {* | string} |
  PROTECTIONS, FILE = {* | (number[-number][,number[-number]]...)} |
  VALUE_CRITERIA, PASSWORD {=|:} {* | string}}
```

このパラメータは、ADASCRユーティリティの定義に従って、ファイルについての現在のセキュリティ情報とパスワードを表示します。

プロテクションレベル、パスワード、パスワード承認レベル、およびセキュリティバイバリュー条件の詳細が表示されます。

FILE = {* | number[-number][,number[-number]]...}

FILE パラメータには、先行する DISPLAY 機能が適用されるファイルのリストまたは範囲（複数可）を指定します。

PASSWORDS

PASSWORDS パラメータは、現在セキュリティファイルに格納されているパスワードのリストをアルファベット順に出力します。

例

```
adascr: display=passwords


List of defined passwords for Database 155 ("ALPHA-TS")

FORTYTWO
G6MON
USERPW1
VOYAGER

Total of 4 defined passwords
```

PERMISSIONS

PERMISSIONSパラメータは、パスワード権限情報と現在のファイルプロテクションレベルを照合した結果、現在ロードされているファイルごとの指定パスワードのアクセス権限および更新権限がどうなっているかを出力します。

アクセス権限または更新権限が与えられている場合には、文字 Y で示されます。権限が与えられていない場合には、文字 N で示されます。

ファイルに対するアクセス権限または更新権限が与えられていて、さらにセキュリティバイバリュー制限がそのファイルに適用されている場合はカッコで囲まれ、(Y) で示されます。

ファイルに対してフィールドレベルセキュリティが定義されていて、フィールドへのアクセス権限またはフィールドの更新の権限がファイル権限と異なる場合は、フィールドおよび対応する権限も表示されます。

例

```
adascr: display=permissions,password=09CD09CD

password : "09CD09CD"

 FILE    | ACCESS | UPDATE
   FIELD |        |
-------------------------
    1    |      Y |    Y
    2    |      Y |    Y
    3    |      Y |    Y
    9    |    (Y) |   (Y)
      AC |      N |    N
      AD |      N |    N
      BC |      N |    N
      FC |      N |   (Y)
      FD |      N |   (Y)
      IC |    (Y) |    N
      ID |    (Y) |    N
      CN |      N |
      S2 |      N |
   11    |      N |    N
   12    |      Y |    Y
   13    |      Y |    Y
   14    |      N |    N
-------------------------
```

注意: CN および S2 は照合ディスクリプタおよびスーパーディスクリプタです。派生ディスクリプタの場合は、明示的に更新できないので、更新権限は表示されません。

PROTECTIONS

DISPLAY=PROTECTIONS は、指定したファイルの保護情報を表示します。アクセスまたは更新のプロテクションレベルが0より大きいフィールドについては、プロテクション情報も表示されます。

例：

```
adascr: disp=protections,file=(9-11,14)

 FILE    | ACCESS | UPDATE
   FIELD |        |
-------------------------
    9    |      9 |      9
      AC |     14 |     14
      AD |     14 |     14
      BC |     14 |     14
      FC |     14 |      0
      FD |     14 |      0
```

```
      IC |       0 |      14
      ID |       0 |      14
      CN |      14 |
      S2 |      14 |
   11    |       1 |       2
      AA |       6 |       9
      AC |       3 |       9
      AE |       5 |       7
      AD |       4 |       8
      AI |      10 |      12
      AJ |      10 |      12
      AK |      10 |      12
      AL |      10 |      12
      AR |       7 |       9
      AS |       7 |       9
      AT |       7 |       9
      AX |      11 |      13
      AY |      11 |      13
      AZ |       8 |      10
      PH |       3 |
      S2 |       3 |
      S3 |       7 |
      S4 |       4 |
   14    |       0 |       0
-------------------------
```

注意: ファイル 9、PH、S2、S3、S4 の CN および S2 は照合ディスクリプタ、スーパーディスクリプタ、またはフォネティックディスクリプタです。派生ディスクリプタの場合は、明示的に更新できないので、更新権限は表示されません。

VALUE_CRITERIA

VALUE_CRITERIA パラメータは、指定パスワードに現在定義されているすべてのセキュリティバイバリュー条件を出力します。

例：

```
adascr: display=value_criteria,password=*

There are no value criteria defined for password "000009CD"

There are no value criteria defined for password "00110203"

There are no value criteria defined for password "00110304"

There are no value criteria defined for password "00110506"

There are no value criteria defined for password "00110809"

There are no value criteria defined for password "09CD0000"
```

```
password : "09CD09CD"

| File | Security by Value criterion
|------+-------------------------------------------------------------------------
|    9 | ACCESS_CRITERION
|      |   SEARCH_BUFFER: "CA,1,EQ,0,CA,1,EQ."
|      |   VALUE_BUFFER:  "AU"
|      | UPDATE_CRITERION
|      |   SEARCH_BUFFER: "CA,1,EQ."
|      |   VALUE_BUFFER:  "U"
|-------------------------------------------------------------------------------


Total of 7 defined passwords
```

EXPORT

```
EXPORT = {PASSWORDS,
            [TARGET_ARCHITECTURE = keyword,]
            PASSWORD {=|:} {* | string} |
          PROTECTIONS,
            [TARGET_ARCHITECTURE = keyword,]
            FILE = {* | (number[-number][,number[-number]]...)} |
          VALUE_CRITERIA,
            [TARGET_ARCHITECTURE = keyword,]
            FILE = {* | (number[-number][,number[-number]]...)},
            PASSWORD {=|:} {* | string}}
```

この機能は、現在のセキュリティ設定（パスワード定義、ファイルプロテクションレベル、およびセキュリティバイバリュー条件）を、シーケンシャルファイル SCROUT にエクスポートします。ファイル SCROUT 内の出力は、セキュリティ定義を別のデータベースにインポートするための ADASCR 入力として使用できます。

注意: SCROUT ファイルがすでに存在する場合は、現在のセキュリティ設定が既存のファイルに追加されます。

セキュリティ定義は、メインフレーム構文でエクスポートしてメインフレームプラットフォームにインポートする（TARGET_ARCHITECTURE=MAINFRAME）ことも、オープンシステム構文でオープンシステムプラットフォームにインポートする（TARGET_ARCHITECTURE=OPEN_SYSTEMS - デフォルト）こともできます。

PASSWORDS

```
PASSWORDS,
TARGET_ARCHITECTURE = keyword,
PASSWORD  {=|:} {*|string}
```

PASSWORDS パラメータは、パスワードの権限レベル（アクセスおよび更新）と、指定されたパスワードに関連付けられたファイルまたはファイルリストをエクスポートします。

TARGET_ARCHITECTURE パラメータは、ターゲットプラットフォームの構文を定義します。次のキーワードを使用できます。

キーワード	説明
MAINFRAME	定義したパスワードをメインフレーム構文でエクスポートします。
OPEN_SYSTEMS	定義したパスワードをオープンシステム構文でエクスポートします。

デフォルトの TARGET_ARCHITECTURE は OPEN_SYSTEMS です。

PASSWORD パラメータは、セキュリティ設定をエクスポートする必要があるパスワードを指定します。データベースに定義されているすべてのパスワードをエクスポートすることもできます。これを行うには、このパラメータにアスタリスクを指定します。

例（オープンシステム用のエクスポート）：

```
adascr: export=passwords,target_architecture=open_systems,password=09CD09CD
%ADASCR-I-PWEXP, Password 09CD09CD and its access and update levels successfully ↩
exported.
```

注意: `target_architecture=open_systems` はオプションで、省略できます。

これにより、SCROUT の出力は以下のようになります。

```
INSERT=09CD09CD,FILE=(9,12,13),ACCESS=(13,13,13),UPDATE=(13,13,13)
```

例（メインフレーム用のエクスポート）：

```
adascr: export=passwords,target_architecture=mainframe,password=09CD09CD
%ADASCR-I-PWEXP, Password 09CD09CD and its access and update levels successfully ↩
exported.
```

これにより、SCROUT の出力は以下のようになります。

```
ADASCR INSERT PW=09CD09CD,FILE=9,ACC=13,UPD=13
ADASCR FILE=12,ACC=13,UPD=13
ADASCR FILE=13,ACC=13,UPD=13
```

PROTECTIONS

```
PROTECTIONS,
TARGET_ARCHITECTURE = keyword,
FILE = {*|(number[-number][,number[-number]]…)}
```

PROTECTIONS パラメータは、指定されたファイル、ファイル範囲（複数可）のプロテクションレベルをエクスポートします。

TARGET_ARCHITECTURE パラメータは、ターゲットプラットフォームの構文を定義します。次のキーワードを使用できます。

キーワード	説明
MAINFRAME	定義したパスワードをメインフレーム構文でエクスポートします。
OPEN_SYSTEMS	定義したパスワードをオープンシステム構文でエクスポートします。

デフォルトの TARGET_ARCHITECTURE は OPEN_SYSTEMS です。

FILE パラメータは、プロテクションレベルをエクスポートするファイルを指定します。

例（オープンシステム用のエクスポート）：

```
adascr: export=protections,file=(9,11,13)
%ADASCR-I-PREXP, Protection settings for file 9 successfully exported.
%ADASCR-I-PREXP, Protection settings for file 11 successfully exported.
%ADASCR-I-PREXP, Protection settings for file 13 successfully exported.
```

これにより、SCROUT の出力は以下のようになります。

```
PROTECT=9,ACCESS=9,UPDATE=9
PROTECT=9,NAME=(**,AC,AD,BC),ACCESS=(0,14,14,14),UPDATE=(0,14,14,14)
PROTECT=9,NAME=(FC,FD,IC,ID),ACCESS=(14,14,14,14),UPDATE=(0,0,0,0)
PROTECT=11,ACCESS=1,UPDATE=2
PROTECT=11,NAME=(**,AA,AC,AE),ACCESS=(0,6,3,5),UPDATE=(0,9,9,7)
PROTECT=11,NAME=(AD,AI,AJ,AK),ACCESS=(4,10,10,10),UPDATE=(8,12,12,12)
PROTECT=11,NAME=(AL,AR,AS,AT),ACCESS=(10,7,7,7),UPDATE=(12,9,9,9)
PROTECT=11,NAME=(AX,AY,AZ),ACCESS=(11,11,8),UPDATE=(13,13,10)
PROTECT=13,ACCESS=13,UPDATE=13
PROTECT=13,NAME=(**),ACCESS=(0),UPDATE=(0)
```

例（メインフレーム用のエクスポート）：

```
adascr: export=protections,target_architecture=mainframe,file=11
%ADASCR-I-PREXP, Protection settings for file 11 successfully exported.
```

これにより、SCROUT の出力は以下のようになります。

```
ADASCR PROTECT FILE=11,ACC=1,UPD=2
ADASCR PROTECT FILE=11,NAME=(AA,AC,AE,AD),ACC=(6,3,5,4),UPD=(9,9,7,8)
ADASCR PROTECT FILE=11,NAME=(AF,AG,AH,AI),ACC=(0,0,0,10),UPD=(0,0,0,12)
ADASCR PROTECT FILE=11,NAME=(AJ,AK,AL,AN),ACC=(10,10,10,0),UPD=(12,12,12,0)
ADASCR PROTECT FILE=11,NAME=(AM,AO,AP,AR),ACC=(0,0,0,7),UPD=(0,0,0,9)
ADASCR PROTECT FILE=11,NAME=(AS,AT,AU,AV),ACC=(7,7,0,0),UPD=(9,9,0,0)
ADASCR PROTECT FILE=11,NAME=(AX,AY,AZ),ACC=(11,11,8),UPD=(13,13,10)
```

VALUE_CRITERIA

```
VALUE_CRITERIA,
TARGET_ARCHITECTURE = (keyword [,keyword]),
FILE = {*|(number[-number][,number[-number]]…)},
PASSWORD  {=|:} {*|string}
```

VALUE_CRITERIA パラメータは、「string」で指定されるパスワードに対して、特定のファイルに定義されているセキュリティバイバリュー設定をエクスポートします。

TARGET_ARCHITECTURE パラメータは、ターゲットプラットフォームの構文およびバイト順序を定義します。次のキーワードを使用できます。

キーワードグループ	有効なキーワード
構文	MAINFRAME OPEN_SYSTEMS
バイト順	HIGH_ORDER_BYTE_FIRST LOW_ORDER_BYTE_FIRST

デフォルトの TARGET_ARCHITECTURE は OPEN_SYSTEMS です。

デフォルトのバイト順序は、ADASCR が実行されているマシンのアーキテクチャに対応したものになります。

注意: セキュリティバイバリュー定義をオープンシステムからメインフレームプラットフォームにエクスポートする場合、サーチバッファに W フォーマットのフィールドが含まれていると警告が表示されます。これは、セキュリティバイバリュー定義では、メインフレームプラットフォームで W フォーマットのフィールドがサポートされていないことが理由です。W フォーマットのフィールドを含むサーチバッファはエクスポートされません。

FILE パラメータは、セキュリティバイバリュー定義をエクスポートするファイルのリストまたは範囲（複数可）を指定します。同一のファイル番号を複数回指定することはできません。

PASSWORDパラメータは、セキュリティバイバリュー定義をエクスポートするパスワードを指定します。アスタリスクを指定すると、定義されているすべてのパスワードがエクスポートされます。

例（オープンシステム用のエクスポート）：

