Adabas for Linux, UNIX and Windows

Adabas 拡張オペレーション

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。

Copyright © 1987-2017 Software AG, Darmstadt, Germany and/or Software AG USA, Inc., Reston, VA, United States of America, and/or their licensors.

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

Software AG およびその子会社が所有する登録商標および特許の詳細については、 http://documentation.softwareag.com/legal/ を確認してください。

本ソフトウェアの一部にはサードパーティ製製品が含まれています。サードパーティの著作権表示およびライセンス規約については『License Texts, Copyright Notices and Disclaimers of Third-Party Products』を参照してください。このドキュメントは製品ドキュメントセットの一部であり、http://documentation.softwareag.com/legal/上、またはライセンス製品のルートインストールディレクトリ内にあります。

本ソフトウェアの利用は、Software AG のライセンス規約に則って行われるものとします。ライセンス規約は製品ドキュメントセット内、http://documentation.softwareag.com/legal/上、またはライセンス製品のルートインストールディレクトリ内にあります。

ドキュメント IDは: ADAOS-EXOP-66-20200619JA

目次

前書き v
1 1
表記規則 2
オンライン情報 2
データ保護 3
2 はじめに 5
Adabas 拡張オペレーションの動作方法 6
3 コンフィグレーションファイル 11
コンフィグレーションファイルのフォーマット 12
ADABAS.INI 13
DBnnn.INI 21
4 アクションテンプレート 35
5 管理コマンド 37
コンフィグレーションファイルに格納されている情報の取得と変更：adaini 38
コンフィグレーションファイルのインストール：adainst 42
データベースの停止：adakill 44
ログファイルの表示：adalog 44
ログファイルにメッセージを書き出す：adamsg 44
デフォルトデータベースの定義：adaset 45
データベース（複数）の表示：adashow 46
データベースの開始：adastart 47
データベースの停止：adastop 48
6 Adabas ユーティリティの認可の設定 49
コンフィグレーションファイルとロギングファイルの場所 50
adaauth.ini 52
adaaudit.ini 54
adarbac.ini 56
サンプル：adarbac.ini 62

前書き

このドキュメントでは、Adabas 拡張オペレーション（AEO）について説明します。これは、Adabas の日々の管理を大きく簡略化するオペレーションモードです。

Adabas 拡張オペレーションは、熟練した DBA を対象としています。Adabas 拡張オペレーションを使用すると、通常 DBA の介入が必要とされる多くの状況を自動的に検出して解決できます。

このドキュメントの構成は次のとおりです。

- *はじめに*
- *コンフィグレーションファイル*
- *アクションテンプレート*
- *管理コマンド*
- ***Adabas** ユーティリティの認可の設定*

1

- 表記規則 2
- オンライン情報 2
- データ保護 3

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。

2 はじめに

■ Adabas 拡張オペレーションの動作方法 6

AEO を使用する代表的な例は、データベースにほとんど空きがなく、追加のディスクスペースを割り当てる必要がある場合です。AEOを使用しない場合、DBAは使用可能なディスクスペース量を定期的にモニタし、ある危険な限界に達するときは、データベースにディスクスペースの割り当てを追加する準備をしなければなりません。AEOを使用する場合、DBAはそのモニタリングおよび追加ディスクスペースの割り当てを自動的に行うことができます。DBA はある制限に達したときにデフォルトで行う処理（例えば、データベースの使用率が 90 %になったらディスクスペースを割り当てるなど）を指定できます。そして、指定された制限に達すると、AEO は、必要なディスクスペースを自動的に割り当てます。

これは、AEO に割り当てることができる、日常のさまざまなデータベース管理タスクの 1 つの例です。このような管理タスクの多くから DBA を解放するとともに、人為的ミスや過ちを最小限に減らし、要求されたときに必要な管理作業を正確に行えるようになるので、データベース環境の信頼性が高くなります。

AEO は次の領域で使用できます。

- トラブルシューティング
- レスポンスコードのロギング
- 自動データベースバックアップの作成
- パフォーマンス解析
- データベースコンフィグレーションの管理

Adabas 拡張オペレーションの動作方法

標準 Adabas 環境は次に挙げるコンポーネントで構成されています。

- Adabas ニュークリアス
- Adabas ユーティリティ
- データベースコンテナファイル
- Adabas という名前のニュークリアス起動ユーティリティ（PC プラットフォームのみ）
- データベースログファイル adanuc.log
- シーケンシャルデータベースファイル（例：NUCPLG、NUCCLG）

AEO は次に挙げるコンポーネントを追加して、この環境を拡張します。

- イベントアナライザ
- ログフィルタ
- アクションフィルタ
- 各 Adabas データベースのコンフィグレーションファイル

- すべての Adabas データベースに共通するコンフィグレーションファイル
- 一連の管理コマンド（例：データベースの開始および終了）

AEO 環境では、データベースログファイルは以前ほど重要ではなくなりました。これは、データベースログファイルの内容が共通ログファイル ADABAS.LOG にも書き込まれるようになったためです（下記参照）。

イベントアナライザ

イベントアナライザ（AEOアナライザまたは単にアナライザとも呼ばれる）はAEOのコンポーネントで、Adabas ニュークリアスや Adabas ユーティリティが生成したイベントメッセージを受け取り、イベントの性質に応じて*ログフィルタ*や*アクションフィルタ*に渡します。

ファイルやレコードの作成などのデータベースイベントは、イベントアナライザに報告されます。イベントがすぐに対処する必要のない単なるステータスメッセージの場合、ログフィルタに渡されます。

データベース管理上、すぐに何らかの対処が必要なデータベースに更新を行った場合、イベントがアクションフィルタに渡されます。

ログフィルタ

ログフィルタはイベントアナライザからイベントを受け取り、ログファイル ADABAS.LOG に書き出します。このログファイルでこれらのイベントを解析できます。

Adabas コンフィグレーションファイル（ADABAS.INI）

AEO を Adabas 環境で稼動させるには、単一コンフィグレーションファイル ADABAS.INI が存在している必要があります。このファイルには、データベースの開始時に AEO で使用されるグローバル情報が含まれています。この情報は、データベースセッション時に、AEOオペレーションのパラメータを設定するために使用されます。ある特定のデータベースに対して DBnnn.INI ファイル（下記参照）がある場合、それに含まれる情報は ADABAS.INI に指定された情報を上書きします。ADABAS.INI の管理およびその内容の詳細については、「コンフィグレーションファイル」の「***ADABAS.INI***」を参照してください。

データベースコンフィグレーションファイル（DBnnn.INI）

Adabas 環境のデータベース DBnnn（nnn はデータベース番号）で AEO を稼動させるには、対応するコンフィグレーションファイル DBnnn.INI が存在している必要があります。このファイルには、データベースセッション時に AEO オペレーションのパラメータを設定するために、データベースが開始するときに AEO が使うグローバル情報が含まれています。このファイルに指定されている情報は Adabas コンフィグレーションファイル ADABAS.INI に指定された情報を上書きします。DBnnn.INI ファイルの管理およびその内容の詳細については、「コンフィグレーションファイル」の「***DBnnn.INI***」を参照してください。

アクションおよびアクションフィルタ

コンフィグレーションファイル ADABAS.INI および DBnnn.INI 内のトピックおよびサブトピックの多くは、関連するアクションを開始するために使用されます。このようなアクションはテンプレートの形で提供されます。通常、該当するイベントが起こるとアクションが開始します。そのようなアクションの例としては、データベースに必要な再編成を行うために Adabas ユーティリティを自動的に呼び出すアクションなどがあります。また、他のアクションとしては、DBA が介入する必要のある状況を示すために、DBA の端末に警告メッセージを出力することがあります。イベントは Adabas ニュークリアスでも Adabas ユーティリティでも初期化できます。通常、これらのアクションは AEO の制御下で開始されます。この規則の例外として、2 つのアクション ARCHIVE_LOGFILE と SAVE_DB があります。