```
adascr: export=value_criteria,file=9,PASSWORD=*
There are no value criteria defined for password "000009CD"

There are no value criteria defined for password "00110203"

There are no value criteria defined for password "00110304"

There are no value criteria defined for password "00110506"

There are no value criteria defined for password "00110809"

There are no value criteria defined for password "09CD0000"

%ADASCR-I-SBVEXP, Security by value settings on file 9 for password 09CD09CD succ
essfully exported.
```

これにより、SCROUT の出力は以下のようになります。

```
SECURITY_BY_VALUE=09CD09CD,FILE=9,ACCESS_CRITERION,SEARCH_BUFFER=CA,1,EQ,0,CA,1,E
Q.
VALUE_BUFFER=AU
UPDATE_CRITERION,SEARCH_BUFFER=CA,1,EQ.
VALUE_BUFFER=U
```

例（メインフレーム用のエクスポート）：

```
adascr: export=protections,target_architecture=mainframe,file=11
%ADASCR-I-PREXP, Protection settings for file 11 successfully exported.
```

これにより、SCROUT の出力は以下のようになります。

```
ADASCR PROTECT FILE=11,ACC=1,UPD=2
ADASCR PROTECT FILE=11,NAME=(AA,AC,AE,AD),ACC=(6,3,5,4),UPD=(9,9,7,8)
ADASCR PROTECT FILE=11,NAME=(AF,AG,AH,AI),ACC=(0,0,0,10),UPD=(0,0,0,12)
ADASCR PROTECT FILE=11,NAME=(AJ,AK,AL,AN),ACC=(10,10,10,0),UPD=(12,12,12,0)
ADASCR PROTECT FILE=11,NAME=(AM,AO,AP,AR),ACC=(0,0,0,7),UPD=(0,0,0,9)
ADASCR PROTECT FILE=11,NAME=(AS,AT,AU,AV),ACC=(7,7,0,0),UPD=(9,9,0,0)
ADASCR PROTECT FILE=11,NAME=(AX,AY,AZ),ACC=(11,11,8),UPD=(13,13,10)
```

Windows でエクスポートして UNIX でインポートする例

この例では、後で UNIX Adabas データベース 34 にインポートするために、Windows Adabas データベース 33 の既存のセキュリティ設定をエクスポートする方法を示します。

ADASCR のエクスポート機能は次のように使用されます。

```
>adascr db=33 export=passwords,password=*
%ADASCR-I-STARTED,      12-JAN-2017 16:17:29, Version 6.5.1.0 (Windows 64Bit)
%ADASCR-I-DBON, database 6 accessed online
%ADASCR-I-PWEXP, Password 000009CD and its access and update levels successfully
exported.
%ADASCR-I-PWEXP, Password 00110203 and its access and update levels successfully
exported.
%ADASCR-I-PWEXP, Password 00110304 and its access and update levels successfully
exported.
%ADASCR-I-PWEXP, Password 00110506 and its access and update levels successfully
exported.
%ADASCR-I-PWEXP, Password 00110809 and its access and update levels successfully
exported.
%ADASCR-I-PWEXP, Password 09CD0000 and its access and update levels successfully
exported.
%ADASCR-I-PWEXP, Password 09CD09CD and its access and update levels successfully
exported.

%ADASCR-I-TERMINATED,   12-JAN-2017 16:17:30, elapsed time: 00:00:01

>adascr db=33 export=protections,file=(9,11)
%ADASCR-I-STARTED,      12-JAN-2017 16:17:55, Version 6.5.1.0 (Windows 64Bit)
%ADASCR-I-DBON, database 6 accessed online
%ADASCR-I-PREXP, Protection settings for file 9 successfully exported.
%ADASCR-I-PREXP, Protection settings for file 11 successfully exported.

%ADASCR-I-TERMINATED,   12-JAN-2017 16:17:55, elapsed time: 00:00:00

D:\ada_build\ada\v6_5>adascr db=33 export=value_criteria,file=9,password=*
%ADASCR-I-STARTED,      12-JAN-2017 16:18:30, Version 6.5.1.0 (Windows 64Bit)
%ADASCR-I-DBON, database 33 accessed online
adascr: target_architecture=(open_systems,high_order_byte_first)
adascr: file=9
adascr: password=09CD09CD
%ADASCR-I-SBVEXP, Security by value settings on file 9 for password 09CD09CD succ
essfully exported.
adascr: q
%ADASCR-I-TERMINATED,   12-JAN-2017 16:19:46, elapsed time: 00:01:16
```

ここで、ファイル scrout.txt を必要に応じて編集し、指定されたパスワードでエクスポートされたファイル以外のファイルを保護できます。

ここで、エクスポートされたテキストファイル scrout.txt を目的のターゲットプラットフォーム（この場合は UNIX プラットフォーム）にコピーします。次のステートメントで ADASCR を使用し、エクスポートされたセキュリティ設定をインポートします。

```
>adascr db=34 + < scrout.txt
%ADASCR-I-STARTED,      12-JAN-2017 17:16:59, Version 6.5.1.0 (Solaris 64Bit)
%ADASCR-I-DBON, database 34 accessed online
%ADASCR-I-PWINS, password "000009CD" inserted
%ADASCR-I-PWINS, password "00110203" inserted
%ADASCR-I-PWINS, password "00110304" inserted
%ADASCR-I-PWINS, password "00110506" inserted
%ADASCR-I-PWINS, password "00110809" inserted
%ADASCR-I-PWINS, password "09CD0000" inserted
%ADASCR-I-PWINS, password "09CD09CD" inserted
%ADASCR-I-FILPRO, protections (access 9, update 9) set for file 9
%ADASCR-I-FIELDPRO, protections (access 0, update 0) set for field ** in file 9
%ADASCR-I-FIELDPRO, protections (access 14, update 14) set for field AC in file 9

%ADASCR-I-FIELDPRO, protections (access 14, update 14) set for field AD in file 9

%ADASCR-I-FIELDPRO, protections (access 14, update 14) set for field BC in file 9

%ADASCR-I-FIELDPRO, protections (access 14, update 0) set for field FC in file 9
%ADASCR-I-FIELDPRO, protections (access 14, update 0) set for field FD in file 9
%ADASCR-I-FIELDPRO, protections (access 14, update 0) set for field IC in file 9
%ADASCR-I-FIELDPRO, protections (access 14, update 0) set for field ID in file 9
%ADASCR-I-FILPRO, protections (access 1, update 2) set for file 11
%ADASCR-I-FIELDPRO, protections (access 0, update 0) set for field ** in file 11
%ADASCR-I-FIELDPRO, protections (access 6, update 9) set for field AA in file 11
%ADASCR-I-FIELDPRO, protections (access 3, update 9) set for field AC in file 11
%ADASCR-I-FIELDPRO, protections (access 5, update 7) set for field AE in file 11
%ADASCR-I-FIELDPRO, protections (access 4, update 8) set for field AD in file 11
%ADASCR-I-FIELDPRO, protections (access 10, update 12) set for field AI in file 1
1
%ADASCR-I-FIELDPRO, protections (access 10, update 12) set for field AJ in file 1
1
%ADASCR-I-FIELDPRO, protections (access 10, update 12) set for field AK in file 1
1
%ADASCR-I-FIELDPRO, protections (access 10, update 12) set for field AL in file 1
1
%ADASCR-I-FIELDPRO, protections (access 7, update 9) set for field AR in file 11
%ADASCR-I-FIELDPRO, protections (access 7, update 9) set for field AS in file 11
%ADASCR-I-FIELDPRO, protections (access 7, update 9) set for field AT in file 11
%ADASCR-I-FIELDPRO, protections (access 11, update 13) set for field AX in file 1
1
%ADASCR-I-FIELDPRO, protections (access 11, update 13) set for field AY in file 1
1
%ADASCR-I-FIELDPRO, protections (access 8, update 10) set for field AZ in file 11

%ADASCR-I-SEVINS, Value criteria for file 9 added to password "09CD09CD"
```

```
%ADASCR-I-TERMINATED,   12-JAN-2017 17:17:00, elapsed time: 00:00:01
```

ADASCR の DISPLAY 制御パラメータを使用して、インポートされたセキュリティ設定を確認できます。

```
adascr
adascr: dbid=34 display=passwords

List of defined passwords for Database 34 ("GENERAL_DATABASE")

MYSECRET

Total of 1 defined password

adascr: dbid=34 display=permissions,password=MYSECRET

password : "MYSECRET"


 FILE | ACCESS | UPDATE
-----------------------
    1 |      Y  |    Y
    2 |      Y  |    Y
    3 |      Y  |    Y
    9 |    (Y) |   (Y)
   11 |      N  |    N
   12 |    (Y) |   (Y)
   13 |    (Y) |   (Y)
   14 |      N  |    N
-----------------------

( ) Further value restrictions apply where brackets shown.


adascr: db=34 display=protections,file=(9,12,13)


FILE | ACCESS | UPDATE
-----------------------
   9 |      7 |     11
  12 |      2 |      2
  13 |      4 |      4
-----------------------

adascr: db=34 display=value_criteria,password=MYSECRET


password : "MYSECRET"

| File | Security by Value criterion
```

```
|------+-------------------------------------------------------------------------
|    9 | ACCESS_CRITERION
|      |   SEARCH_BUFFER: "AZ,2,GE."
|      |   VALUE_BUFFER:  "FR"
|      | UPDATE_CRITERION
|      |   SEARCH_BUFFER: "AH,8,GE,D,AH,8,LE,D,AA,8,LE. "
|      |   VALUE_BUFFER:  "19520101195202016000000O"
|   12 | ACCESS_CRITERION
|      |   SEARCH_BUFFER: "AM,3,GE."
|      |   VALUE_BUFFER:  0x33000C
|      | UPDATE_CRITERION
|      |   SEARCH_BUFFER: "AM,3,GE."
|      |   VALUE_BUFFER:  0x50000C
|   13 | ACCESS_CRITERION
|      |   SEARCH_BUFFER: "B0,GE."
|      |   VALUE_BUFFER:  0x01000000
|      | UPDATE_CRITERION
|      |   SEARCH_BUFFER: "B0,3,GE."
|      |   VALUE_BUFFER:  0xFF0201
|------+-------------------------------------------------------------------------
```

INSERT

```
INSERT {=|:} string, FILE = (number[-number][,number[-number]]...)
         ,ACCESS = (number[,number]...)
         ,UPDATE = (number[,number]...)
```

この機能は、"string" で指定したパスワードをパスワードテーブルに挿入します。

等号を指定する場合、「文字列」に指定された値が大文字に変換されます。コロンを指定した場合、大文字の変換は実行されません。

パスワードは 1～8 文字です。指定された文字列が 8 文字未満の場合、末尾に空白が追加されます。パスワードに特殊文字や空白を含めることはできません。

ACCESS = (number[,number]...)

ACCESS パラメータは、FILE パラメータで指定したファイルやファイル範囲に関連するアクセスプロテクションレベルを指定します。各プロテクションレベルは1つのファイルまたは1つのファイル範囲に対応します。指定可能な値の範囲は0～14です。プロテクションレベルは、FILE パラメータの対応するファイルまたはファイル範囲と同じ順序で指定する必要があります。

FILE = (number[-number[,number[-number]]...)

FILE パラメータは、権限レベルを付与するファイルのリストまたは範囲を指定します。同一のファイル番号を複数回指定することはできません。

UPDATE = (number[,number]...)

UPDATE パラメータは、FILE パラメータに指定したファイルやファイル範囲に関連する更新プロテクションレベルを指定します。各プロテクションレベルは1つのファイルまたは1つのファイル範囲に対応します。指定可能な値の範囲は 0～14 です。プロテクションレベルは、FILE パラメータの対応するファイルまたはファイル範囲と同じ順序で指定する必要があります。

例

```
adascr: insert=userpwx, file=(1,2,3),
                  access=(7,7,7), update=(0,8,8)

adascr: insert=userpwy, file=(1,2,3),
                  access=(7,2,2), update=(8,0,0)

adascr: insert=userpwz, file=(1-3,5),
                  access=(2,3), update=(4,6)
```

PROTECT