アクションフィルタはイベントアナライザからイベントを受け取り、イベントの性質に応じて、事前に定義されたアクションを実行します。

管理コマンド

AEO はデータベース管理を簡単にする管理コマンドを提供します。次のコマンドを利用できます。

adaini：コンフィグレーションファイルに格納されている情報を取得および変更する

adainst：ファイル ADABAS.INI を作成および更新する

adakill：データベースを直ちに停止する

adalog：Adabas ログファイルを表示する

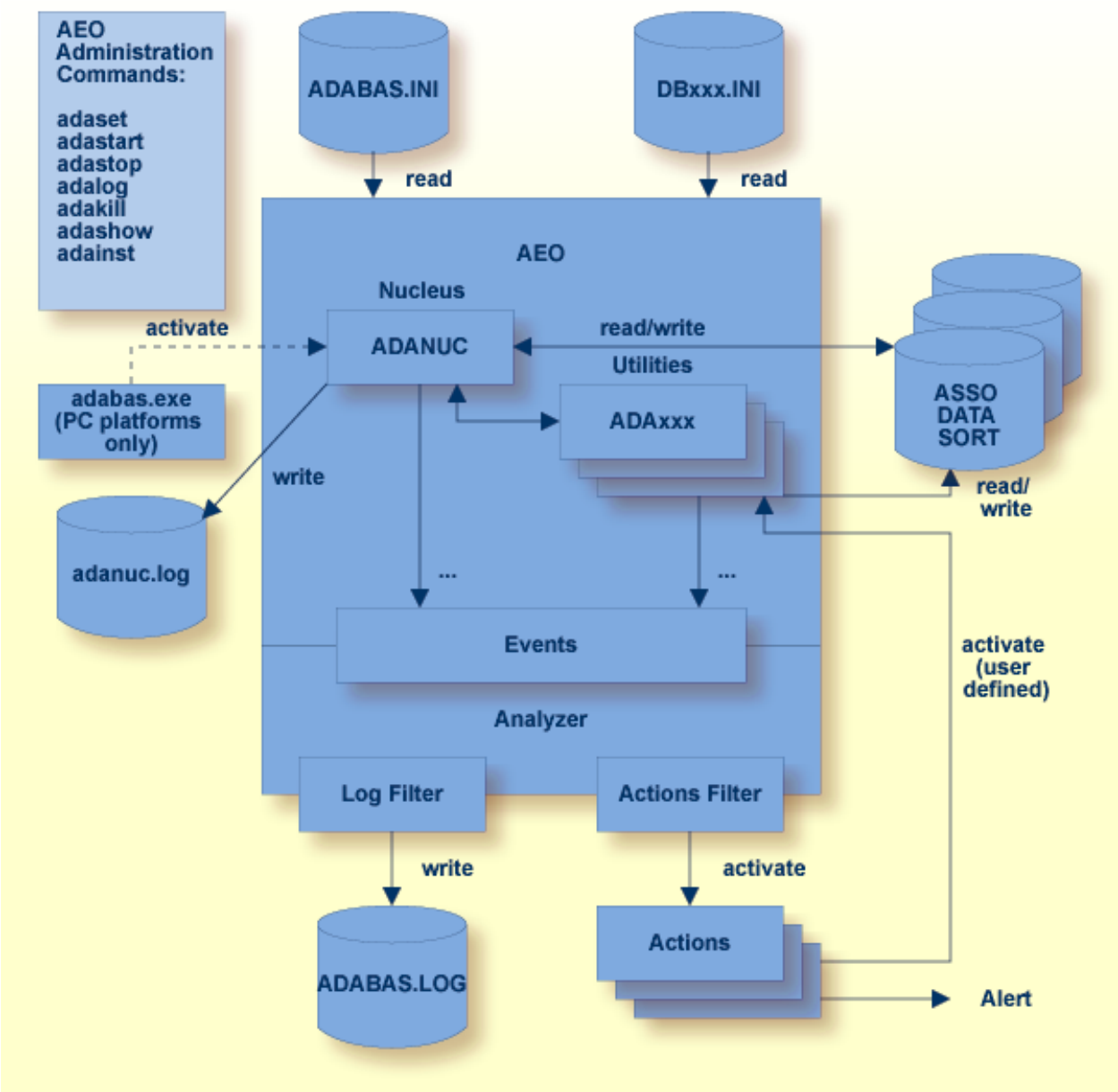
adaset：他の管理スクリプトが必要とするデフォルトのデータベース番号をセットする

adashow：データベースに関する一般情報を表示する

adastart：データベースを開始する

adastop：データベースを正しい手順で終了する

重要: AEO を有効にするには、トピック NODE_PARAMETER 内の項目 ACTION 、サブトピック ANALYSER を "yes" に設定する必要があります。詳細については、「*管理コマンド*」の管理コマンド **adaini** を参照してください。



Adabas Extended Operation environment

3 コンフィグレーションファイル

- コンフィグレーションファイルのフォーマット 12
- ADABAS.INI 13
- DBnnn.INI 21

Adabas 拡張オペレーション（AEO）環境にある各データベースには、コンフィグレーションパラメータが必要です。DBA がデータベースに定義した特殊な規準は、コンフィグレーションパラメータを通じて AEO に通知されます。これにより、AEO は特定のデータベースでイベントが発生したときの対応を認識できます。

コンフィグレーションパラメータはコンフィグレーションファイルに格納されています。コンフィグレーションファイルの 1 つ、ADABAS.INI は、そのノード上の Adabas 環境にあるすべてのデータベースに適用するコンフィグレーションパラメータを含んでいます。さらに各データベースにおいて、DBnnn.INI（nnn はデータベース番号）があります。

ADABAS.INI および DBnnn.INI ファイルには、コンテナ割り当て、ニュークリアスパラメータ、AEO の制御パラメータなどの定義が含まれています。両方とも ASCII ファイルなので、標準のテキストエディタで編集できます。ただし、テキストエディタでは、コンフィグレーションファイルの構文が正しいことは保証されません。コンフィグレーションファイル内の AEO 固有エントリの維持管理には、管理コマンド adaini の使用をお勧めします。

コンフィグレーションファイルのフォーマット

各コンフィグレーションファイルは、次の構文でトピックに分けられています。

```
[topic name]                                        # start of topic "topic name"
  item1 name = item1 value                          # name and value of item 1
  item2 name = item2 value                          # name and value of item 2
  ....
[topic name-END]                                    # end of topic
```

各トピックには項目定義やサブトピックが含まれています。いくつかのレベルを入れ子にしてもかまいませんが、通常 2 レベルまで、つまりメイントピックレベルとサブトピックレベルがあります。

トピック名、項目名、値はすべて大文字と小文字を区別します。コメント行は "#" で始まります。

空白行は無視されます。トピック名、項目名および値の前後の空白部分、つまりブランクとタブも無視されます。値に空白が含まれている場合は引用符（単一引用符（'）または二重引用符（"））デリミタで囲まれていなければなりません。値にデリミタが含まれている場合は 2 回入力しなければなりません。

注意: 環境変数はコンフィグレーションファイルの値に使用できません。明示的な値で置き換える必要があります。

以下に、2 レベルのネストと 1 つのコメント行が含まれるコンフィグレーションファイルの例を示します。

```
[topic1 name]                                   # start of topic on nesting level 1
item1 name = item1 value                          # name and value of item 1

# Start of topic2                                 # comment line
  [topic2 name]                                 # start of topic on nesting level 2
    item2 name = item2 value                      # name and value of item 2
  [topic2 name-END]                               # end of topic 2

[topic1 name-END]                                 # end of topic 1
```

ADABAS.INI

コンフィグレーションファイル ADABAS.INI は、AEO 環境のすべてのデータベースに適用する情報を含みます。ある特定のデータベースに DBnnn.INI ファイルがある場合、そのファイル内にある情報は ADABAS.INI ファイル内の情報を上書きします。ADABAS.INI は、標準のテキストエディタまたは管理コマンド adaini を使用して変更できます。ユーティリティの構文形式については、adaini のオンラインヘルプを参照してください。

ADABAS.INI には次の情報が含まれます。

1. データベース独自の情報。例えば、新しく作成するデータベースに対するデフォルトのニュークリアスパラメータおよび環境情報。ADABAS.LOG（ローカルノード上のすべてのデータベースに対するログファイル）の場所、レポートユーティリティ（adarep、adafin、adapri、adaclp、adaplp、adaerr、adatst）または表示機能（adaopr/adadbm display=...）を実行したときの処理内容などの全般的な情報。
2. データベースの基本的な定義。例えば、個々の DBnnn.INI ファイルの場所やデータベース ID。

ADABAS.INI の場所

UNIX および Windows プラットフォームでは、環境変数 *ADADATADIR* によってポイントされるディレクトリのサブディレクトリ *etc* 内に ADABAS.INI が存在する必要があります。このディレクトリに ADABAS.INI ファイルがない場合、AEO は自動的に使えなくなり、標準の Adabas 機能だけが使えます。

ADABAS.INI の構造

ADABAS.INI は複数のセクションから構成され、セクションごとにトピックが 1 つあります。ADABAS.INI の各セクションはカッコ（[ ]）で囲まれたトピック名を含む行、*[topic-name]* から始まります。AEO に関連するトピックを次に挙げます。

- DB_DEFAULTS（いろいろなサブトピックの指定あり）
- DB_LIST（DBID ごとに 1 つのサブトピックの指定あり）
- MISCELLANEOUS
- NODE_PARAMETER（サブトピックの指定あり）
  - ALERT
  - ANALYSER
  - ARCHIVE_LOGFILE
  - LOGGING

重要: AEO を有効にするには、トピック NODE_PARAMETER 内の項目 ACTION 、サブトピック ANALYSER を "yes" に設定する必要があります。詳細については、「*管理コマンド*」の管理コマンド **adaini** を参照してください。

重要: Audit-Trail 機能を有効にするには、トピック NODE_PARAMETER、サブトピック ANALYSER および LOGGING 内の項目 ACTION を「yes」に設定する必要があります。

上記のトピックを次に説明します。

トピック：DB_DEFAULTS

トピック DB_DEFAULTS には、ADAFRM または adainst で DBnnn.INI を作成する際に使用されるデフォルトの定義が含まれています。DB_DEFAULTS の内容として、[DB_DEFAULTS] と [DB_DEFAULTS-END] 以外の行が DBnnn.INI にコピーされます。詳細については、このセクションの後半の「DBnnn.INI」セクションを参照してください。

```
# configuration                                    # recommended values

[DB_DEFAULTS]
 ...
 [NUCPARMS]
   ...
 [NUCPARMS-END]
   ...
 [DB_PARAMETER]
   ...
 [DB_PARAMETER-END]
   ...
 [ENVIRONMENT]
   ...
```

```
 [ENVIRONMENT-END]
 ...
[DB_DEFAULTS-END]
```

トピック：DB_LIST

トピック DB_LIST は、使用可能なデータベースの番号と名前、および関連する DBnnn.INI ファイルの名前とその場所の一覧です。各データベースの詳細については、トピック DB_LIST 内の別のサブトピック DBID_<*dbid*> に設定されています。

```
# configuration                                    # recommended values

[DB_LIST]

  [DBID_<dbid>]                                     PC platforms:
    INI_FILE = <configuration file name            %ADADATADIR%\db<dbid>\DB<dbid>.INI
                for this database>                   UNIX:
                                                     $ADADATADIR/db<dbid>/DB<dbid>.INI

    NAME = <database name>
  [DBID_<dbid>-END]

[DB_LIST-END]
```

注意: 環境変数は、コンフィグレーションファイル内のパラメータとして使用できません。INI_FILE 項目内では、ADADATADIR 環境変数をフルパス名で置き換える必要があります。

トピック：MISCELLANEOUS

トピック MISCELLANEOUS には、このノードに関するデータベース非依存の情報が含まれます。項目 NODE_NAME は、トピック NODE_PARAMETERS のサブトピック LOGGING（このセクションの後半参照）で有効化されるロギングに使います。

UNIX プラットフォームの場合は、UNIX コマンドの "uname -n" を使って、システムノード名を取得できます。

トピック MISCELLANEOUS の構文は次のとおりです。

```
# configuration                                    # recommended values

[MISCELLANEOUS]
  NODE_NAME = <node name>
[MISCELLANEOUS-END]
```