```
PROTECT = (number[-number][,number[-number]]...)
          [,NAME = (field_name [, field_name] …)]
          ,ACCESS = (number[,number]...)
          ,UPDATE = (number[,number]...)
```

この機能は、指定したファイルまたはファイル内のフィールドのアクセスプロテクションレベルや更新プロテクションレベルを挿入（または更新）します。

- NAME が指定されていない場合は、ファイルのプロテクションレベルが定義されます。同一のファイル番号を複数回指定することはできません。
- NAMEが指定されている場合は、フィールドのプロテクションレベルが定義されます。NAMEの指定は、単一のファイル番号が指定されている場合にのみ許可されます。同一のフィールド名は複数回指定できません。グループまたはピリオディックグループの名前が指定されている場合は、指定されたフィールドプロテクションレベルが、フィールド名のリストで指定されていないグループ／ピリオディックグループのすべてのコンポーネントフィールドに設定されます。グループまたはピリオディックグループのコンポーネントとして明示的にも暗黙的にも指定されていないフィールドの場合、フィールドプロテクションレベルは変更されません。
- ファイルおよびフィールドのプロテクションレベルのデフォルトは、アクセスプロテクションレベル 0 および更新プロテクションレベル 0 です。

ACCESS = (number[,number]...)

指定可能な値の範囲は 0～15 です。

- NAME が指定されていない場合、UPDATE パラメータは、PROTECT パラメータで指定したファイルに関連付けられる更新プロテクションレベルを指定します。各プロテクションレベルは1つのファイルまたは1つのファイル範囲に対応します。プロテクションレベルは、PROTECT パラメータの対応するファイルまたはファイル範囲と同じ順序で指定する必要があります。指定する値の数は、PROTECT パラメータ内のファイルまたはファイル範囲の数と同じになっている必要があります。
- NAME が指定されている場合、UPDATE パラメータは、NAME で指定したフィールドに関連付けられる更新プロテクションレベルを指定します。各プロテクションレベルは1つのフィールド名に対応します。プロテクションレベルをファイルのプロテクションレベルよりも低く設定すると、ファイルプロテクションレベルが実際のフィールドプロテクションレベルになります。プロテクションレベルは、NAME パラメータの対応するフィールド名と同じ順序で指定する必要があります。指定する値の数は、NAMEパラメータのフィールド名の数と同じになっている必要があります。

INSERT機能で使用される更新プロテクションレベルの最大値は14であるのに対して、PROTECT パラメータでは最大値は 15 です。したがって、UPDATE=15 を指定したファイルまたはフィールドは、いかなるユーザーも更新できなくなります。

UPDATE = (number[,number]...)

指定可能な値の範囲は 0～15 です。

- NAME が指定されていない場合、ACCESS パラメータは、PROTECT パラメータで指定したファイルに関連付けられるアクセスプロテクションレベルを指定します。各プロテクションレベルは 1 つのファイルまたは 1 つのファイル範囲に対応します。プロテクションレベルは、PROTECTパラメータの対応するファイルまたはファイル範囲と同じ順序で指定する必要があります。指定する値の数は、PROTECTパラメータ内のファイルまたはファイル範囲の数と同じになっている必要があります。
- NAMEが指定されている場合、ACCESSパラメータは、NAMEで指定したフィールドに関連付けられるアクセスプロテクションレベルを指定します。各プロテクションレベルは 1 つのフィールド名に対応します。プロテクションレベルをファイルのプロテクションレベルよりも低く設定すると、ファイルプロテクションレベルが実際のフィールドプロテクションレベルになります。プロテクションレベルは、NAME パラメータの対応するフィールド名と同じ順序で指定する必要があります。指定する値の数は、NAME パラメータのフィールド名の数と同じになっている必要があります。

PROTECT 機能で使用されるアクセスプロテクションレベルの最大値は 15 であるのに対して、INSERT 機能では最大値は 14 です。したがって、ACCESS=15 を指定したファイルまたはフィールドには、いかなるユーザーもアクセスできなくなります。

例

```
adascr: protect=25,access=7,update=11
adascr: protect=25,name=(AA,AB,AC),access=(6,8,7),update=(14,13,12)
adascr: protect=(1,2,3),access=(7,7,7),update=(8,8,8)
adascr: protect=(4-6),access=10,update=12
```

2 番目の例で、AA はエレメンタリフィールド、AB はエレメンタリフィールド AC、AD、AE で構成されるグループであると仮定します。このとき、プロテクションレベルは次の表のようになります。

	フィールド AA	グループ AB		
		フィールド AC	フィールド AD	フィールド AE
アクセスプロテクションレベル	6	7	8	8
更新プロテクションレベル	14	12	13	13

ADADBM ユーティリティの RENUMBER 機能を実行すると、セキュリティ保護されたファイルのファイル番号が結果的に変更される場合、そのファイルのセキュリティプロテクションレベルを再設定するために PROTECT 機能を再実行する必要があります。パスワードも、変更前のファイル番号に関連付けられているため、再設定する必要があります。

SECURITY_BY_VALUE

```
SECURITY_BY_VALUE {=|:} string, FILE = number
                     ,ACCESS_CRITERION
                     ,SEARCH_BUFFER = string *
                     ,VALUE_BUFFER = string *
                     ,UPDATE_CRITERION
                     ,SEARCH_BUFFER = string *
                     ,VALUE_BUFFER = string *
```

* サーチ／アクセスバッファ文字列パラメータの後には、途中にコンマを付けずに、すぐに <新規行> を続ける必要があります。

この機能は、"string" で指定したパスワードに対する特定のファイルのセキュリティバイバリューを挿入（または更新）します。パスワードは、INSERT 機能を使用して事前にセキュリティファイルに挿入されている必要があります。各パスワードは、最高 99 ファイルに対して定義されたセキュリティバイバリュー条件を持つことができます。

等号を指定する場合、「文字列」に指定された値が大文字に変換されます。コロンを指定した場合、大文字の変換は実行されません。

ACCESS_CRITERION

指定したパスワードを使ってデータへのアクセスを制限する場合には、ACCESS_CRITERION キーワードをサーチバッファおよびバリューバッファの前に指定し、制限する条件を定義する必要があります。

アクセスバリュー条件を指定するために、ファイルの ACCESS パスワード権限レベル（つまりゼロ以外）も事前に設定しておく必要があります。レベルを設定していない場合、ACCESS_CRITERION キーワードを指定できません。

FILE = number

FILE パラメータは、バリュー条件を定義するファイルを指定します。必ず 1 ファイルだけを指定します。また、そのファイルはデータベースに現在ロードされている必要があります。

UPDATE_CRITERION

指定したパスワードを使ってデータの更新を制限する場合には、UPDATE_CRITERION キーワードをサーチバッファおよびバリューバッファの前に指定し、制限する条件を定義する必要があります。

更新バリュー条件を指定するために、ファイルの UPDATE パスワード権限レベル（つまりゼロ以外）も事前に設定しておく必要があります。レベルを設定していない場合、UPDATE_CRITERION キーワードを指定できません。

SEARCH_BUFFER = string

SEARCH_BUFFER パラメータを使用して、アクセス／更新条件に対する検索式を指定します。サーチバッファの指定例および構文については、『*コマンドリファレンスマニュアル*』の「*Adabas の呼び出し*」の「*サーチバッファとバリューバッファ*」を参照してください。

セキュリティバイバリュー条件に使用する場合、一定の制限がサーチバッファに適用され、ソフトカップリング、サブディスクリプタ、スーパーディスクリプタ、ハイパーディスクリプタ、およびフォネティックディスクリプタはサポートされなくなります。

必要な条件が、制限の適用が不要なものである場合、関連するサーチバッファには次のように、終端を表す文字だけを指定してください。

```
SEARCH_BUFFER = .
```

この場合、VALUE_BUFFER パラメータは必要ないので、指定しないでください。

VALUE_BUFFER = string

VALUE_BUFFERパラメータを使用して、先行するサーチバッファに指定したアクセス／更新条件の検索式に対応する値を指定します。文字列は、英数字文字列として、または 16 進表記の文字列として直接指定することができます。

例

```
adascr: security_by_value=fortytwo, file=10,
        access_criterion, search_buffer=CA,4,U,LT.
adascr: value_buffer=1707
adascr: update_criterion, search_buffer=A,4,S,CA,4,D,AA,O,AA.
adascr: value_buffer=01001599MS
```

```
adascr: security_by_value=g6mon, file=3
adascr: access_criterion, search_buffer=.
adascr: update_criterion, search_buffer=AC,3.
adascr: value_buffer=HAM
```

指定したファイルのアクセスまたは更新プロテクションレベルがゼロの場合、そのファイルのレコードをアクセス／更新するためにパスワードが使用されても、関連するバリュー条件はテストされません。

30 ADATST（Adabas コマンドの発行）

この章では ADATST ユーティリティについて説明します。

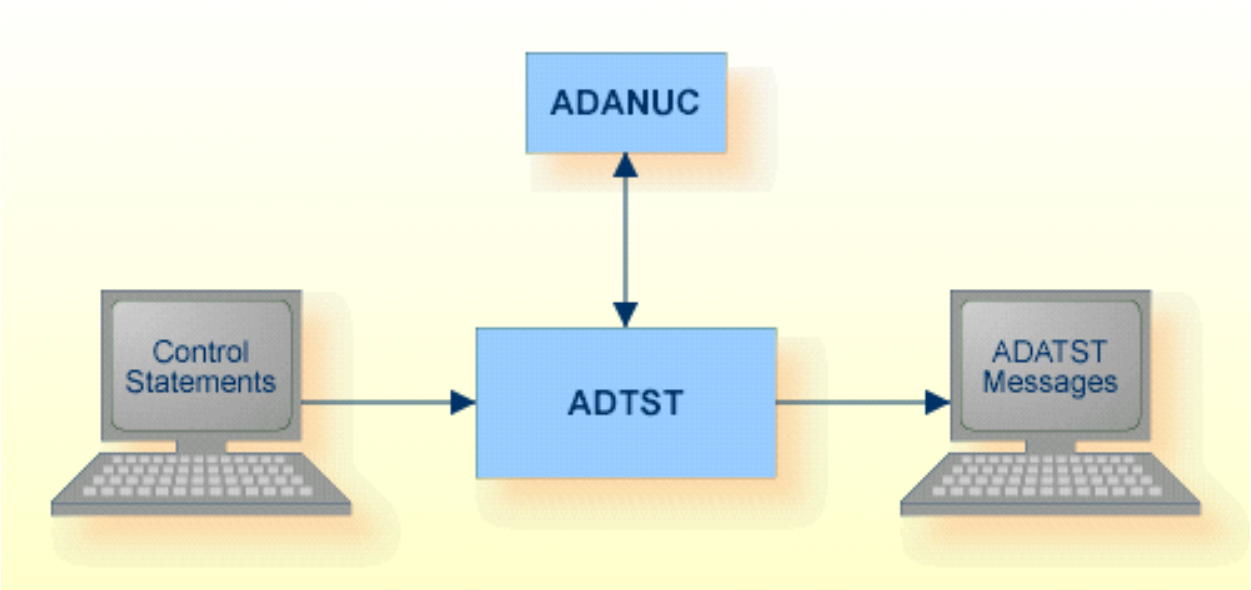
機能概要

ADATSTユーティリティは、Adabasコマンドを発行するためのコントロールブロックおよび必要なバッファを設定するために使用します。

古い ACB Adabas コマンドインターフェイスと新しい ACBX コマンドインターフェイスの両方がサポートされます。コマンドインターフェイスの詳細については、『*コマンドリファレンス*』を参照してください。ACB インターフェイスは、ACBX インターフェイスのサブセットと見なすことができます。古い ACB Adabas コントロールブロック内のフィールドは、新しい ACBX コントロールブロック（または ABD、Adabas バッファ記述）にも含まれますが、一部の ACB フィールドは、新しいインターフェイスの対応するフィールドよりも小さいという違いがあります。特にバッファの長さが拡大され、現在では 64 KB より大きい Adabas バッファが可能です。インターフェイス間の切り替えが可能ですが、古い ACB インターフェイスに戻すときは、古い ACB インターフェイスの制約事項に配慮する必要があります。

このユーティリティは多機能ユーティリティです。

処理フロー



データセット	環境変数／論理名	記憶媒体	追加情報
コントロールステートメント	stdin/ SYS$INPUT		ユーティリティマニュアル
ADATST メッセージ	stdout/ SYS$OUTPUT		メッセージおよびコード

チェックポイント

このユーティリティはチェックポイントを書き込みません。

制御パラメータ

注意: 次の中で使用される "string" は、ASCII 文字列か、0x を伴う 16 進です。

次のコントロールパラメータを使用できます。

```
     A1 {=|:} string

     A2 {=|:} string

     A3 {=|:} string

     A4 {=|:} string

     A5 {=|:} string

     A6 {=|:} string

     ABD

     ALOOP [= number]

     CB

     CBDUMP

     CC = string

     CID = string

     CO1 = string

     CO2 = string

     CO3 = string

     CO4 = string

     CO5 = string

     CO6 = string

     CO7 = string

     CO8 = string

M    DBID = number
```

```
      DLOOP

      ELOOP

      ERRORS = number

      EXECUTE = { number |  ISNQ }

      FB [{=|:} string]

      FB2  [{=|:} string]

      FB3  [{=|:} string]

      FBL = number

      FB2L [= number]

      FB3L [= number]

      FILE = number

      GO  [= { number | ISNQ } ]

      IB [= (number [,number]...)]

      IBL = number

D     INTERFACE = keyword

      ISN = number

      ISND = number

      ISNI = number

      ISNL = { number |  ISN }

      ISNQ = number

      LOOP

      MB = (number_buffers, number_isns)

D     [NO]OUTPUT

      OVERWRITE_RB = string

      OVERWRITE_RB2 = string

      OVERWRITE_RB3 = string
```

```
      RB [{=|:} string]
      RB2 [{=|:} string]
      RB3 [{=|:} string]
      RBL = number
      RB2L [= number]
      RB3L [= number]
      READ_RB = string
      READ_RB2 = string
      READ_RB3 = string
      RESPONSE = number
      SB [{=|:} string]
      SBL = number
      SET_PWD = string
      SET_UID = string
      TIME
D   [NO]TRACE
      VB [{=|:} string]
      VBL = number
D   WAIT [= [time]]
      WRITE_RB = string
      WRITE_RB2 = string
      WRITE_RB3 = string
```

A1

```
A1 {=|:} string
```

このパラメータは、アディション 1 フィールドを設定します。

等号を指定する場合、「文字列」に指定された値が大文字に変換されます。コロンを指定した場合、大文字の変換は実行されません。

A2

```
A2 {=|:} string
```

このパラメータは、アディション 2 フィールドを設定します。

等号を指定する場合、「文字列」に指定された値が大文字に変換されます。コロンを指定した場合、大文字の変換は実行されません。

A3

```
A3 {=|:} string
```

このパラメータは、アディション 3 フィールドを設定します。

等号を指定する場合、「文字列」に指定された値が大文字に変換されます。コロンを指定した場合、大文字の変換は実行されません。

A4

```
A4 {=|:} string
```

このパラメータは、アディション 4 フィールドを設定します。

等号を指定する場合、「文字列」に指定された値が大文字に変換されます。コロンを指定した場合、大文字の変換は実行されません。

A5

```
A5 {=|:} string
```

このパラメータは、アディション 5 フィールドを設定します。

等号を指定する場合、「文字列」に指定された値が大文字に変換されます。コロンを指定した場合、大文字の変換は実行されません。

A6

```
A6 {=|:} string
```

このパラメータは、アディション 6 フィールドを設定します。

等号を指定する場合、「文字列」に指定された値が大文字に変換されます。コロンを指定した場合、大文字の変換は実行されません。

ABD

```
ABD
```

このパラメータは、INTERFACE=ACBX の指定後に限って使用できます。現在定義されている Adabas バッファの Adabas バッファ定義が表示されます。

ALOOP

```
ALOOP [= number]
```

このパラメータは、行を何行でも追加できるようにループを開始します。"number" には行番号を指定します。行番号が指定されると、その位置から既存行を上書きする形で新規行が追加されます。行番号が指定されない場合、新規行は最後に追加されます。このループは ELOOP があるところまで続きます。

CB

```
CB
```

この機能は、コントロールブロックの内容を表示します。

パラメータ INTERFACE=ACB を指定した場合は、古い ACB Adabas コントロールブロックに含まれるフィールドのみが表示されます。

パラメータ INTERFACE=ACBX を指定した場合は、新しい ACBX Adabas コントロールブロックに含まれるフィールドだけでなく、古い ACB Adabas コントロールブロックに含まれているフィールドも表示されます。

CBDUMP

```
CBDUMP
```

この機能は、コントロールブロックを 16 進形式でダンプ出力します。

CC

```
CC = string
```

このパラメータは、コマンドコードを指定するものです。

CID

```
CID= string
```

このパラメータは、コマンド ID を指定するものです。

CO1

```
CO1 = string
```

このパラメータは、コマンドオプション 1 を設定します。

CO2

```
CO2 = string
```

このパラメータは、コマンドオプション 2 を設定します。

CO3

```
CO3 = string
```

このパラメータは、コマンドオプション 3 を設定します。

CO4

```
CO4 = string
```

このパラメータは、コマンドオプション 4 を設定します。

CO5

```
CO5 = string
```

このパラメータは、コマンドオプション 5 を設定します。

CO6

```
CO6 = string
```

このパラメータは、コマンドオプション 6 を設定します。

CO7

```
CO7 = string
```

このパラメータは、コマンドオプション 7 を設定します。

CO8

```
CO8 = string
```

このパラメータは、コマンドオプション 8 を設定します。

DBID

```
DBID = number
```

このパラメータは使用するデータベースを指定します。

DLOOP

```
DLOOP
```

この機能は、蓄積されたコマンドループを表示します。

ELOOP

```
ELOOP
```

この機能は、ループを終了させます。

ERRORS

```
ERRORS = number
```

このパラメータは、強制的に終了されるまでのエラー許容数を指定するものです。

EXECUTE

```
EXECUTE = { number |  ISNQ }
```

このパラメータは、n 回分のループを実行するためのものです。n の部分には "number" または ISNQ を指定します。ループを終了するには、Ctrl キーを押したまま C キーを押します。

FB

```
FB [{=|:} string]
```

このパラメータは、最初のフォーマットバッファの表示またはフォーマットバッファへのデータ入力に使用します。長さは暗黙的に設定されます。

FB2

```
FB2 [{=|:} string]
```

このパラメータは、第2フォーマットバッファの表示またはフォーマットバッファへのデータ入力に使用します。長さは暗黙的に設定されます。

FB3

```
FB3 [{=|:} string]
```

このパラメータは、第3フォーマットバッファの表示またはフォーマットバッファへのデータ入力に使用します。長さは暗黙的に設定されます。

FBL

```
FBL = number
```

このパラメータは、コントロールブロック内の最初のフォーマットバッファ長を定義するものです。

FB2L

```
FB2L = number
```

このパラメータは、コントロールブロック内の2番目のフォーマットバッファ長を定義するものです。

FB3L

```
FBL3 = number
```

このパラメータは、コントロールブロック内の3番目のフォーマットバッファ長を定義するものです。

FILE

```
FILE = number
```

このパラメータは、ファイル番号を指定するものです。

GO

```
GO [= { number |  ISNQ } ]
```

この機能は、Adabas を1回コールするか、n 回コールするもので、n の部分には "number" または ISNQ を指定します。ループを終了するには、Ctrl キーを押したまま C キーを押します。

IB

```
IB [= (number [,number]...)]
```

このパラメータは、ISN バッファの表示または ISN の入力に使用します。長さは暗黙的に設定されます。

IBL

```
IBL = number
```

このパラメータは、コントロールブロック内の ISN バッファ長を指定するものです。

INTERFACE

```
INTERFACE = keyword
```

このパラメータは、古い Adabas コマンドインターフェイスと新しい Adabas コマンドインターフェイスを切り替えるために使用します。有効なキーワードは ACB と ACBX です。デフォルトは ACB です。INTERFACE = ACB を指定すると、Adabas コールは古い ACB Adabas インターフェイスで実行されます。新しい ACBX Adabas コントロールブロックのみに含まれるフィールド、および増設のフォーマットバッファとレコードバッファは無視されます。古い Adabas インターフェイスと新しい Adabas インターフェイスの切り替えは、その Adabas セッションのみで有効です。

ISN

```
ISN = number
```

このパラメータは、数値指定により ISN を設定します。

ISND

```
ISND = number
```

このパラメータは ISN から "number" を減算します。

ISNI

```
ISNI = number
```

このパラメータは ISN に "number" を加算します。

ISNL

```
ISNL = { number |  ISN }
```

このパラメータは、指定数値で ISN 下限値を設定するか、コントロールブロックから ISN 下限値内へ ISN を移動するのに使用します。

ISNQ

```
ISNQ = number
```

このパラメータは、指定数値により ISN 量を指定するものです。

LOOP

```
LOOP
```

この機能は、ループの開始を定義するものです。後続のコマンドはすべてELOOPが入力されるまで蓄積されます。

MB

```
MB = (number_buffers, number_isns)
```

このパラメータは、マルチフェッチバッファの数、およびマルチフェッチバッファに保存できるIS エントリの数を定義します。number_buffers は 0 から 3 の数値にすることができます。number_isns は 0 より大きい数値にする必要があります。

MB パラメータは、INTERFACE = ACBX の指定後に限って指定できます。マルチフェッチバッファは純粋な出力バッファなので、マルチフェッチバッファに情報を入力することはできません。

マルチフェッチバッファを指定したら、RB、RB2、RB3 のパラメータでその内容を表示できます。

[NO]OUTPUT

```
[NO]OUTPUT
```

このオプションをNOOUTPUTに設定すると、Adabasがn回コールされた場合にメッセージが表示されません。エラーメッセージだけが出力されます。

デフォルトは OUTPUT です。

OVERWRITE_RB

```
OVERWRITE_RB = string
```

このパラメータは、最初のレコードバッファの内容を書き込む既存ファイルの名前を指定します。ファイルの現在の内容は上書きされます。

OVERWRITE_RB2

```
OVERWRITE_RB2 = string
```

このパラメータは、2番目のレコードバッファの内容を書き込む既存ファイルの名前を指定します。ファイルの現在の内容は上書きされます。

OVERWRITE_RB3

```
OVERWRITE_RB3 = string
```

このパラメータは、3番目のレコードバッファの内容を書き込む既存ファイルの名前を指定します。ファイルの現在の内容は上書きされます。

RB

```
RB [{=|:} string]
```

このパラメータは、最初のレコードバッファの表示か、または最初のレコードバッファへのデータ入力に使用します。ファイルに入力する場合、長さは暗黙的に設定されます。

等号を指定する場合、「文字列」に指定された値が大文字に変換されます。コロンを指定した場合、大文字の変換は実行されません。

最初のレコードバッファを表示し、最低 1 つのマルチフェッチバッファが MB パラメータで定義されている場合は、最初のマルチフェッチバッファも表示されます。

RB2

```
RB2 [{=|:} string]
```

このパラメータは、2 番目のレコードバッファの表示か、または 2 番目のレコードバッファへのデータ入力に使用します。ファイルに入力する場合、長さは暗黙的に設定されます。

等号を指定する場合、「文字列」に指定された値が大文字に変換されます。コロンを指定した場合、大文字の変換は実行されません。

2 番目のレコードバッファを表示し、最低 2 つのマルチフェッチバッファが MB パラメータで定義されている場合は、2 番目のマルチフェッチバッファも表示されます。

RB3

```
RB3 [{=|:} string]
```

このパラメータは、3 番目のレコードバッファの表示か、または 3 番目のレコードバッファへのデータ入力に使用します。ファイルに入力する場合、長さは暗黙的に設定されます。

等号を指定する場合、「文字列」に指定された値が大文字に変換されます。コロンを指定した場合、大文字の変換は実行されません。

3 番目のレコードバッファを表示し、最低 3 つのマルチフェッチバッファが MB パラメータで定義されている場合は、3 番目のマルチフェッチバッファも表示されます。

RBL

```
RBL = number
```

このパラメータは、コントロールブロック内の最初のレコードバッファ長を指定するものです。

RB2L

```
RB2L = number
```

このパラメータは、コントロールブロック内の 2 番目のレコードバッファ長を指定するものです。

RB3L

```
RB3L = number
```

このパラメータは、コントロールブロック内の 3 番目のレコードバッファ長を指定するものです。

READ_RB

```
READ_RB = string
```

このパラメータは、最初のレコードバッファに読み込むファイルの名前を指定します。

READ_RB2

```
READ_RB2 = string
```

このパラメータは、2 番目のレコードバッファに読み込むファイルの名前を指定します。

READ_RB3

```
READ_RB3 = string
```

このパラメータは、3 番目のレコードバッファに読み込むファイルの名前を指定します。

RESPONSE

```
RESPONSE = number
```

このパラメータは、指定ニュークリアスレスポンスコードに対するエラーテキストを表示します。

SB

```
SB [{=|:} string]
```

このパラメータは、サーチバッファの表示またはサーチバッファへのデータ入力に使用します。長さは暗黙的に設定されます。

等号を指定する場合、「文字列」に指定された値が大文字に変換されます。コロンを指定した場合、大文字の変換は実行されません。

SBL