トピック：NODE_PARAMETER

トピック NODE_PARAMETER には、このノード上のすべてのデータベースに対する AEO 定義が含まれています。

サブトピック：ALERT

```
# configuration                                  # recommended values

[NODE_PARAMETER]
  [ALERT]
    ACTION = <yes/no>
    ACTION_ROUTINE = <action routine>            ada_alrt
    MINIMUM = <minimum severity required         E
                    to call the action>
  [ALERT-END]

[NODE_PARAMETER-END]
```

推奨のアクションルーチンは、ada_alrt です。このルーチンには次のパラメータがあります。

パラメータ	説明
1	ユーティリティ名
2	プロセス ID
3	DBID
4	ユーザーログイン名
5	ノード名
6	重要度
7	メッセージヘッダーキー
8	メッセージヘッダー
9	メッセージテキスト

ALERT アクションは、Adabas イベントが発生したタイミングを DBA（他のユーザーの場合もあり）に通知します。例えば、Adabas はすべてのメッセージに重要度 E（エラー）と F（致命的）を付けて DBA へ自動的にメールすることができます。

トピック NODE_PARAMETER のサブトピック ALERT は Adabas 警告アクションを定義します。ALERT アクションが有効（ACTION=yes）であると、各 Adabas メッセージの重要度をチェックし、定義済みの MINIMUM の重要度と同じか大きければ、ACTION_ROUTINE で指定したアクションが非同期に開始されます。重要度の優先度は、I（情報）< W（警告）< E（エラー）<F（致命的）です。よって、MINIMUM=E のときには、重要度 E と F の付いたメッセージだけが ALERT アクションをコールします。

警告を受け取る人のリストは環境変数 ADA_ALRT_LIST の内容によって決まり、それは空白で区切られた1つ以上のユーザーの指定を連結させたものです。ユーザーの指定はプラットフォー

ムに依存します。ユーザーの指定は Windows ではコンピュータ名となり、例えばマシンが同じドメイン内にある場合は pcABC1、同じドメイン内にない場合は \DOMAIN:pcABC1 のようになります。UNIX ではユーザー名または E-mail アドレスになります。

例

```
set ADA_ALRT_LIST ="pcABC1 /DOM-HQ:pcABC2"    ( Windows の場合 )

setenv ADA_ALRT_LIST "abc user.xyz@myCompany.com"    ( C シェルの場合 )
```

サブトピック：ANALYSER

トピック NODE_PARAMETER のサブトピック ANALYSER は AEO アナライザの有効／無効を切り替えます。アナライザが無効の場合、LOGGING と ALERT を含む、すべての Adabas アクションは自動的に無効になります。

注意: これは、AEO の有効／無効を切り替えるグローバルスイッチです。

```
# configuration                                    # recommended values

[NODE_PARAMETER]

  [ANALYSER]
    ACTION = <yes/no>
  [ANALYSER-END]

[NODE_PARAMETER-END]
```

注意: "ACTION=no" が設定されている場合は、その他のすべての AEO 設定が無視されて、AEO が無効になります。

サブトピック：ARCHIVE_LOGFILE

```
# configuration                                    # recommended values

[NODE_PARAMETER]

  [ARCHIVE_LOGFILE]
    ACTION = <yes/no>
    ACTION_ROUTINE = <action routine>              ada_svlg
    MAXIMUM = <maximum number of                   7
              archive generations>
  [ARCHIVE_LOGFILE-END]

[NODE_PARAMETER-END]
```

推奨のアクションルーチンは、ada_svlg です。このルーチンにはパラメータはありません。

各ロギングメッセージは、Adabas ログファイルの最後に追加されます。このファイルがむやみに大きくならないように、項目 ACTION_ROUTINE で指定したコマンドを外部スケジューラと組み合わせて毎日サブミットすることができます。このルーチンは項目 ACTION を読み、ACTION=no であれば何もしないで終了します。ACTION=yes かつ MAXIMUM=0 のときは、ログファイルが記録されずに削除されます。ACTION=yes で MAXIMUM が 0 より大きく 100 よりも小さいときは、ログファイルの MAXIMUM 生成数（ada_svlg の実行ごとに 1 つ生成される）が記録されます。このとき、PC プラットフォームの場合は 1 からMAXIMUM までの世代番号、UNIX の場合は 01 からMAXIMUM までの世代番号がログファイルに付加されます。例えば、PC プラットフォームの場合には ADABAS.LOG.1、UNIX の場合には、ADABAS.LOG.01 になります。MAXIMUM と同じ世代番号のファイルは、次に ada_svlg が実行されたときに削除されます。

注意: ada_svlg は、ログファイルが他のツール（adalog -t など）によって使用されている間は実行できません。

サブトピック：LOGGING

```
# configuration                                        # recommended values

[NODE_PARAMETER]

  [LOGGING]
    ACTION = <yes/no>
    FILTER_MESSAGES_WITHOUT_HEADER = <yes/no>
    FILTER_REPORT_UTILITIES = <yes/no>                 PC platforms:
    LOG_FILE = <log file name>                         %ADADATADIR%\etc\ADABAS.LOG
                                                       UNIX:
                                                       $ADADATADIR/etc/ADABAS.LOG

  [LOGGING-END]

[NODE_PARAMETER-END]
```

注意: 環境変数は、コンフィグレーションファイル内のパラメータとして使用できません。したがって、INI_FILE 項目に明示的にパス名を指定して、ADADATADIR 環境変数を置き換える必要があります。

トピック NODE_PARAMETER のサブトピック LOGGING は、Adabas メッセージのログに関するパラメータを定義します。Adabas メッセージは、通常、標準出力にのみ書き込まれますが、関連付けられたロギングフィルタの条件に当てはまれば、分析されて AEO ログファイルに追加されます。さらに、AEO には、特殊なイベントやアクションの情報を含む新しい Adabas メッセージが記録されます（標準出力には書き込まれません）。

ログファイルの名前は、項目 LOG_FILE に指定されています。

項目 ACTION はこのノード上のすべてのデータベースに対する AEO ロギングを有効／無効にします。ロギングが有効の場合、各 Adabas ユーティリティはそのロギング情報を項目 LOG_FILE

で定義したシーケンシャルファイルに追加します。ロギングメッセージは書き込みの同期には、Adabas セマフォが使用されます。ロギングフィルタは次の 2 つの条件を認識します。

次の条件の場合、ヘッダーのないメッセージは無視されます。
FILTER_MESSAGES_WITHOUT_HEADER = yes

次の条件の場合、Adabas レポートユーティリティは無視されます（記録されません）。
FILTER_REPORT_UTILITIES = yes

Adabas レポートユーティリティは、adaclp、adafin、adaplp、adapri、adarep、adatst です。

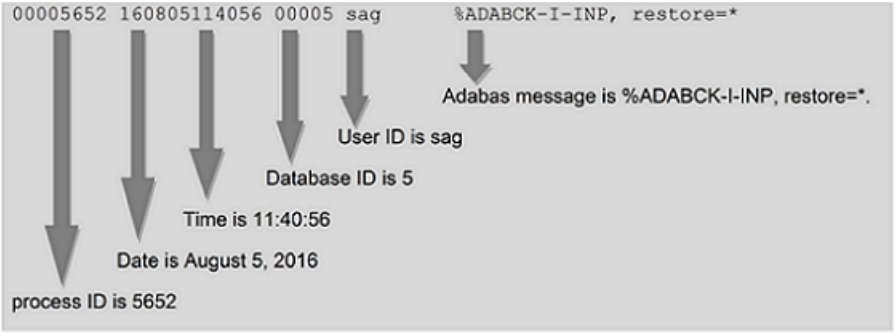
ログファイル内の各エントリのフォーマットは次のとおりです。

```
<process id> <yymmddhhmmss> <dbid> <user ID> <Adabas message>
```

yymmdd と hhmmss は、現在の日付と時間です。

ログファイルエントリ内の項目は空白で区切ります。

例



注意: Adabas ログメッセージにデータベース ID として 0 が含まれている場合は、メッセージが作成されたときにデータベースIDがまだ決定されていなかったことを示します。

Audit-Trail エントリ

Audit-Trail エントリは次のいずれかから開始されます。

- %ADANUC-I-SECURITY（情報）、
- %ADANUC-W-SECURITY（警告）または
- %ADANUC-W-SECURITY（エラー）。

次の情報が表示されます。

- `SECURITY_USER`: 認証されるユーザー。
- `ET_USER`: Adabas コントロールブロックに指定した ET ユーザー。
- `UQID=(terminal ID, process ID, node ID, timestamp)`: Adabas ユーザーキューに指定されているクライアント情報。
- `AUTHN`: ユーザーが正常に認証された場合は "success"、それ以外の場合は "rejected"。

例

```
%ADANUC-I-SECURITY, SECURITY_USER=myuser ET_USER=*test UQID=(myname, myprocess, ↩
mypc, timestamp) AUTHN=success
```

DBnnn.INI

AEO が管理するデータベースそれぞれに、DBnnn.INI ファイル（nnn はデータベース番号）が存在しなければなりません。DBnnn.INI ファイルにはデータベース番号 nnn に対応する情報が含まれていて、ADABAS.INI ファイルにある情報よりも優先されます。

あるデータベースに対して DBnnn.INI ファイルが存在しない場合、AEO はそのデータベースを処理しないため、標準 Adabas 機能のみ使用可能です。また、データベース nnn に対して ADABAS.INI の DB-LIST セクション内の INI_FILE エントリも必ず存在しなければなりません。

DBnnn.INI ファイルには次の情報が含まれます。

1. データベース情報。例えば、コンテナ割り当て、ニュークリアスパラメータなど。
2. アクションの実行権限を規定するユーザー定義。

DBnnn.INI の場所

DBnnn.INI ファイルの場所は、ADABAS.INI の DB_LIST トピックの項目 INI_FILE で定義します。

ユーティリティ ADAFRM を使用して新しいデータベースを作成すると、DBnnn.INI ファイルが生成されます。ユーティリティ adaini を使用して、DBnnn.INI の内容を変更できます。

DBnnn.INI の構造

DBnnn.INI は複数のセクションから構成され、セクションごとにトピックが 1 つあります。各セクションは角カッコ（[ ]）で囲まれたトピック名を含む行、*[topic-name]* から始まります。使用可能なトピックを次に挙げます。