```
SBL = number
```

このパラメータは、コントロールブロック内のサーチバッファ長を指定するものです。

SET_PWD

```
SET_PWD = string
```

このパラメータは、パスワード資格情報を設定します。

この資格情報は、コマンド処理中に Adabas サーバーによってチェックされます。

使用する場合は、コントロールパラメータ SET_UID も設定する必要があります。どちらかの値が欠落している場合や無効な場合は、Adabas エラーが返されます。

SET_UID

```
SET_UID = string
```

このパラメータは、ユーザー ID 資格情報を設定します。

この資格情報は、コマンド処理中に Adabas サーバーによってチェックされます。

使用する場合は、コントロールパラメータ SET_PWD も設定する必要があります。どちらかの値が欠落している場合や無効な場合は、Adabas エラーが返されます。

TIME

```
TIME
```

この機能は、現在の時刻をマークし、そのマークとそれ以前の最終マーク間の違いを表示します。

[NO]TRACE

```
[NO]TRACE
```

このオプションは、ループ実行のトレースを行います。

デフォルトは NOTRACE です。

VB

```
VB [{=|:} string]
```

このパラメータは、バリューバッファの表示、またはバリューバッファへのデータ入力に使用します。長さは暗黙的に設定されます。"string" が RB と同じである場合、レコードバッファはバリューバッファに移動されます。

等号を指定する場合、「文字列」に指定された値が大文字に変換されます。コロンを指定した場合、大文字の変換は実行されません。

VBL

```
VBL = number
```

このパラメータは、コントロールブロック内のバリューバッファ長を指定するものです。

WAIT

```
WAIT [= seconds]
```

このパラメータを使用すると、指定した時間まで ADATST が待ち状態となります。待機時間は秒単位で入力します。一度時間を設定すると、何も追加せずに「WAIT」と入力するだけで、同じ時間待機します。

デフォルト時間は 10 秒です。

例

```
adatst: wait = 15
```

ADATST は 15 秒待機します。

WRITE_RB

```
WRITE_RB = string
```

このパラメータは、最初のレコードバッファの内容を書き込むファイルの名前を指定します。レコードバッファは、指定した名前のファイルが存在していない場合にのみ書き込まれます。

WRITE_RB2

```
WRITE_RB2 = string
```

このパラメータは、2番目のレコードバッファの内容を書き込むファイルの名前を指定します。レコードバッファは、指定した名前のファイルが存在していない場合にのみ書き込まれます。

WRITE_RB3

```
WRITE_RB3 = string
```

このパラメータは、3番目のレコードバッファの内容を書き込むファイルの名前を指定します。レコードバッファは、指定した名前のファイルが存在していない場合にのみ書き込まれます。

31 ADAULD（ファイルのアンロード）

この章では ADAULD ユーティリティについて説明します。

機能概要

ユーティリティ ADAULD は、Adabas ファイルをアンロードします。すなわち、データベースまたは Adabas バックアップコピーからレコードが読み込まれ、シーケンシャルファイルに書き込まれます。

ファイルをアンロードする主な理由は次のとおりです。

- スペース割り当ての変更、インデックス、アドレスコンバータまたはデータストレージに割り当てられた論理エクステント数の削減、パディングファクタの変更を行うため。この場合には、ファイルのアンロード、削除、再ロードを行う必要があります。これらの機能は ADAORD にも装備されています。
- 同じデータを持つ1つまたは複数のテストファイルの作成を行うため。この手順では、ファイルをアンロードし、それから異なるファイル番号を使用して再ロードする必要があります。この機能は ADAORD にも装備されています。
- 後続の ADAMUP への入力のためにファイルからデータの抽出を行うため。これは、本番データベースからアーカイブデータベースにレコードを移行するときに便利です。
- Adabas バックアップコピーにアーカイブされたファイルの再設定を行うため。

ファイルをデータベースからアンロードするときは、次の順番でレコードをアンロードできます。

論理順
: レコードは、ユーザー指定ディスクリプタの値に基づいて昇順にアンロードされます。

ISN 順
: レコードは、昇順の ISN 順でアンロードされます。

物理順
: レコードは、データストレージ内で物理的に位置する順番でアンロードされます。

論理順または ISN 順でアンロードする場合には、ニュークリアスがアクティブでなければなりません。ADAULD がデータベースコンテナファイルにアクセスするのであれば、物理順でアンロードするときに、ニュークリアスはアクティブでなくてもかまいません。

Adabas バックアップコピーからアンロードする場合は、レコードは ADABCK によって格納された順にアンロードされます。これは、一般には昇順のデータ RABN です。ただし、DRIVES オプションが使用された場合またはダンプがオンラインで行われた場合には、必ずしもこの順番になるとは限りません（詳細については、ユーティリティ ADABCK の DRIVES オプションの説明を参照）。

アンロードされたレコードは、圧縮形式で出力され、圧縮ユーティリティ ADACMP によって生成されるレコードと同一です。各データレコードの先頭にはその ISN が付くので、ファイル

の再ロード時にこれらの ISN をユーザー ISN として使用できます（詳細についてはユーティリティ ADAMUP の USERISN オプションの説明を参照）。

ユーザーは、UNLOADプロセスの間に、ファイルに対するインデックスの再作成に必要なディスクリプタ値が省略されるように指定できます（SHORTオプション）。これを指定するとアンロード処理時間が短くなります。出力を ADAMUP に対する直接の入力とする場合には、このオプションを使用しないでください。

注意: ファイルに照合ディスクリプタが含まれている場合、アンロードされたデータのICUバージョンは変更されません。ADAMUPを使用して、同じフィールドを含んでいてICUバージョンは異なるファイルにデータをロードできますが、これはファイルが空で、ADAMUP に NEW_FDT オプションを使用している場合に限ります。

完了すると、ADAULD は以下のいずれかの終了ステータス値を返します：

0
レコードは正常にアンロードされ、データベースの破損は検出されませんでした。

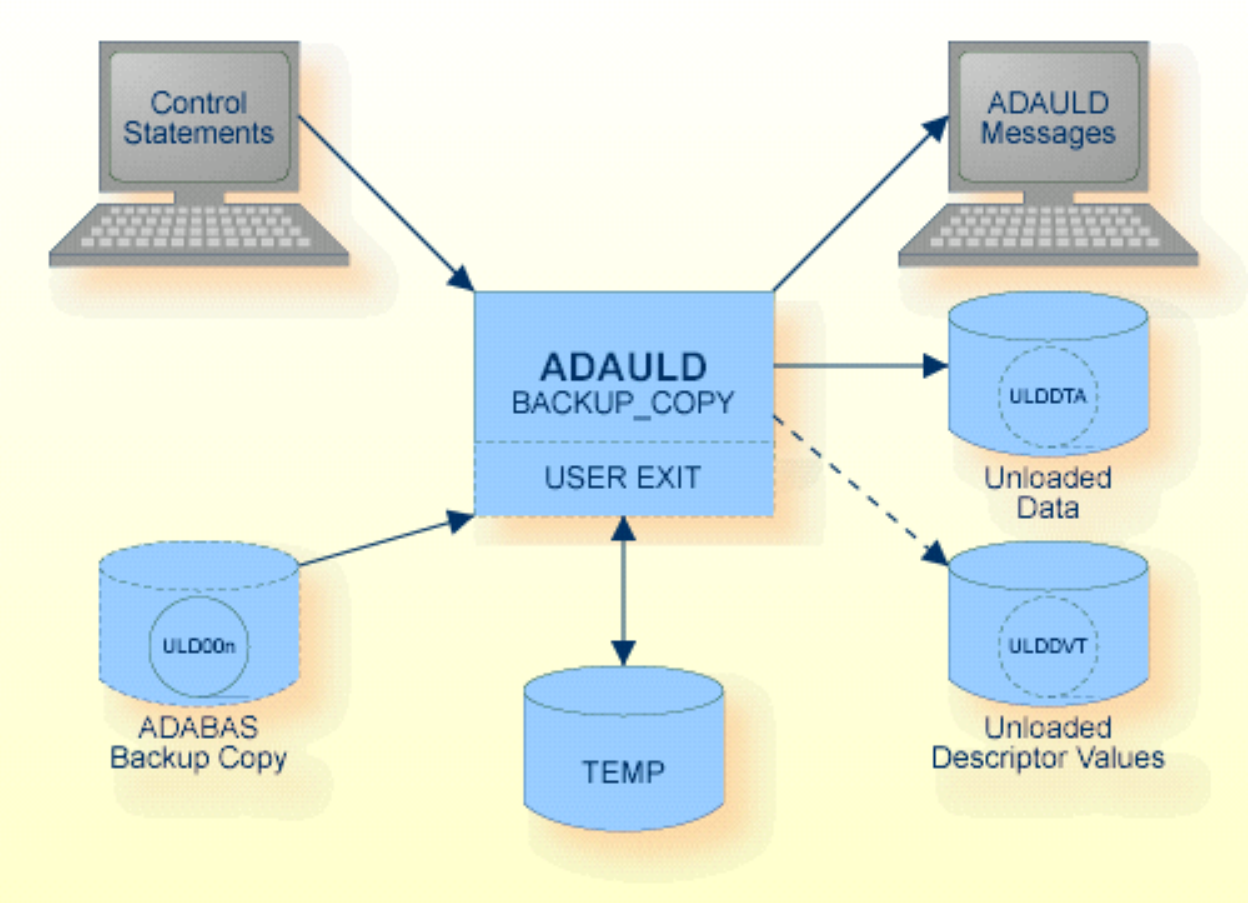
12
アンロードに成功しましたが、破損したデータレコードが検出され、それらはアンロードされませんでした。ADAVFY を実行して、データベースの破損に関する詳細情報を取得することをお勧めします。

15
アンロードは成功しましたが、アンロードされたレコードはありませんでした。少なくとも1つのレコードがアンロードされた後でのみ追加のアクティビティが必要な場合に、スクリプトでこのステータス値を確認できます。

255
アンロードに失敗しました。

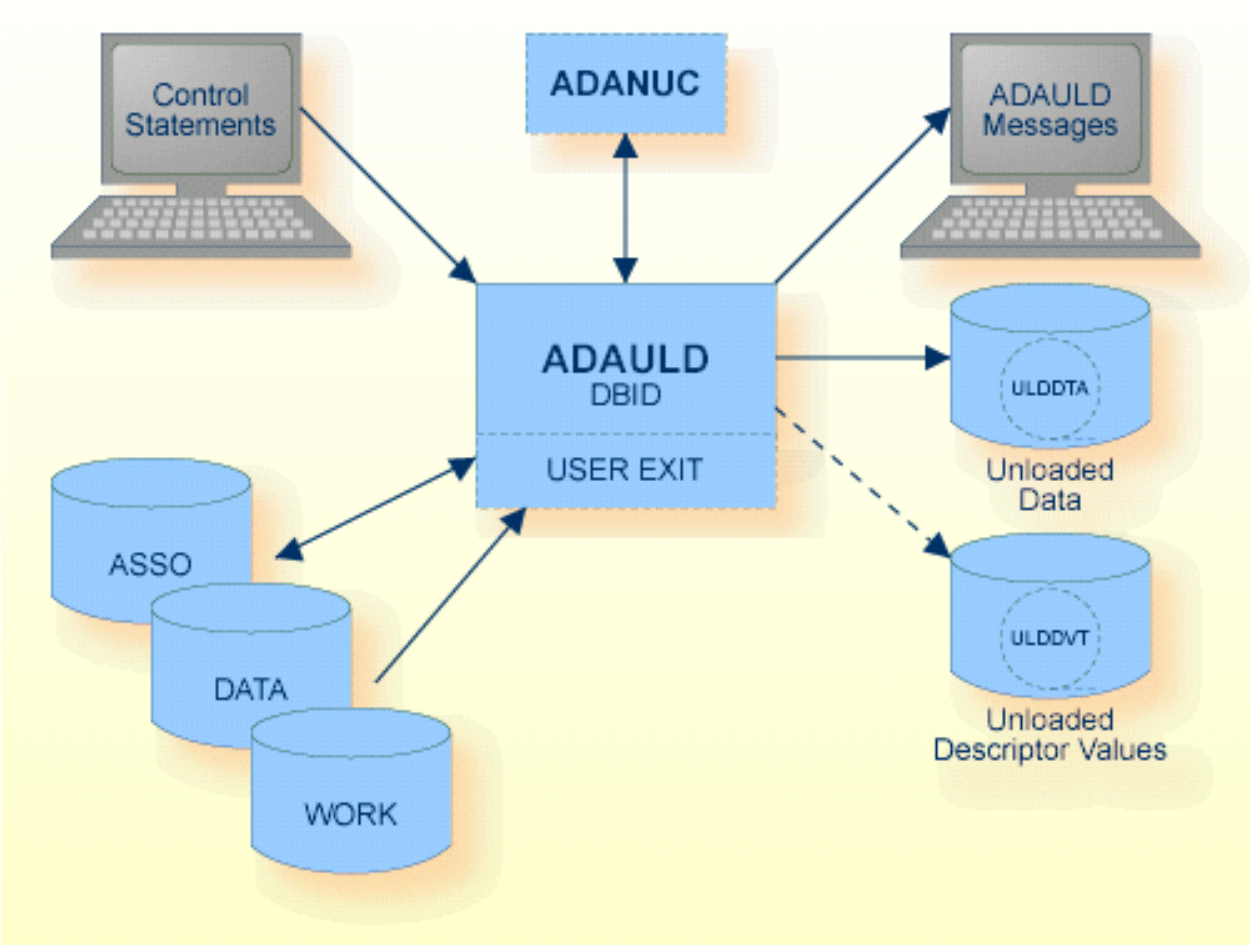
このユーティリティは単一機能ユーティリティです。

処理フロー



BACKUP_COPY Function

シーケンシャルファイルULD00n、ULDDTA、ULDDVTは複数エクステントを持つことができます。複数のエクステントを持つファイルの詳細については、『*Adabas Basics*』の「ユーティリティの*使用*」を参照してください。



DBID Function

シーケンシャルファイル ULDDTA、ULDDVT は複数エクステントを持つことができます。複数のエクステントを持つファイルの詳細については、『Adabas Basics』の「ユーティリティの使用」を参照してください。

データセット	環境変数／論理名	記憶媒体	追加情報
アソシエータ	ASSOx	ディスク、テープ	注 2 参照
データストレージ	DATAx	ディスク、テープ	注 2 参照
バックアップコピー	ULD00n	ディスク、テープ	ADABCK の DUMP 機能の出力、ADAULD に対する入力
アンロードデータ	ULDDTA	ディスク、テープ（注 1 参照）	

データセット	環境 変数／ 論理名	記憶 媒体	追加情報
アンロードディスクリプタ値	ULDDVT	ディスク、テープ（注 1 参照）	
コントロールステートメント	stdin/ SYS$INPUT		ユーティリティマニュアル
ADAULD メッセージ	stdout/ SYS$OUTPUT		メッセージおよびコード
一時ストレージ	TEMPx	ディスク、テープ	注 3 と 4 を参照
ワークストレージ	WORK1	ディスク、テープ	注 2 参照

注意:

1. このシーケンシャルファイルには、名前付きパイプを使用できます（OpenVMS にはない。詳細については、『*Adabas Basics*』の「*ユーティリティの使用*」を参照）。
2. オフラインでのアンロードに必要です。物理順に従ってオンラインでアンロードするときのスピードも高くなります。
3. オンラインオプションを使用したバックアップコピーからアンロードする場合だけ必要です。ユーティリティをオフラインで実行するとき、環境変数 TEMP1 を WORK1 と同じ値に設定すると、自動再スタートが保留状態でなければ、WORK を TEMP として使用することができます。
4. ADAULD BACKUP_COPY 機能は、TEMP を探すために DBxxx.INI ファイルを読み込みません。そのため、TEMP は環境変数で指定する必要があります。

チェックポイント

次の表は、各機能に対するニュークリアス条件と記録チェックポイントを示しています。

機能	ニュークリアスのアクティブ化が必要	ニュークリアスの非アクティブ化が必要	ニュークリアスは不要	記録チェックポイント
BACKUP_COPY			X	-
DBID	X（注 1 参照）	X（注 3 参照）	X（注 2 参照）	SYNX

注意:

1. 論理順および ISN 順でアンロードする場合、または ADAULD がデータベースコンテナファイルにアクセスできない場合（例えばリモートノードからアンロードする場合）。ファイルに LOB データが含まれる場合も適用されます。LOB データも論理順でアンロードする必要があるからです。サーチバッファとバリューバッファを指定した場合も適用されます。

2. 物理順でアンロードし、ADAULD がデータベースコンテナファイルにアクセスする場合。
3. Adabas システムファイルをアンロードする場合。

制御パラメータ

次のコントロールパラメータを使用できます。