- CONTAINER
- DB_PARAMETER（サブトピックの指定あり）
  - ACTION_DBA
  - ADANUC_STARTED
  - ADANUC_TERMINATED
  - AUDIT_TRAIL
  - DELETE_CHECKPOINTS
  - OFFLINE_CHECKPOINTS
  - RECOVER_LOST_BLOCKS
  - REORDER_FILE
  - SAVE_DB

  - SSX_CONFIGURATION
  - TERMINATE_ADANUC
- ENVIRONMENT
- NUCPARMS
- RESERVED_LOCATION
- TEMPORARY_LOCATION

トピック：CONTAINER

```
# configuration                                    # recommended values

[CONTAINER]

  ASSOx = <ASSOx device>                           # n lines for ASSO1 ... ASSOn
  DATAx = <DATAx device>                           # n lines for DATA1 ... DATAn
  WORK1 = <WORK1 device>                           # 1 line  for WORK1
  SORTx = <SORTx device>                           # n lines for SORT1 ... SORTn
  TEMPx = <TEMPx device>                           # n lines for TEMP1 ... TEMPn
  NUCCLGx = <NUCCLG device>
  NUCPLGx = <NUCPLG device>

[CONTAINER-END]
```

Adabas コンテナ ASSOx、DATAx、WORK1、SORTx および TEMPx は、このトピックで定義できます。さらに、Adabas シーケンシャルファイルである NUCCLGx および NUCPLGx もこのトピックで定義できます。

これらの項目に対して明示的な外部環境定義が存在する場合、それらの値は DBnnn.INI 値を上書きします。したがって、デフォルト値を定義するために DBnnn.INI の項目を使用することができます。

重要：DB_PARAMETER トピックの INCREASE_<container> アクションを有効にする場合は、新しいコンテナが必要なときに使用するディスクの場所を AEO に知らせる必要があります。これは、少なくとも 1 つの余分な DATAx と ASSOx をリストに追加し、INCREASE_<container> アクションの開始時に AEO が使用する場所をそれらに定義することによって行います。

ユーティリティ ADADBM を使用してコンテナを追加または削除すると、DBnnn.INI ファイル内のコンテナ定義が更新されます。

トピック：DB_PARAMETER

トピック DB_PARAMETER は AEO のアクションおよびその他を定義します。

サブトピック：ACTION_DBA

トピック ACTION_DBA には、必要に応じて、AEO を開始できるすべてのユーザーの名前が含まれています。Adabas と全ユーティリティは他のユーザーに対しても動作しますが、トピック DB_PARAMETER に定義されたすべてのアクションは、ACTION_DBA として定義されているユーザーに対してだけ動作します。

UNIX：ユーザー名では大文字と小文字が区別されます。

PC プラットフォーム：定義されているユーザーがなく、かつ AEO が使用可能な場合、それぞれのアクションがすべてのユーザーに対して開始されます。

```
# configuration                                    # recommended values

[DB_PARAMETER]

  [ACTION_DBA]
    <user name 1>                              (see platform-specific notes given ↩
above)
    <user name 2>
    ...
  [ACTION_DBA-END]

[DB_PARAMETER-END]
```

サブトピック：ADANUC_STARTED

```
# configuration                                    # recommended values

[DB_PARAMETER]

  [ADANUC_STARTED]
    ACTION=<yes/no>
    ACTION_ROUTINE=<action routine>                ada_nsta
  [ADANUC_STARTED-END]

[DB_PARAMETER-END]
```

推奨のアクションルーチンは、ada_nsta です。このルーチンには次の入力パラメータがあります。

パラメータ	説明
1	ユーティリティ名 = ADANUC
2	プロセス ID
3	DBID
4	DB ステータス = ONLINE
5	ユーザーログイン名
6	ノード名
7	セッション番号

ニュークリアスが正常に開始されたときにのみ、Adabas ユーティリティ ADANUC によってアクションルーチンがサブミットされます。

ACTION_ROUTINE は、ACTION=yes の場合のみサブミットされます。

サブトピック：ADANUC_TERMINATED

```
# configuration                                    # recommended values

[DB_PARAMETER]

  [ADANUC_TERMINATED]
    ACTION=<yes/no>
    ACTION_ROUTINE=<action routine>                ada_nsto
  [ADANUC_TERMINATED-END]

[DB_PARAMETER-END]
```

推奨のアクションルーチンは、ada_nsto です。このルーチンには次の入力パラメータがあります。

パラメータ	説明
1	ユーティリティ名 = ADANUC
2	プロセス ID
3	DBID
4	DB ステータス = ONLINE
5	ユーザーログイン名
6	ノード名
7	終了ステータス（=TERMINATED または ABORTED）

ニュークリアスが終了するときにのみ、Adabas ユーティリティ ADANUC によってアクションルーチンがサブミットされます。

ACTION_ROUTINE は、ACTION=yes の場合のみサブミットされます。

サブトピック：AUDIT_TRAIL

```
# configuration                                    # recommended values

[DB_PARAMETER]

  [AUDIT_TRAIL]
    ACTION=<yes/no>
    FILTER=<all/rejected>
  [AUDIT_TRAIL-END]

[DB_PARAMETER-END]
```

監査証跡を書き込むには、ACTION を YES に設定します。デフォルト値は NO です。

拒否された認証のみをログに記録するには、FILTER を REJECTED に設定します。デフォルト値は ALL です。

サブトピック：DELETE_CHECKPOINTS

```
# configuration                                                # recommended values

[DB_PARAMETER]

  [DELETE_CHECKPOINTS]
    ACTION=<yes/no>
    ACTION_ROUTINE=<action routine>                            ada_dlcp
    MINIMUM=<minimum age (in days) for checkpoints             100
                to be deleted>
  [DELETE_CHECKPOINTS-END]

[DB_PARAMETER-END]
```

推奨のアクションルーチンは ada_dlcp です。このルーチンには次の入力パラメータがあります。

パラメータ	説明
1	ユーティリティ名 = ADANUC
2	プロセス ID
3	DBID
4	DB ステータス = ONLINE
5	ユーザーログイン名
6	ノード名
7	ADADBM コマンド DELCP の日付文字列

次のときにのみ、Adabas ユーティリティ ADANUC によってアクションルーチンがサブミットされます。

- ニュークリアスが開始するとき
- ニュークリアスがチェックポイントを書き出し、チェックポイントが最後に書かれた日が短くても 1 日前であるとき（この場合は通常、いつもオンラインであるデータベースに起こります）

ACTION_ROUTINE は、ACTION=yes の場合のみサブミットされます。

MINIMUM は 0 より大きい値である必要があります。この値は、削除されるチェックポイントの最小経過時間（日数）を定義します。

サブトピック：INCREASE_<container>

```
# configuration                                                # recommended values

[DB_PARAMETER]

  [INCREASE_<container>]                                       ASSO or DATA
    ACTION=<yes/no>
    ACTION_ROUTINE=<action routine>                            ada_iass or ada_idat
    MINIMUM=<minimum free space (in percent)>                  15
    EXTEND_RATE=<minimum extend rate (in percent)>             10
    MESSAGE = (I=<i>,W=<w>,E=<e>,F=<f>)                        (I=40,W=20,E=10,F=5)
  [INCREASE_<container>-END]

[DB_PARAMETER-END]
```

推奨されるアクションルーチンは、ada_iass（ASSO の場合）および ada_idat（DATA の場合）であり、どちらにも次の入力パラメータがあります。

パラメータ	説明
1	ユーティリティ名（大文字）
2	プロセス ID
3	DBID
4	DB ステータス（ONLINE または OFFLINE）
5	ユーザーログイン名
6	ノード名
7	ASSO/DATA エクステントの現在の数（7 など）
8	新しいエクステント名（ASSO8 など）
9	エクステントのサイズ（MB 単位）

アクションルーチンは、次の Adabas ユーティリティによってサブミットされます。

- ADAREP は、"adarep dbid=<dbid> free" の実行時にアクションを非同期でサブミットします。
- ADANUC は、スペース割り当ての後にアクションを非同期でサブミットします。
- ADAFDU、ADABCK、ADAFRM、ADAREC、ADAORD は、ADANUC のオフライン時にスペース割り当てが行われた後、アクションを非同期でサブミットします。
- ADAMUP、ADAINV は、adanuc のオフライン時にスペース割り当てが行われている間に、アクションを同期的にサブミットします。

メッセージ DBFREE は、スペース割り当てを行うすべての Adabas ユーティリティおよび adarep によってログとして記録されます。

ACTION_ROUTINE は、ACTION=yes の場合のみサブミットされます。

アナライザは、現在の空きスペース（% 単位）と MINIMUM の設定値（有効範囲：5〜50 %）を比較します。指定された最小値よりも空きスペースの容量が少ない場合、アナライザは必要なスペースの容量を計算し、追加スペースを割り当てるためのアクションを開始します。

追加スペースの計算は、次の 2 つのステップによって行われます。

1. 現在のデータベースのサイズに、EXTEND_RATE の値（有効範囲：5〜100 %）を掛けて追加スペースを計算します。このように定義すると、現在のサイズに応じた追加スペースを特定できます。ここで計算された追加スペースは、ステップ 2 で算出される値に加えます。
2. ステップ 1 で算出した追加スペースを加えても、全データベーススペースに対する空きスペースの割合が、MINIMUM に指定された割合よりも少ない場合があります。大きいブロックが割り当てられ、それが原因で空きスペースが必要最低限の割合に到達し、さらに大幅に不足するような場合に、こうした状況が発生する可能性があります。この場合は、MINIMUM に指定された割合に到達するように、再度スペースを割り当てる必要があります。

このため、割り当てる追加スペースの容量は、ステップ 1 と 2 の値を加えたものになります。

Adabas は、ニュークリアスがアクティブな間は ASSO/DATA を増加させることができます。ニュークリアスがアクティブな場合、スペース割り当ては ADANUC から実行されます（ユーティリティに対しても同様）。アクションルーチンは、新しいコンテナ定義をトピック CONTAINER から取得するため、このアクション用に 1 つ以上の CONTAINER 項目を定義しておく必要があります。こうした定義がない場合には、データベースのサイズは増えないため、アクションが停止します。

アナライザは、計算された空きスペースの割合を項目 MESSAGE の定義済みパーセント値 i、w、e、f と比較します。これらの値は、1〜100 の範囲で降順（i > w > e > f）に定義されている必要があります。これらの値と比較して、フリースペースが同じか小さい場合には、メッセージ DBFREE が記録されます。

サブトピック：OFFLINE_CHECKPOINTS

```
# configuration                                                # recommended values

[DB_PARAMETER]

  [OFFLINE_CHECKPOINTS]
    MESSAGE = (I=<i>,W=<w>,E=<e>,F=<f>)                        (I=50,W=20,E=5,F=2)
  [OFFLINE_CHECKPOINTS-END]

[DB_PARAMETER-END]
```

項目 MESSAGE は Adabas メッセージ CPBFREE の重要度を定義します。Adabas がオフラインのときは、Adabas チェックポイントがチェックポイントブロックに書き出されるたびに、AEO アナライザがコールされます。アナライザはチェックポイントブロック内の空きエントリ数を見つけて、MESSAGE パラメータ i、w、e、f の数値（0 より大きな値である必要があります）と比べます。これらの値は降順、つまり i > w > e > f の順に定義されている必要があります。したがって、空きエントリ数が 10 で MESSAGE=(I=50,W=20,E=5,F=2) の場合は、重要度 W の Adabas メッセージ CPBFREE が書き出されます。

注意: メッセージ CPBFREE は、ニュークリアスのオフライン時にチェックポイントを書き出す、すべての Adabas ユーティリティによってログとして記録されます。

サブトピック：RECOVER_LOST_BLOCKS

```
# configuration                                                # recommended values

[DB_PARAMETER]

  [RECOVER_LOST_BLOCKS]
    ACTION=<yes/no>
    ACTION_ROUTINE=<action routine>                            ada_rlst
  [RECOVER_LOST_BLOCKS-END]

[DB_PARAMETER-END]
```

推奨のアクションルーチンは ada_rlst です。このルーチンには次の入力パラメータがあります。

パラメータ	説明
1	ユーティリティ名（大文字）
2	プロセス ID
3	DBID
4	DB ステータス（ONLINE または OFFLINE）
5	ユーザーログイン名
6	ノード名

アクションルーチンは、次の Adabas ユーティリティによってサブミットされます。

- Adabas リターンコード 77 が発生し、かつ関連コマンドが Adabas ユーティリティコマンドの場合は、Adabas ユーティリティ ADANUC
- Adabas ユーティリティ ADAREP："adarep dbid=<dbid> layout"
- Adabas ユーティリティ ADAVFY："adavfy dbid=<dbid> lost"

ACTION_ROUTINE は、ACTION=yes の場合のみサブミットされます。

ブロックの損失は、ユーティリティの処理が原因で起こります。この状態のとき AEO が有効でなく、かつ損失ブロックがすでに存在する場合は、ADAREP または ADAVFY を使用して、損失ブロックを検出し、このアクションをサブミットできます。

注意: RECOVER_LOST_BLOCKS は指定されたデータベースの UCB エントリの暗黙的なリセットを行います。

サブトピック：REORDER_FILE

```
# configuration                                                # recommended values

[DB_PARAMETER]

  [REORDER_FILE]
    ACTION=<yes/no>
    ACTION_ROUTINE=<action routine>                            ada_reor
    MAXIMUM=<maximum number of AC/NI/UI/DS extents>            12
    MESSAGE =(I=<i>,W=<w>,E=<e>,F=<f>)                         (I=5,W=8,E=12,F=14)
  [REORDER_FILE-END]

[DB_PARAMETER-END]
```

推奨のアクションルーチンは ada_reor です。このルーチンには次の入力パラメータがあります。

パラメータ	説明
1	ユーティリティ名 = ADAREP
2	プロセス ID
3	DBID
4	DB ステータス（ONLINE または OFFLINE）
5	ユーザーログイン名
6	ノード名
7～106	Adabas ファイル番号リスト

アクションルーチンは、SAVE_DB アクションの処理の中で、adarep dbid=<dbid> content というパラメータを指定した Adabas ユーティリティ ADAREP をコールするとサブミットされます。

メッセージ FIFREE は、スペース割り当てを行うすべての Adabas ユーティリティによってログとして記録されます。

ACTION_ROUTINE は、ACTION=yes の場合のみサブミットされます。

ADAREP ユーティリティは AEO アナライザをコールします。AEO アナライザは、すべてのデータベースユーザーファイル（Adabas システムファイルは無視される）の AC/NI/UI/DS エクステントの数をチェックします。いずれかのエクステントの数が MAXIMUM（有効範囲は3～32）と同じか大きいと、このファイルはリオーダされます。一度アクションを開始すると、このようなファイルすべてがリオーダされます（アクションパラメータ 7 - 106 参照）。したがって、リオーダタスクは他のファイルの後に続けて 1 ファイルに対して実行されます。

重要：Adabas がエクステントを割り当てるときに、Adabas はまだファイルを使用しているので、この ACTION_ROUTINE は直接サブミットされません。したがって、アクセスとリオーダアクションとの間に不整合が発生します。リオーダアクションはデータベースが正常に保管された後でだけ開始できます。アクション REORDER_FILE は、アクション SAVE_DB（「サブトピック：SAVE_DB」セクションを参照）が正常終了したかどうかチェックします。これは、入力パラメータ 3 を持つ環境変数 ADA_SAVE_DB の値と比較することによって行われます。この値が一致した場合に、リオーダが開始します。他のすべてのコンテキストにおいて、アクションはファイルをリオーダせずに終了します。

Adabas がファイルエクステントを割り当てるたびに、アナライザは現在のエクステント数と定義済みのメッセージ値 i、w、e、f を比較します。これらの値は 1 より大きく、かつ昇順（i < w < e < f）になっている必要があります。いずれかのエクステントタイプ（AC、NI、UI、DS）が定義済みのメッセージ値 i、w、e、f のいずれかと同じか大きいすべてのファイルについて、Adabas オペレーションメッセージ FIFREE がログに記録されます。

サブトピック：SAVE_DB

```
# configuration                                                # recommended values

[DB_PARAMETER]

  [SAVE_DB]
    ACTION=<yes/no>
    ACTION_ROUTINE=<action routine>                            ada_svdb
  [SAVE_DB-END]

[DB_PARAMETER-END]
```

推奨のアクションルーチンは、ada_svdb です。このルーチンには次の入力パラメータがあります。

パラメータ	説明
1	データベース ID = dbid

このアクションルーチンは、外部スケジューラと組み合わせ、ada_actn を使って開始する必要があります。次の手順で行います。

- アクション定義を確認します（ACTION が yes であること）。
- バックアップファイル名に拡張子 <dbid>.<nn> を付けて、次のように指定します。

  （PC プラットフォーム：）BCK001=%ADADATADIR%\db<dbid>\BCK001.<dbid>.<nn>（<nn> は、そのディレクトリで見つかった最後のバックアップファイルより 1 つ大きい 01から 99 の範囲の数値）

  （UNIX：）BCK001=$ADADATADIR/db<dbid>/BCK001.<dbid>.<date>.<time>（<date> は yymmdd（年、月、日）の形式、<time> は hhmm（時、分）の形式）
- DUMP オプションを指定してバックアップを開始します。
- バックアップ成功後：環境変数 ADA_SAVE_DB=<dbid> を設定します。
- 'adarep dbid=<dbid> content' を開始して、REORDER_FILE アクションの状態をチェックします。

注意: バックアップオプションまたはバックアップの場所を変更したいときは、ada_svdb をカスタマイズしてユーザー固有の条件に合わせます。

サブトピック：SSX_CONFIGURATION

```
# configuration                                    # recommended values

[DB_PARAMETER]

  [SSX_CONFIGURATION]
    <option>=<value>
  [SSX_CONFIGURATION-END]

[DB_PARAMETER-END]
```

各種認可タイプのサンプルとサンプルテンプレートについては、「*Adabas 認証*」セクションの「*SSXLoginModule コンフィグレーションテンプレート*」セクションと、『*ユーティリティ*』ドキュメントの「*付録C*」を参照してください。これらのテンプレートは、一部の設定がユーザーに固有であり、必要に応じて変更する必要があるので、完全ではありません。

サブトピック：TERMINATE_ADANUC

```
# configuration                                          # recommended values

[DB_PARAMETER]

  [TERMINATE_ADANUC]
    SHUTDOWN=<n1>                                            0
    CANCEL=<n2>                                              12
    ABORT=<n3>                                               12
  [TERMINATE_ADANUC-END]

[DB_PARAMETER-END]
```

トピック TERMINATE_ADANUC には adastop の情報が含まれています。項目値 n1、n2、n3 は数値（0 以上）でなければなりません。

adastop は、ADAOPR ユーティリティを実行してデータベースを停止します。ADAOPR には、データベースを停止する 3 つのオプションがあります。

1. ADAOPR SHUTDOWN は、すべてのトランザクションが終了するまで待機し、データベースをシャットダウンします。
2. ADAOPR CANCEL は、すべてのオープントランザクションを直ちにロールバックし、データベースをシャットダウンします。
3. ADAOPR ABORT は、クリーンシャットダウンを実行せずにデータベースを停止し、ダンプを書き込みます。

adastop は次の処理を行います。

1. n 1 > 0 の場合、adastop はまず ADAOPR SHUTDOWN を実行し、データベースの終了を最大 n1 * 5 秒間待機します。n1 = 0 の場合、ADAOPR SHUTDOWN は実行されません。
2. ステップ 1 の後にデータベースがまだ終了しておらず、かつ n2 > 0 の場合、adastop は ADAOPR CANCEL を実行して、データベースの終了を最大 n2 * 5 秒間待機します。n2 = 0 の場合、ADAOPR CANCEL は実行されません。
3. ステップ 2 の後にデータベースがまだ終了しておらず、かつ n3 > 0 の場合、adastop は ADAOPR ABORT を実行して、データベースの終了を最大 n3 * 5 秒間待機します。n3 = 0 の場合、ADAOPR ABORT は実行されません。