```
     BACKUP_COPY = number, FILE = number
                   [,FDT]
                   [,NUMREC = number]
D                  [,[NO]ONLINE]
D                  [, [NO]SHORT | [NO]SINGLE_FILE ]
                   [,SKIPREC = number]
D                  [,[NO]USEREXIT]

     DBID = number , FILE = number
            [,FDT]
D           [,[NO]LITERAL]
            [,NUMREC = number]
            [,SEARCH_BUFFER = string, VALUE_BUFFER = string]
D           [, [NO]SHORT | [NO]SINGLE_FILE  ]
            [,SKIPREC = number]
            [,SORTSEQ = { string | ISN } ]
            [,STARTISN = number]
D           [,[NO]USEREXIT]
```

BACKUP_COPY

```
BACKUP_COPY = number
              ,FILE = number
              [,FDT]
              [,NUMREC = number]
              [,[NO]ONLINE]
              [,[NO]SHORT | [NO]SINGLE_FILE ]
              [,SKIPREC = number]
              [,[NO]USEREXIT]
```

このパラメータは、Adabas バックアップコピーからレコードをアンロードするものです。LOB ファイルを指定することはできません。"BACKUP_COPY=number" は、バックアップコピー元のデータベース ID を指定し、"FILE=number" は、ファイル番号を指定します。オフラインおよびオンライン両方のバックアップコピーを使用できます。指定ファイルに LOB ファイルが割り当てられている場合、ADAMUP パラメータ NUMREC や SKIPREC を使用した部分的再ロードはできません。

FDT

このパラメータは、アンロードするファイルの FDT を表示します。

FILE = number

このパラメータは、アンロードするファイルを指定します。

NUMREC = number

このパラメータは、アンロード時にファイルから取得するデータレコードの数を制限します。NUMREC を省略し、SKIPREC を指定しなかった場合、すべてのレコードがアンロードされます。再ロードするファイルに LOB ファイルが割り当てられている場合、NUMREC を使用することはできません。

[NO]ONLINE

このオプションは、バックアップコピーの中に、ファイルのアンロード対象であるオンラインデータストレージブロックを入れてもよいかどうかを指示します。

バックアップコピーの中にオンラインデータストレージブロックが入っている可能性がある場合は、バックアップコピーの処理時に2つのパスが作成されます。これは、各データストレージブロックの最新バージョンを見つける必要があるからです。このパラメータを NOONLINE に設定した場合には、1つのパスでアンロードされるため、かなりの処理時間の節約となりますが、オンラインデータストレージブロックが検出されると、ADAULDがエラーメッセージを発行して終了する場合があります。

使用されるデフォルトは、バックアップが作成されたときにAdabasニュークリアスがアクディブだったかどうかに依存します。

[NO]SHORT

このオプションは、インデックスの構築に使用するディスクリプタ値を出力に含めるか省略するかを指定します。

SHORT を指定した場合、ディスクリプタ値はアンロードされません。

出力を、一括更新ユーティリティの入力として直接使用する場合、NOSHORT モードでファイルをアンロードする必要があります。

SHORT と SINGLE_FILE は相互に排他的です。

NOSHORT がデフォルトです。

[NO]SINGLE_FILE

このオプションを SINGLE_FILE に設定した場合、ADAULD は DVT および DATA 情報を単一のデータセット（ULDDTA）に書き込みます。

SINGLE_FILE と SHORT は相互に排他的です。

デフォルトは NOSINGLE_FILE です。

SKIPREC = number

このパラメータは、アンロードを開始する前にスキップするレコード数を指定します。再ロードするファイルにLOBファイルが割り当てられている場合、SKIPRECを使用することはできません。

[NO]USEREXIT

ユーザー作成ルーチンはダイナミックにロードされます。入力パラメータブロックへのポインタおよび出力パラメータへのポインタはコールごとに渡されます（詳細については、インクルードファイル adauex.h を参照）。データベースから取得された各レコードに対して、レコードをアンロードするか（アンロードファイルに書き込むか）、スキップするか、直ちに実行を終了するかを選択できます。

環境変数／論理名 ADAUEX_7 はユーザー作成ルーチンをポイントしている必要があります。

詳細については、『*管理*マニュアル』の「*ユーザー出口とハイパー出口*」に関する説明を参照してください。

NOUSEREXIT がデフォルトです。

DBID

```
DBID = number
       ,FILE = number
       [,FDT]
       [,[NO]LITERAL
       [,NUMREC = number]
       [,SEARCH_BUFFER = string]
       [,[NO]SHORT | [NO]SINGLE_FILE ]
       [,SKIPREC = number]
       [,SORTSEQ = { string | ISN }]
       [,STARTISN = number]
       [,[NO]USEREXIT]
       [,VALUE_BUFFER = string]
```

このパラメータは、指定データベースからレコードをアンロードするものです。

FDT

このパラメータは、アンロードするファイルの FDT を表示します。

FILE = number

このパラメータは、アンロードするファイルを指定します。LOB ファイルを指定することはできません。

[NO]LITERAL

このオプションを LITERAL に設定する場合、先頭が空白の文字や小文字をバリューバッファに指定できるようになり、文字列の中でも有効になります。つまり、先頭の空白や小文字が削除されたり、大文字に変換されたりすることはありません。NOLITERAL を設定すると、値を 16 進数値として指定する場合を除いて、小文字が大文字に変換され、先頭の空白は削除されます。

NOLITERAL がデフォルトです。

NUMREC = number

このパラメータは、アンロード時にファイルから取得するデータレコードの数を制限します。NUMREC が省略されて SKIPREC と STARTISN のどちらも指定されないと、ファイルの全レコードがアンロードされます。

SEARCH_BUFFER = string

このパラメータは、指定した選択基準に合うようにアンロードするレコードを制限します。選択条件は、『コマンドリファレンスマニュアル』に記述されているサーチバッファエントリの構文に従って指定する必要があります。

このパラメータの最大長は 200 バイトです。エントリが複雑な場合は、次の方法を使用します。

```
adauld: search_buffer=aa,20,a,d,\
> ab,10,a.
```

ADAULD は次のように連結します。

```
aa,20,a,d,ab,10,a.
```

選択条件に対応する値は VALUE_BUFFER パラメータで指定します。

[NO]SHORT

このオプションは、インデックスの構築に使用するディスクリプタ値を出力に含めるか省略するかを指定します。

SHORT を指定した場合、ディスクリプタ値はアンロードされません。

出力を、一括更新ユーティリティの入力として直接使用する場合、NOSHORT モードでファイルをアンロードする必要があります。

SHORT と SINGLE_FILE は相互に排他的です。

NOSHORT がデフォルトです。

[NO]SINGLE_FILE

このオプションを SINGLE_FILE に設定すると、ADAULD は DVT 情報および DTA 情報を 1 つのデータセット（ULDDTA）に書き込みます。

SINGLE_FILE と SHORT は相互に排他的です。

デフォルトは NOSINGLE_FILE です。

SKIPREC = number

このパラメータは、アンロードを始める前にスキップされるデータレコード数を指定します。

STARTISN パラメータと組み合わせて使用すると、まず開始位置に移動してから、スキップが実行されます。

SORTSEQ = string

このパラメータは、ファイルをアンロードするときの順番を制御します。指定する場合、ディスクリプタ、サブディスクリプタ、またはスーパーディスクリプタのフィールド名、またはキーワード ISN のいずれかを指定できます。デフォルトは物理順です。

1. 論理順

   ディスクリプタまたはサブ／スーパーディスクリプタのフィールド名を文字列として指定すると、フィールド名が示すディスクリプタ値の昇順の論理順でレコードがアンロードされます。フィールド名は、ピリオディックグループ内に含まれるディスクリプタを示すことはありません。

   フィールド名がマルチプルバリューフィールドであるディスクリプタを示す場合、同じレコードが複数回アンロードされることもあります（レコード内の異なるディスクリプタ値ごとに1回）。そのため、アンロードの順番を制御するうえで、このタイプのディスクリプタを使用することは望ましくありません。

フィールド名が NU または NC パラメータ付きで定義されたディスクリプタを示す場合は、そのディスクリプタに対して空値を持つレコードはアンロードされません。

2. ISN 順

   ISN を指定すると、レコードは昇順の ISN 順でアンロードされます。

3. 物理順

   SORTSEQ パラメータを省略すると、データストレージに格納されている物理順に従ってレコードはアンロードされます。

   サーチバッファを指定していて SORTSEQ パラメータを省略した場合には、レコードは ISN の昇順にアンロードされます。

STARTISN = number

SORTSEQ = ISN オプションを使用するか、サーチバッファを指定する場合、ファイル内の最小 ISN からではなく、指定の ISN からアンロードを開始することができます。指定した ISN が存在しない場合、その次に大きい ISN からアンロードが開始されます。

[NO]USEREXIT

ユーザー作成ルーチンはダイナミックにロードされます。入力パラメータブロックへのポインタおよび出力パラメータへのポインタはコールごとに渡されます（詳細については、インクルードファイル adauex.h を参照）。データベースから取得された各レコードに対して、レコードをアンロードするか（アンロードファイルに書き込むか）、スキップするか、直ちに実行を終了するかを選択できます。

環境変数／論理名 ADAUEX_7 はユーザー作成ルーチンをポイントしている必要があります。

詳細については、『*管理マニュアル*』の「*ユーザー出口とハイパー出口*」に関する説明を参照してください。

NOUSEREXIT がデフォルトです。

VALUE_BUFFER = string

SEARCH_BUFFER パラメータで選択条件を指定した場合、その選択条件に対応する値をこのパラメータに指定します。このパラメータの最大長は 2000 バイトです。

注意: 「[NO]LITERAL」も参照してください。これは、バリューバッファの大文字への変換を制御します。

例

例 **1**

```
adauld: backup_copy = 3, file = 6
```

データベース 3 のバックアップコピーのファイル 6 がアンロードされます。[NO]ONLINE オプションのデフォルトの設定によっては、TEMP データセットとバックアップコピーにアクセスするための 2 つのパスが必要なることがあります。

例 **2**

```
adauld: backup_copy = 3, file = 6
adauld: single, noonline
```

同じファイルがアンロードされます。データレコードとディスクリプタバリューテーブルエントリの両方が同じ出力ファイルに書き込まれます。オンラインブロックがあるかどうか不明なため、バックアップコピーは 1 回のパスで処理されます。TEMP データセットは必要ありません。

例 **3**

```
adauld: dbid = 3, file = 6, skiprec = 100
```

データベース 3 のファイル 6 がアンロードされます。レコードは、データストレージに格納されている物理順に従ってアンロードされます。検出された最初の 100 レコードは出力ファイルに書き込まれません。

例 **4**

```
adauld: dbid = 3, file = 6
adauld: numrec = 10
adauld: sortseq = ab
adauld: short
```

データベース 3 のファイル 6 からの 10 レコードがアンロードされます。ディスクリプタ AB の値が、レコードの検索順序の制御に使用されます。再ロード時にインバーテッドリストを再作成するのに必要な値は省略されています。

例 **5**

```
adauld: dbid = 3, file = 6, sortseq = isn, startisn = 123
```

データベース 3 のファイル 6 がアンロードされます。レコードは、ISN 123 で始まる昇順の ISN 順でアンロードされます。

TEMP データセットのスペース見積もり

NOONLINE オプションを設定せずに Adabas バックアップコピーからアンロードする場合、オンラインブロックオカレンスについての情報を累積するために TEMP データセットが必要です。

TEMP ブロックサイズのデフォルトを 4 キロバイトとして概算で算出する場合には、公式 TRH=DRH/1000 を使用できます。

次の公式は、正確な容量の計算に使用できます。

```
X = ENTIRE ((DRH / BSTD) * 4)

TRH = X + ENTIRE (X / BSTD / 8) + 1
```

上記式内の各項目の意味は次のとおりです。

ENTIRE
2 番目に大きい整数値。

BSTD
TEMP ブロックサイズ（バイト単位）。

DRH
バックアップコピー上のデータベース内の最大データストレージ RABN。ADABCK ユーティリティの SUMMARY 機能を使用して、この数を得ることができます。

TRH
TEMP に必要な最大 RABN。

再スタートに関する考慮事項

ADAULD は再スタート機能を備えていません。中断された ADAULD の実行は、最初から再実行しなければなりません。

32 ADAVFY（データベースの整合性チェック）

次のトピックについて説明します。

機能概要

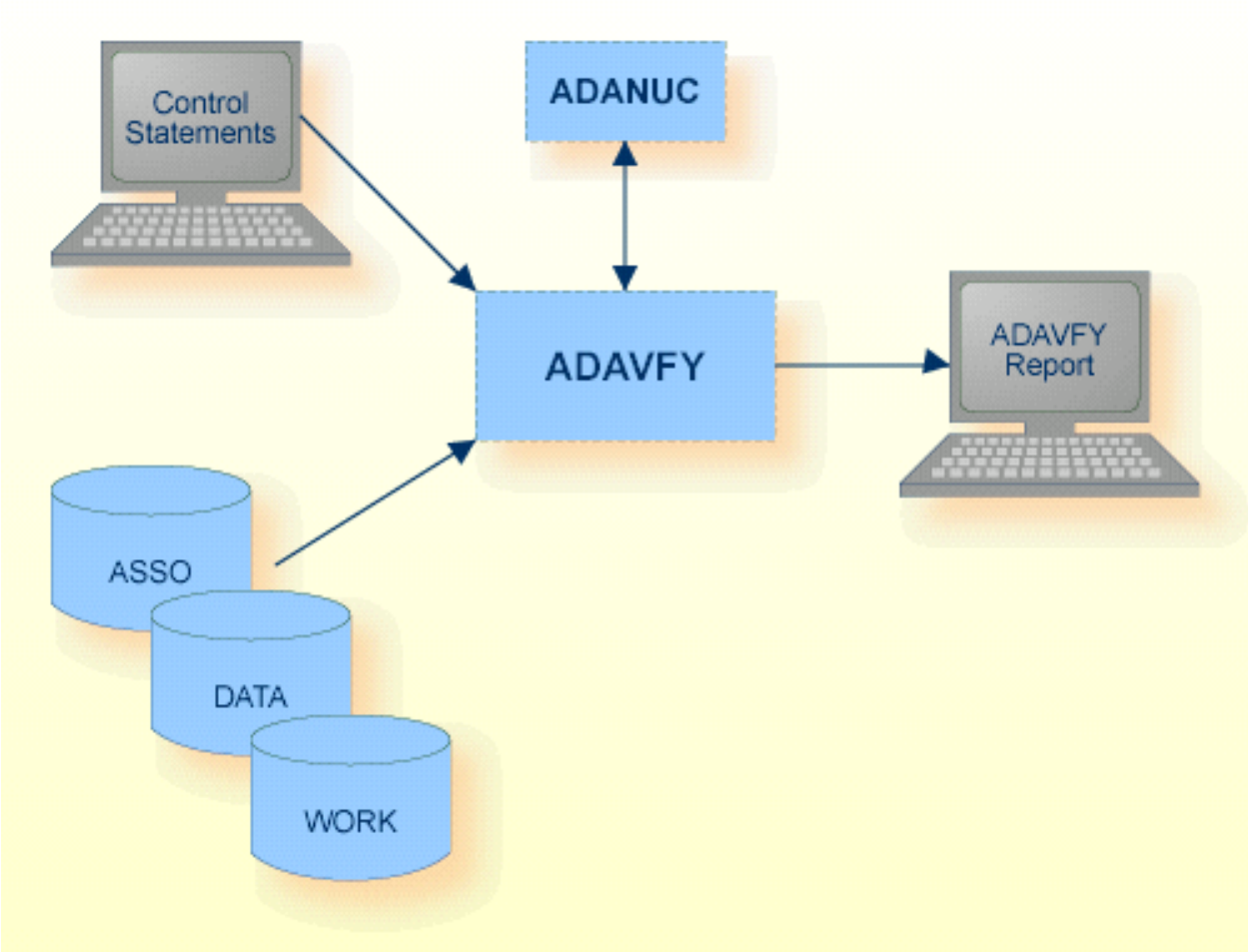
ADAVFY ユーティリティは、データベースの整合性チェックを行います。ロードされたファイルの各ファイルコントロールブロック（FCB）と各フィールド定義テーブル（FDT）とともにジェネラルコントロールブロック（GCB）が検証されます。インデックス構造およびデータストレージも検証されます。また、消失した RABN も検出されます。

アクティブなニュークリアスに対して ADAVFY を実行するか、またはデータベース更新を実行するユーティリティと並行して実行すると、エラーがレポートされます。これは、ユーティリティが終了するまでの間に更新が行われる可能性があり、その更新のうちの一部がニュークリアスバッファプール内にのみ反映されるためです。ADAVFY を使用する場合、Adabas ニュークリアスがアクティブである必要はありません。ADAVFY はデータベースをオフラインで処理します。

一般に、ADAVFYは、検出した整合性エラーを表示するだけで、データベースは修正しません。ただし、ADAVFY は、FCB および FDT の一部のエラーを検出したときに、オフラインモードでこれらを修正します。例えば、ファイル内のレコード数に関する FCB のレコードカウンタの値が無効な場合のエラーなどです。

このユーティリティは多機能ユーティリティです。

処理フロー



データセット	環境変数／論理名	記憶媒体	追加情報
アソシエータ	ASSOx	ディスク、テープ	
データストレージ	DATAx	ディスク、テープ	
コントロールステートメント	stdin/ SYS$INPUT		ユーティリティマニュアル
ADAVFY メッセージ	stdout/ SYS$OUTPUT		メッセージおよびコード
WORK	WORK1	ディスク、テープ	

チェックポイント

このユーティリティはチェックポイントを書き込みません。

制御パラメータ

次のコントロールパラメータを使用できます。

```
     AC


     DATA
M    DBID = number
D    ERRORS = number
     FCB
     FIELD
D    FILES = { * | (number [-number][,number[-number]]...)  }
     FROM = number - number
     INDEX
D    LEVEL = number
     LOST
     RECORD
```

パラメータ AC、DATA、FCB、FIELD、INDEX、LOST、および RECORD は、対応する検証機能を直ちに呼び出します。残りのパラメータは、上記パラメータの前に指定されている場合にのみ評価されます。

AC

```
AC
```

この機能は、FILESパラメータで指定したファイルを対象に、アドレスコンバータからデータストレージまでの間を検証し、指定データストレージの中にレコードがあるかどうかをチェックします（DATA も参照）。

DATA

```
DATA
```

この機能は、指定ファイル番号を対象にデータストレージを検証します。この機能は、FILESパラメータで指定したファイルを対象に、アドレスコンバータからデータストレージの間、およびデータストレージからアドレスコンバータの間を検証します。ADAVFY DATA 機能は、次のエラーがオフラインモードで検出された場合に、エラーを修正します。FCB には、ファイル内のレコード数のレコードカウンタが含まれていますが、このカウンタの値が正しくない場合は、値が修正されます。

DBID

```
DBID = number
```

このパラメータは、検証対象データベースを選択するためのものです。

ERRORS

```
ERRORS = number
```

このパラメータは、1 ファイルの検証が終了するまでに発生したエラーのうち、何件までをエラーとしてレポートするのかを指定します。最小値は 1、デフォルト値は 20 です。

FCB

```
FCB
```

この機能は、FILESパラメータで指定したファイルを対象に、ファイルコントロールブロックとフィールド定義テーブルを検証します（INDEX も参照）。

フィールド

```
FIELD
```

この機能は、データストレージを検証します。指定ファイルのレコード構造をチェックし、アンパック形式、パック形式、および浮動小数点の値の内容を検証します。

FILES

```
FILES = { * | (number[-number][,number[-number]]...)  }
```

このパラメータは、検証対象となるファイルを指定するものです。アスタリスク（*）を指定すると、検証対象が全ファイルになります。LOST 機能を除くすべての機能に FILES パラメータは必要です。

デフォルトは、ファイルの指定なしです。

FROM

```
FROM = number - number
```

指定する値は、各種構造を出力するために LEVEL オプションと組み合わせて使用されます。詳細については、このセクションの LEVEL パラメータを参照してください。

INDEX

```
INDEX
```

この機能は、レベル1（ノーマルインデックス）に対するインデックスの完全性を検証します。これには、FCB および FDT の検証も含まれます。

ADAVFY では、used（使用済み）、free（空き）、reusable（再利用可能）、および lost（損失済み）の NI（ノーマルインデックス、インデックスレベル 1）、MI（メインインデックス、インデックスレベル2）、および UI（アッパーインデックス、インデックスレベル3以上）ブロックの数もカウントされます。

例：