ステップ 3 の後でデータベースがまだ終了していない場合は、adastop が異常終了し、エラーメッセージが表示されます。

例：

推奨値の場合、adastop は ADAOPR SHUTDOWN を実行せずに、直接 ADAOPR CANCEL を開始します。12 * 5 秒 = 1 分経過してもデータベースが停止しない場合は、ADAOPR ABORT が

実行されます。その後、12 * 5 秒 = 1 分経過してもデータベースが停止しない場合は、adastop が異常終了し、エラーメッセージが表示されます。

トピック：ENVIRONMENT

すべての Adabas 環境変数（ニュークリアスコンテナと SAM ファイルを除く）をこのトピック内に定義できます。これらの項目に明示的な外部環境定義があると、外部値は DBnnn.INI 値を上書きします。したがって、DBnnn.INI にこれらの項目を使用すれば、デフォルト値を定義できます。

```
# configuration                       # recommended values

[ENVIRONMENT]                         # Any Adabas environment variable may
                                      # be defined here.

  ADAUEX_n = ...                      # Adabas user exit

  ADATRT = ....                       # file name of the shared library (UNIX)
                                      # or dynamic link library (PC) containing
                                      # the Adabas translation table

  ADAHYX_n                            # Adabas hyperexit

[ENVIRONMENT-END]
```

上記の環境設定は、対応する環境変数が定義されていないときに Adabas ユーティリティで使います。

トピック：NUCPARMS

```
# configuration                       # recommended values

[NUCPARMS]

  PLOG                                # Any nucleus parameter may be defined here.
  NU=100
  ....

[NUCPARMS-END]
```

トピック NUCPARMS は、DBID を除くすべてのニュークリアスパラメータを定義します。この値はデータベース開始時に adastart に使用されます。DBID は adastart によってニュークリアスに渡されます。adastart はパラメータとして DBID を受け取ります。

トピック：RESERVED_LOCATION

```
# configuration                      # recommended values

[RESERVED_LOCATION]
  LOCATIONx = <path name>            # n lines for disk locations with free space
[RESERVED_LOCATION-END]
```

データベースの自動拡張が必要で、新規コンテナを作成するとき、Adabas は、指定された位置で空きスペースを探します。

トピック：TEMPORARY_LOCATION

```
# configuration                      # recommended values

[TEMPORARY_LOCATION]
  TEMPLOCx = <path name>            # n lines for disk locations with free space
[TEMPORARY_LOCATION-END]
```

ニュークリアスまたはユーティリティのために Adabas が一時的なディスクスペースを必要とするとき、Adabas は指定された位置で空きスペースを探します。

4 アクションテンプレート

次の表は、配布キットに収録されている AEO のアクションのテンプレートをリストしています。テンプレートのディレクトリは、*%ADAPROGDIR%\tools*（Windows 環境）、*$ADAPROGDIR/tools*（UNIX 環境）となっています。

注意: 使用環境に合わせてテンプレートの設定を変更する必要があります。

サブトピック *ARCHIVE_LOGFILE* および *SAVE_DB* だけは、ada_actn を使って開始することを強く推奨します。他のすべてのアクションは、AEO 自身の管理下で使用したほうが良いでしょう。

たいていのアクションテンプレートはバッチファイルとして導入されています。しかし、それらはバイナリ実行形式ファイルで置き換え可能です。

アクションテンプレート	説明
ada_actn	Adabas アクションを開始します。 アクションルーチン名を読み込み、そのルーチンを開始します（実行例については、サブトピック SAVE_DB および ARCHIVE_LOGFILE を参照）。
ada_alrt	環境変数 ADA_ALRT_LIST にリストされているすべてのユーザーに通知をします。
ada_dlcp	古いチェックポイントを削除するために "adadbm dbid=<nnn> delcp=" を使用します。
ada_iass	トピック CONTAINER 内に新しいコンテナが定義されているかどうかチェックします。定義されている場合、"adadbm dbid=<nnn> add_container" をコールしてコンテナを追加します。
ada_idat	トピック CONTAINER 内に新しいコンテナが定義されているかどうかチェックします。定義されている場合、"adadbm dbid=<nnn> add_container" をコールしてコンテナを追加します。

アクションテンプレート	説明
ada_inuc	次の式を使用して、ニュークリアスパラメータの新しい値を計算します。 新しい値 = (現在の値) * 1.1 + (デフォルト値) / 10 （ニュークリアスパラメータのデフォルト値については、『Adabas ユーティリティ』の adanuc の章を参照） DB<nnn>.INI のニュークリアスパラメータを変更します。 ニュークリアスパラメータ TNAA、TNAE、TNAX および TT については、adaopr を使用して、現在のニュークリアスセッションのパラメータ値を大きくします。 新しいパラメータ値の計算は、プログラム adainuc.exe で実行されます。このファイルは、他の値の計算用に変更できるように、ソースに含まれています（EXAMPLE\AEO ディレクトリに adainuc.c が含まれています）。
ada_nsta	空です。
ada_nsto	空です。
ada_reor	環境変数 ADA_SAVE_DB をチェックして、このアクションがデータベースバックアップアクション SAVE_DB の正常終了後に稼動しているのかどうかを確認します（そうでない場合は、このアクションを中止します）。 ファイルリストのすべてのファイル（パラメータ 7～106）に対して、次の処理を行います。 - "adaord dbid=<dbid> export=<fnr>" を使用してファイルをセーブします。 - "adadbm dbid=<dbid> delete=<fnr>" を使用してファイルを削除します。 - "adaord dbid=<dbid> import=<fnr>" を使用してファイルをロードします。
ada_rlst	adanuc がオンライン（DB ステータスが ONLINE である）かどうかチェックします。 "adadbm dbid=<nnn> recover" を呼び出します。
ada_svdb	SAVE_DB の定義をチェックします。 バックアップファイル名を次のように拡張します。 BCK001=%ADADATADIR%\db<nnn>\BCK001.<dbid>.<xx>（<xx> は範囲 01～99 の数字） バックアップを実行します。 xx は 2 桁の数字です。ada.svdb はファイル %BCK001 が存在しない場合一番小さい番号を検索します。 バックアップが成功すると、環境変数 ADA_SAVE_DB が設定され、"adarep dbid=<nnn> content" が開始されて、REORDER_FILE のアクション条件がチェックされます（REORDER_FILE アクションが有効で、かつ条件が真の場合、adarep はアクション REORDER_FILE を開始します）。
ada_svlg	トピック ARCHIVE_LOGFILE の定義をチェックします。 トピック LOGGING の LOG_FILE の項目を読み込み、書き込みが可能かどうかをチェックします。 ログファイルが空の場合、何の処理も実行しません。 一番古いログファイルを削除します（ファイル名は <LOG_FILE>.<MAXIMUM>）。 1 から <MAXIMUM-1> の範囲のすべてのログファイル名を変更します。<LOG_FILE>.05 が <LOG_FILE>.06 になるように、ログ番号を 1 増やします。 一番新しいログファイルの名前を <LOG_FILE> から <LOG_FILE>.01 に変更します。

5 管理コマンド

- コンフィグレーションファイルに格納されている情報の取得と変更：adaini 38
- コンフィグレーションファイルのインストール：adainst 42
- データベースの停止：adakill 44
- ログファイルの表示：adalog 44
- ログファイルにメッセージを書き出す：adamsg 44
- デフォルトデータベースの定義：adaset 45
- データベース（複数）の表示：adashow 46
- データベースの開始：adastart 47
- データベースの停止：adastop 48

この章では、Adabas拡張オペレーション（AEO）で一般的に使用するコマンドについて説明します。

コンフィグレーションファイルに格納されている情報の取得と変更：adaini

使用法：*adaini [DBID=<dbid>] {<add>｜<del>｜<show>}*

<dbid>は、0～255の範囲の数値です。DBIDパラメータが指定されていない場合、またはDBID=0が指定されている場合は、ADABAS.INIファイルが処理されます。それ以外の場合は、DB<dbid>.INIファイルが処理されます。

コンフィグレーションファイルへの情報の追加または変更

<add>は、コンフィグレーションファイルに1つ以上の項目を追加または変更する際に使用します。構文は次のとおりです。

{ADD｜MOD[IFY] } <topic_list> <item_value_list>

注意: ADDとMOD[IFY]は同じです。ADDを使用して項目を変更することも、MOD[IFY]を使用して項目を追加することもできます。

<topic_list> の構文は次のとおりです。

{ TOPIC=<topic> } ...

ここで、<topic> はトピックの名前です。処理する項目がサブトピックに属している場合は、その項目が属しているトピックの完全な階層を指定する必要があります。

注意: トピック名は大文字に変換されます。

<item_value_list> の構文は次のとおりです。

{ITEM=<item>[=<value>] } ...

ここで、<item> は項目の名前で、<value> は項目の値です。

注意:

1. 項目は、値ありでも値なしでも定義できます。
2. トピック名とは異なり、項目名と項目値は大文字に変換されません。

3. adaini では、指定したトピックや項目が Adabas で実際に使用されているかどうか、指定した項目値が有効かどうかはチェックされません。adaini は、コンフィグレーションファイルの構文が正しいことのみを保証します。

例

```
adaini mod topic=node_parameter topic=analyser item=ACTION=no
```

このコマンドで、トピック NODE_PARAMETER、サブトピック ANALYSER の項目 ACTION が no に設定され、その結果 AEO が非アクティブ化されます。

コンフィグレーションファイルからの情報の削除

<del> は、コンフィグレーションファイルから 1 つ以上の項目を削除する際に使用します。構文は次のとおりです。

DEL[ETE] <topic_list> <item_list>

<topic_list> は、<add> の場合と同じ方法で使用されます。

<item_list> の構文は次のとおりです。

*{ITEM= *}* | *{ITEM = <item>} ...*

ここで、<item> は項目の名前です。指定した項目が削除されます。「*」を指定すると、すべての項目とその項目が属しているトピックが削除されます。

注意:

1. Unix シェルでは「*」は特殊文字となっているので、前にバックスラッシュを付けるか、引用符または二重引用符で囲む必要があります。
2. トピックに属しているすべての項目を明示的に指定した場合は、項目のみが削除され、項目が属しているトピックは、コンフィグレーションファイルに空のトピックとして残ります。トピックも削除するには、ITEM=* を指定する必要があります。

例

```
adaini dbid=36 del topic=environment item=ADAHYX_4
```

このコマンドは ADAHYX_4 の環境設定を削除し、ハイパー出口 4 を非アクティブ化します。

コンフィグレーションファイルの情報の表示

<show> は、コンフィグレーションファイルに格納されている 1 つ以上の項目を表示する際に使用します。構文は次のとおりです。

show [<format>] <topic_list> [<item_list>]

<format> の構文は次のとおりです。

FORMAT={ BAT | BSH | CMD | CSH}

FORMAT を指定した場合、指定したシェルのステートメントが生成されます。これにより環境変数が作成されますが、その名前は処理する項目名になり、その値は項目の値になります。

<topic_list> は、<add> の場合と同じ方法で使用されます。

<item_list> の構文は <del> と同じです。ITEM=* を指定した場合は、指定したトピックに属しているすべての項目が表示されます。<format>が指定されていない場合、これらは<item>=<value> のフォーマットで表示され、その後に改行が続きます。項目を名前で明示的に指定した場合は、これらの項目が表示されます。<format>が指定されていない場合は、項目の値のみが表示され、その後に改行が続きます。

<item_list> が指定されていない場合は、トピックに属しているすべての項目が表示されます。フォーマットが指定されていない場合、これらは <item>=<value> のフォーマットで表示されます。サブトピックも表示されます。以下の例にレイアウトを示します。

例

このコマンドは、

```
adaini show topic=db_list
```

次の出力を生成する場合があります。

```
[DBID_001]
  INI_FILE=C:\Program Files\Software AG\Adabas\db001\DB001.INI
  NAME=V33-DATABASE
  STRLVL=12
[DBID_001-END]

[DBID_002]
  AUTOSTART=V616
  INI_FILE=C:\Program Files\Software AG\Adabas\db002\DB002.INI
  NAME=P289591
  STRLVL=12
[DBID_002-END]

[DBID_003]
  INI_FILE=C:\Program Files\Software AG\Adabas\db003\DB003.INI
```

```
  NAME=DEFAULT-DATABASE
  STRLVL=15
[DBID_003-END]

[DBID_004]
  INI_FILE=C:\Program Files\Software AG\Adabas\db004\DB004.INI
  NAME=TEST-ICU
  STRLVL=15
[DBID_004-END]

[DBID_005]
  INI_FILE=C:\Program Files\Software AG\Adabas\db005\DB005.INI
  NAME=ADA618-DB
  STRLVL=15
[DBID_005-END]

[DBID_012]
  INI_FILE=C:\Program Files\Software AG\Adabas\db012\DB012.INI
  NAME=GENERAL_DATABASE
  STRLVL=15
[DBID_012-END]

[DBID_036]
  AUTOSTART=NO
  INI_FILE=C:\Program Files\Software AG\Adabas\db036\DB036.INI
  NAME=GENERAL_DATABASE
  STRLVL=15
[DBID_036-END]

[DBID_062]
  AUTOSTART=NO
  INI_FILE=C:\Program Files\Software AG\Adabas\db062\DB062.INI
  NAME=GENERAL_DATABASE
  STRLVL=16
[DBID_062-END]
```

このコマンドは、

```
adaini dbid=36 show format=bat topic=backup item=BCK001 item=BCK002 item=BCK003
```

次の出力を生成する場合があります。

```
set BCK001=C:\Program Files\Software AG\Adabas\db036\BCK001.036
set BCK002=C:\Program Files\Software AG\Adabas\db036\BCK002.036
set BCK003=C:\Program Files\Software AG\Adabas\db036\BCK003.036
```

コンフィグレーションファイルのインストール：adainst

PC プラットフォームの場合：

用途：*adainst* <*dbid*>

<*dbid*> が欠落し、ADABAS.INI が %ADADATADIR%\etc に存在しない場合、*adainst* は、%ADADATADIR%\etc\ADABAS.INI を作成します。

このスクリプトにより、次の手順が実行されます。

- ディレクトリ %ADADATADIR%\etc の作成
- まだ存在しない場合は、テンプレートファイル %ADAPROGDIR%\ADABAS.INI を %ADADATADIR%\etc\ADABAS.INI にコピーします。
- %ADADATADIR%\etc\ADABAS.INI がすでに存在している場合、それにトピック NODE_PARAMETER と、DB_DEFAULTS トピックの DB_PARAMETER サブトピックがすでに含まれているかどうかをチェックします。含まれていない場合、テンプレートからコピーします。
- 必要に応じて次の値を代用します。
  トピック MISCELLANEOUS の NODE_NAME
  トピック NODE_PARAMETERS 内のサブトピック LOGGING の LOG_FILE

<*dbid*> が指定されていて、かつトピック DBID_<*dbid*> が ADABAS.INI に存在しない場合、*adainst* は %ADADATADIR%\db<*dbid*>\DB<*dbid*>.INI を作成します。このスクリプトにより、次の手順が実行されます。

- ADABAS.INI からトピック DB_DEFAULTS を読み取り、それを DB<*dbid*>.INI にコピーします。
- ディレクトリ %ADADATADIR%\db<*dbid*> 内の *assign.*sh* および *adanuc.*sh* ファイルを検索し、コンテナ定義をトピック CONTAINER にコピーします。
- ディレクトリ %ADADATADIR%\db<*dbid*> 内の *adanuc.*sh* ファイルを検索し、ニュークリアスパラメータをトピック NUCPARMS にコピーします。
- トピック ACTION_DBA 内のユーザー名を問い合わせます。
- 有効／無効のアクションを表示します。
- adarep dbid=<*dbid*>summary を使って、データベース名を取得し、項目 NAME を ADABAS.INI のトピック DBID_<*dbid*> に挿入します。

UNIX の場合：

用途：*adainst* <*dbid*>

<*dbid*> が欠落し、ADABAS.INI が $ADADATADIR/etc に存在しない場合、*adainst* は、$ADADATADIR/etc/ADABAS.INI を作成します。

このスクリプトにより、次の手順が実行されます。

- ディレクトリ $ADADATADIR/etc の作成
- まだ存在しない場合は、テンプレートファイル $ADAPROGDIR/ADABAS.INI を $ADADATADIR/etc/ADABAS.INI にコピーします。
- $ADADATADIR/etc/ADABAS.INI がすでに存在している場合、それにトピック NODE_PARAMETER と、DB_DEFAULTS トピックの DB_PARAMETER サブトピックがすでに含まれているかどうかをチェックします。含まれていない場合、テンプレートからコピーします。
- コピー先ファイルで次の値を置き換えます。
  トピック MISCELLANEOUS の NODE_NAME
  トピック LOGGING（NODE_PARAMETER）内の LOG_FILE

<*dbid*> が指定されていて、かつトピック DBID_<*dbid*> が ADABAS.INI に存在しない場合、*adainst* は $ADADATADIR/db<*dbid*>/DB<*dbid*>.INI を作成します。このスクリプトにより、次の手順が実行されます。

- ADABAS.INI からトピック DB_DEFAULTS を読み取り、それを DB<*dbid*>.INI にコピーします。
- ディレクトリ $ADADATADIR/db<*dbid*> 内の *assign.*sh* および *adanuc.*sh* ファイルを検索し、コンテナ定義をトピック CONTAINER にコピーします。
- ディレクトリ $ADADATADIR/db<*dbid*> 内の *adanuc.*sh* ファイルを検索し、ニュークリアスパラメータをトピック NUCPARMS にコピーします。
- トピック ACTION_DBA 内のユーザー名を問い合わせます。
- 有効／無効のアクションを表示します。
- adarep dbid=<*dbid*>summary を使って、データベース名を取得し、項目 NAME を ADABAS.INI のトピック DBID_<*dbid*> に挿入します。

データベースの停止：adakill

使用法：*adakill <dbid>*

adakill はデータベース <dbid> の Adabas ニュークリアスを次のように停止します。

- PC プラットフォーム：割り込み（CTRL/BREAK）の送信による停止。<dbid> パラメータは必須です。
- UNIX：UNIX シグナル 15 で停止。<dbid> パラメータは必須です。

注意: このコマンドは、ABORT オプションを指定した adastop がニュークリアスを停止できない場合以外は使わないでください。Adabas はメモリダンプを出力し、次回の起動時に自動再スタートを行います。

このスクリプトコマンドは、次の手順を行います。

- adanuc のプロセス ID を取得します。
- 割り込みを送信します。

ログファイルの表示：adalog

使用法：*adalog [<dbid>] [-t]*

adalog は Adabas ログファイルを表示します。

<dbid> を指定した場合、このデータベースの全エントリを表示します。<dbid> を指定しない場合、すべてのデータベースのエントリを表示します。オプション -t を指定すると、adalog はログファイルの最後の行を表示し、ログファイルに新しい行が書き加えられるつれて、その行に続けて表示していきます。

ログファイルにメッセージを書き出す：adamsg

使用法：*adamsg DBID=<dbid> PID=<プロセス ID> UTILITY=<ユーティリティ名> MESSAGE=<メッセージ ID> TEXT=<メッセージテキスト>*

次のメッセージ ID をサポートします。

- ABORTED

  TEXT には、このメッセージ ID に応じた異常終了の原因が出力されます。

- INCNUCP

  TEXT には、このメッセージ ID に応じた、nucleus parameter=<パラメータ>、current size=<現在のサイズ>、new size=<新サイズ>が出力されます。このオプションはアクション ada_inuc（ニュークリアスパラメータの増加）で使います。

- INP

  TEXT には、このメッセージ ID に応じたパラメータ割り当てが出力されます。

- STARTED

  このメッセージ ID では、TEXT は空白です。

- TERMINATED

  このメッセージ ID では、TEXT は空白です。

adamsg はバッチファイルから Adabas ログファイルへのインターフェイスです。すべてのバッチファイル（AEO アクションと同様に）は、ログメッセージに対してこのインターフェイスを使うことができます。TEXT パラメータは必ずパラメータリストの最後にすることを除けば、パラメータの順番は任意です。TEXT パラメータ値以外のすべてのパラメータ値は大文字に変換されます。