```
%ADAVFY-I-INDSTR, Index verification
%ADAVFY-I-INDCNT, NI: used: 210, free: 1773, reusage: 17, lost: 0
%ADAVFY-I-INDCNT, UI: used: 1, free: 87, reusage: 2, lost: 1
%ADAVFY-I-INDCNT, MI: used: 9, free: 87, reusage: 2, lost: 1
%ADAVFY-I-INDEND, Index verification completed                    ↩
```

注意:

1. used（使用済み）インデックスブロックは、現在使用されているインデックスブロックです。
2. free（空き）インデックスブロックは、まだ使用されていないインデックスブロックです。

3. reusable（再利用可能）インデックスブロックは、すでに使用されていますが、再び空になり、再利用キューに含まれているインデックスブロックです。これらのブロックは、再度使用できます。
4. lost（損失済み）インデックスブロックは、現在使用されておらず、再利用キューにないインデックスブロックです。再利用できません。lost（損失済み）ブロックの値1は正常です。これは ADAINV REINVERT を実行した後に発生する可能性があります。
5. free（空き）、reusable（再利用可能）、lost（損失済み）の MI および UI ブロックは、同じ論理エクステントから取得されるので、ブロックの数は同じです。表示される数は MI と UI を合わせた数です。MIブロック用にスペースを追加すると、UIブロックに使用できるスペースの数が減少することに注意してください。

LEVEL

```
LEVEL = number
```

このパラメータは、ADAVFY が出力する、内部構造に関する情報量を指定するものです。このパラメータを指定しても、実行する検証の内容には影響はありません。このパラメータを使用する場合、該当する機能の前に指定する必要があります。

デフォルト値は、有効な最高インデックスレベルに 1 を加えた値です。

INDEX 機能の指定時

レベル n	レベル n 以上に関する情報を出力
レベル 0	インデックスブロックの詳細な構造を出力

FROM オプションは、インデックス RABN の範囲指定に使用します。指定した RABN のみがダンプされます。

AC/DATA/RECORD/FIELD 機能の指定時

レベル 2	処理中の RABN を出力
レベル 1	レコード構造（RECORD または FIELD を使用）または各 ISN の参照先（DATA または AC を使用）を出力
レベル 0	レコードのフィールドをダンプ

LOST 機能の指定時

レベル 0	データベースの物理構造をダンプ

LOST

```
LOST
```

このオプションを指定すると、ADAVFY は失われた RABN をデータベース内で探索します。失われた RABN が見つかった場合、ADADBM の RECOVER パラメータを使用してそのスペースを回復することができます。

RECORD

```
RECORD
```

この機能はデータストレージを検証し、指定したファイルの各レコードの構造をチェックします（FIELD も参照）。

例

例 **1**

```
adavfy: dbid=3,file=*,data,field,index
```

機能 DATA、FIELD、および INDEX を使用して、データベース 3 のすべてのファイルを検証します。この機能の組み合わせは、検証のレベルが最も高いものです。

例 **2**

```
adavfy: dbid=3, file=7, level=1, field
```

データベース3のファイル7が検証されます。アンパック形式、パック形式、および浮動小数点のフィールドの内容とデータストレージ内のレコード構造が検証されます。ADAVFY は処理済みの RABN のリストと、処理対象のレコードごとに対応する RABN のオフセット、長さおよび ISN を出力します。

A 付録 A - ユーティリティ入力ファイル例

Adabas キットにはサンプルユーティリティ入力データが収録されています。これらを使用して Adabas ユーティリティを試したり、データベースにサンプルデータをロードして Adabas を体験したりできます。

次の Adabas デモファイルが Adabas キットに収録されています。

File Number	Adabas ファイル名	説明
9	EMPLOYEES	C サンプルプログラムに使用されるファイル（コマンドリファレンス参照）
11	EMPLOYEES-NAT	従業員データを含む、Natural によりサンプルファイルとして使用されるファイル
12	VEHICLES	車両データを含む、Natural によりサンプルファイルとして使用されるファイル
13	MISCELLANEOUS	多数のフィールドを含むファイルのための例
14	9 番の LOBFILE	Adabas ファイル EMPLOYEES の LOB ファイル

注意:

1. OpenVMS では、ファイル 9 は LOB ファイル 14 がない古い Employees ファイルのままです。OpenVMS 向けに提供されている C の例は、この古いサンプルファイルに対応する古いバージョンのプログラムです。これは、コマンドリファレンスドキュメントに記載されている、Windows 向けおよび UNIX 向けに提供されている新しいバージョンとは異なります。
2. UNIX 上で Adabas デモデータベース（Adabas デモファイルを含む Adabas データベース）を作成するために、コマンド crdemodb <dbid> を使用できます。Windows では、[Create Demo Database］アイコンがあります。

Adabas キットには、次のユーティリティ入力ファイルが含まれています。UNIX の場合はディレクトリ 「$ADAPROGDIR/demodb」、Windows の場合はインストールディレクトリのサブディレクトリ 「Adabas\demodb」 にあります。

ファイル名	説明
LoadDemo.bsh（UNIX） loadall.bat（Windows）	ADAFDU、ADACMP、および ADAMUP を介して指定のデモファイルをすべてロードするためのスクリプト
loadfile.bat （Windows のみ）	ADACMP と ADAMUP（loadall.bat から呼び出される）を介してデモファイルの 1 つをロードするスクリプト
emp.cmp	EMPLOYEES ファイル の ADACMP パラメータ
emp.cmpin	ADACMP と ADAMUP を介して EMPLOYEES ファイルにロードされる非圧縮デモデータ
emp.fdt	EMPLOYEES ファイルの FDT を含んでいる FDUFDT ファイル
emp.fdu	EMPLOYEES ファイルの ADAFDU パラメータ
emp_nat.cmp	EMPLOYEES_NAT ファイルの ADACMP パラメータ
emp_nat.cmpin	ADACMP と ADAMUP を介して EMPLOYEES_NAT ファイルにロードされる非圧縮デモデータ
emp_nat.fdt	EMPLOYEES_NAT ファイルの FDT を含んでいる FDUFDT ファイル
emp_nat.fdu	EMPLOYEES_NAT ファイルの ADAFDU パラメータ
mis.cmp	MISCELLANEOUS ファイルの ADACMP パラメータ
mis.cmpin	ADACMP と ADAMUP を介して MISCELLANEOUS ファイルにロードされる非圧縮デモデータ
mis.fdt	MISCELLANEOUS ファイルの FDT を含んでいる FDUFDT ファイル
mis.fdu	MISCELLANEOUS ファイルの ADAFDU パラメータ
napp_backup.csh （UNIX のみ）	ネットワークアプライアンスファイラの外部バックアップのためのサンプルスクリプト
napp_conf （UNIX のみ）	ネットワークアプライアンスファイラの外部バックアップのためのサンプルコンフィグレーションファイル
napp_restore.csh （UNIX のみ）	ネットワークアプライアンスファイラの外部リストアのためのサンプルスクリプト
ordexp.demo	すべての Adabas デモファイルを含んでいる ORDEXP ファイル。
veh.cmp	VEHICLES ファイルの ADACMP パラメータ
veh.cmpin	ADACMP と ADAMUP を介して VEHICLES ファイルにロードされる非圧縮デモデータ
veh.fdt	VEHICLES ファイルの FDT を含んでいる FDUFDT ファイル
veh.fdu	VEHICLES ファイルの ADAFDU パラメータ

B 付録 B - prilogc

prilogc は、ニュークリアスパラメータ CLOGLAYOUT を 6 に設定して作成したコマンドログを出力するサンプルプログラムです。

Adabas キットには、ADANUC パラメータ CLOGLAYOUT=6 で作成したコマンドログから印刷可能な出力を作成する、正式なユーティリティは収録されていません。ただし、提供されている C サンプルプログラム `prilogc` を変更して、出力を調整できます。Software AG はこのプログラムを正式にサポートしません。Adabas の今後のバージョンでもこれが提供されることは保証されません。

ソースファイル *prilogc.c* は、UNIX と Windows の両方で、サブディレクトリ「Adabas/examples/server」 にあります。このディレクトリには、実行ファイルをビルドするためのメイクファイル *makefile* も含まれています。使用法はメイクファイル内で説明されています。必要なヘッダーファイルは、UNIX および Windows の両方とも、インストールディレクトリのサブディレクトリ Adabas /inc にあります。

prilogc の実行ファイルは、UNIX の場合は $ADATOOLS に、Windows の場合は %ADATOOLS% にあります。このパスは、Adabas のインストール時に PATH 設定に組み込まれます。

prilogc は、評価するコマンドログに環境変数 PRICLG が設定されていることを前提に作成されています。prilogc に指定できるパラメータを表示するには、次のように入力します。

```
prilogc -h
```

C 付録 C - Adabas チェックポイント

次のチェックポイントは、Adabas ユーティリティによって書き込まれます。

SYNC
ニュークリアスの初期化時、終了時、キャンセル時、ADAOPR 機能の FEOF=PLOG の実行時、ADAOPR EXT_BACKUP=CONTINUE 機能の実行時、または ADABCK NEW_PLOG 機能の実行時に作成されたチェックポイントを示します。

SYNP
特権制御を必要とするユーティリティによって作成されたチェックポイント。このようなユーティリティでは、Adabas ニュークリアスを使用しなくても更新を実行できます。

SYNX
1 つまたは複数のファイルの排他制御（EXF）を必要とするユーティリティによって作成されたチェックポイント。

次の表に、各ユーティリティまたはユーティリティ機能が書き込むチェックポイントを示します。必要に応じて、データベースのオンライン／オフラインステータスに関する情報も示しています。

ユーティリティ	ユーティリティ機能	チェックポイント	データベースのオンライン／オフライン、コメント
ADABCK	DUMP	SYNX	この機能の最後で AUTORESTART が保留状態のときのみ、ニュークリアスが必要です。
	EXU_DUMP	SYNC	この機能の最後で AUTORESTART が保留状態のときのみ、ニュークリアスが必要です。
	OVERLAY	SYNP	データベースファイルまたはシステムファイルのオーバーレイのために、ニュークリアスは非アクティブになっている必要があります。 ニュークリアスは、ファイルのオーバーレイには必要ありません。

ユーティリティ	ユーティリティ機能	チェックポイント	データベースのオンライン／オフライン、コメント
	RESTORE	SYNP	データベースファイルまたはシステムファイルのリストアのために、ニュークリアスは非アクティブになっている必要があります。 ニュークリアスは、ファイルのリストアには必要ありません。
ADACVT		SYNP	ADACVT は、正常に完了した後に SYNP チェックポイントを書き込みます。
ADADBM	ADD_CONTAINER	SYNP	
	ADD_FIELDS	SYNP	オフライン
		SYNX	オンライン
	ALLOCATE	SYNP	
	CHANGE_FILEDS	SYNP	オフライン
		SYNX	オンライン
	DEALLOCATE	SYNP	
	DELCP	SYNP	
	DELETE	SYNP	オフライン
		SYNX	オンライン
	DROP_FIELDS	SYNP	オフライン
		SYNX	オンライン
	DROP_LOBFILE	SYNP	
	EXTEND_CONTAINER	SYNP	
	NEW_DBID	SYNP	
	NEWWORK	SYNP	
	PGM_REFRESH	SYNP	
	RECOVER	SYNP	
	REDUCE_CONTAINER	SYNP	
	REFRESH	SYNP	
	REMOVE_CONTAINER	SYNP	
	REMOVE_REPLICATION	SYNP	オフライン
	RENAME	SYNP	
	RENUMBER	SYNP	
	REPLICATION_FILES	SYNP	オフライン
		SYNX	オンライン
	RESET	SYNX	
	REUSE	SYNP	
	SYFMAX	SYNP	オフライン

ユーティリティ	ユーティリティ機能	チェックポイント	データベースのオンライン／オフライン、コメント
		SYNX	オンライン
ADAFDU	-	SYNP	オフライン
		SYNX	オンライン
ADAINV	INVERT	SYNP	
	REINVERT	SYNP	
	RELEASE	SYNP	
	RESET_UQ	SYNP	
	SET_UQ	SYNP	
	VERIFY	SYNX	
ADAMUP	UPDATE	SYNP	
ADANUC	スタートアップおよび終了時	SYNC	
ADAOPR	FEOF=PLOG	SYNC	オンライン FEOF=PLOG チェックポイントの後、ADANUC は新しい PLOG セッションの開始時に SYNC チェックポイントを書き込みます。
	EXT_BACKUP=PREPARE	SYNX	オンライン EXT_BACKUP 開始時に書き込まれます。
	EXT_BACKUP=CONTINUE	SYNC	オンライン FEOF = PLOG で書き込まれます。
		SYNX	オンライン EXT_BACKUP 終了時に書き込まれます。
ADAORD	EXPORT	SYNX	
	IMPORT	SYNP	
	IMPORT_RENUMBER	SYNP	
	REORDER	SYNP	
ADAREC	REGENERATE	SYNX	
ADAULD	DBID	SYNX	