例：

```
adamsg DBID=77 PID=4711 UTILITY=my_uti MESSAGE=ABORTED TEXT=file abc is empty
```

ログファイルの最後に次のメッセージ行を生成します。

```
004711 <date + time> 00077 <user name> %my_uti-F-ABORTED, file abc is empty
```

さらに、現在のユーザー ID（有効な場合）が ACTION_DBA トピックに挿入されます。

デフォルトデータベースの定義：adaset

使用法（PC プラットフォームの場合）：*[CALL] adaset <dbid>*

使用法（UNIX）：*adaset <dbid>*

adaset は次の環境変数を定義します。

- ADADBID=<dbid>（これは、デフォルトのデータベース ID です）
- （PC プラットフォーム：）ADADBDIR=%ADADATADIR%\db<dbid>
  （UNIX：）ADADBDIR=$ADADATADIR/db<dbid>（これは、デフォルトデータベースのデータベースワーキングディレクトリです）。

PC プラットフォーム

さらに、adaset は次のように PATH 変数を次のように拡張します。

```
PATH=%ADAPROGDIR%;%ADAPROGDIR%\tools;%PATH%
```

adasetが、コマンドプロンプトからではなくバッチファイルから実行されるときは、CALL コマンドを使用しなければならないことに注意してください。

UNIX

さらに、adaset は次のように PATH 変数を次のように拡張します。

```
PATH=$ADAPROGDIR:$ADAPROGDIR/tools:$PATH
```

C シェル環境では、adaset は adaset.csh のコールによって作成されるエイリアスです。Bourne シェル環境では、adaset は adaset.bsh のコールによって作成される機能です。adaset を使う前に、次のコマンドのいずれかを発行する必要があります。

```
. $ADATOOLS/adaset.bsh  (Bourne shell)
source $ADATOOLS/adaset.csh (C shell)
```

これらのステートメントは、adaenv.bsh（Bourne シェル）または adaenv.csh（C シェル）によってすでに実行されています。

データベース（複数）の表示：adashow

使用法：*adashow [<dbid>] [-a]*

adashow はデータベース <dbid> の次の情報を表示します。

- データベース ID：ADADBID によって指定された値
- 名前：トピック DBID_<ADADBID の値> 内の NAME
- バージョン：廃止
- コンフィグレーションファイル：トピック DBID_<ADADBID の値> 内の INI_FILE
- ステータス：ONLINE、OFFLINE、ERROR のいずれか

<dbid> がない場合、adashow は、環境変数 $ADADBID に指定されたデフォルトのデータベースの情報を表示します。

オプション -a を指定すると、adashow は Adabas.INI 内の DB_LIST セクションにある、このノード上に構成された Adabas データベースすべてのデータベース ID、名前、バージョン、ステータスを表示します。

データベースの開始：adastart

使用法：*adastart [<dbid>]*

adastart は Adabas データベース <dbid> を開始します。最初にコールされたときにニュークリアスログファイル *adanuc.log,* が作成され、後続のコールでニュークリアスログがタイムスタンプとともに保存されます（*adanuc.log.timestamp*）。

PC プラットフォームの場合：

このスクリプトにより、次の手順が実行されます。

- ニュークリアスがすでにオンラインかどうか調べます。
- Adabas という名前のユーティリティを使用してニュークリアスを開始します。これは DB<dbid>.INI からパラメータを読み込みます。
- ニュークリアスがオンラインになるか、または Adabas エラーが発生するまで待機します。

UNIX の場合：

このスクリプトにより、次の手順が実行されます。

- DB<dbid>.INI からニュークリアスパラメータを読み、ファイル $ADADBDIR\nucparms.<dbid> に書き出します。
- ニュークリアスがすでにオンラインかどうか調べます。
- 環境変数 ADANUCLOGOLD が "COPY" として定義されていて、前のセッションのニュークリアスログが存在する場合（ファイル名 $ADADATADIR/db<dbid>/adanuc.log）、ニュークリアスログが $ADADATADIR/db<dbid>/adanuc.log.old にコピーされます。
- 環境変数 ADANUCLOGOLD が "APPEND" として定義されていて、前のセッションのニュークリアスログが存在する場合（ファイル名 $ADADATADIR/db<dbid>/adanuc.log）、ニュークリアスログが $ADADATADIR/db<dbid>/adanuc.log.old に追加されます。
- パラメータファイル $ADADBDIR/nucparms.<dbid> を使ってニュークリアスを開始します。
- ニュークリアスがオンラインになるか、または Adabas エラーが発生するまで待機します。

データベースの停止：adastop

使用法：*adastop [<dbid>]*

adastop はデータベース <dbid> を停止します。<dbid> がない場合には、adastop は環境変数 ADADBID によって指定されたデフォルトのデータベースを停止します。

このスクリプトにより、次の手順が実行されます。

- ニュークリアスがオンラインかオフラインか調べます。
- DB<dbid>.INI からトピック定義 TERMINATE_ADANUC を読みます。
- トピック内に定義されているシャットダウンオプションを受け取り、定義されている時間の間、ニュークリアスの停止を待機します。

6 Adabas ユーティリティの認可の設定

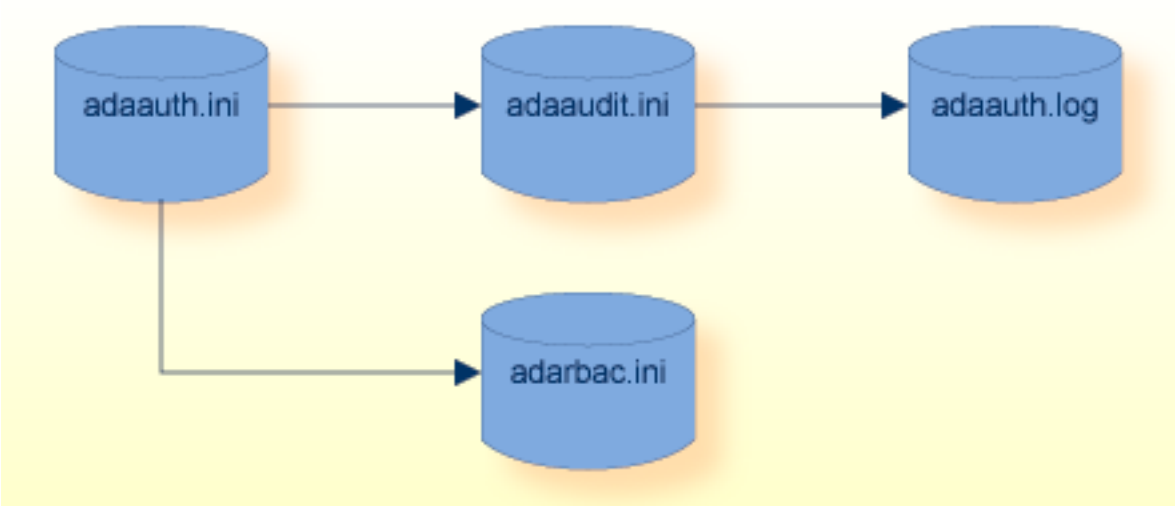
- コンフィグレーションファイルとロギングファイルの場所 50
- adaauth.ini 52
- adaaudit.ini 54
- adarbac.ini 56
- サンプル：adarbac.ini 62

ユーティリティのセキュリティコンフィグレーションは、次のファイルに格納されます。

- adaauth.ini
- adaaudit.ini
- adarbac.ini

これらのファイルは、ローカルマシンのセキュリティを構成し、すべてのデータベース、すべての製品インストール、およびマシン上のバージョン6.5以上の製品バージョンに適用されます。

これらは ASCII ファイルなので、標準のテキストエディタで編集できます。



コンフィグレーションファイルとロギングファイルの場所

コンフィグレーションファイル adaauth.ini は一元的な場所に配置されます。場所はプラットフォーム固有です。また、固定されているので変更できません。初期状態では、ファイル adaaudit.ini と adarbac.ini も事前定義された場所に配置されています。これらのファイルは、必要に応じて他の場所に移動できます。

ファイル	説明	場所固定
adaauth.ini	コンフィグレーションの定義	○
adarbac.ini	セキュリティ定義	×
adaaudit.ini	監査ログのコンフィグレーション	×
adaaudt.log	監査ログ	×

注意: Adabas ユーティリティの実行が許可されているすべてのユーザーは、LOG_FILE とそれが配置されているディレクトリの両方に対して、READ/WRITE アクセス権限を持っている必要があります。

- Windows の場合
- UNIX/Linux の場合
- ファイルおよびディレクトリの権限

Windows の場合

コンフィグレーションフィアルと監査ログファイルは、次の場所にインストールされます。

```
%PROGRAMDATA%\Software AG\Adabas\auth
     adaauth.ini
     adarbac.ini
     adaaudit.ini
```

```
%PROGRAMDATA%\Software AG\Adabas\log
     adaaudit.log
```

UNIX/Linux の場合

コンフィグレーションフィアルと監査ログファイルは、次の場所にインストールされます。

```
/etc/softwareag/Adabas/auth
     adaauth.ini
     adarbac.ini
     adaaudit.ini
```

```
/var/log/softwareag/Adabas
     adaaudit.log
```

ファイルおよびディレクトリの権限

重要: 上記のコンフィグレーションファイルと監査ログファイルは、制限のあるファイル権限なしでインストールされます。データセットのセキュリティ保護（「強化」）方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*」セクションにある「*セキュリティの考慮事項*」を参照してください。

Adabas ユーティリティのすべてのユーザーは、少なくとも次のファイル権限とディレクトリ権限を持っている必要があります。

1. コンフィグレーションファイルに対する READ 権限。
2. 監査ログファイルに対する WRITE 権限（LOG_FILE 設定）。
3. 監査ログファイルが配置されているディレクトリに対する WRITE 権限。

adaauth.ini

コンフィグレーションファイル `adaauth.ini` には、そのマシンとすべてのデータベース、すべての製品のインストール、およびマシン上のバージョン6.5以上の製品バージョンに適用される情報が含まれています。

このファイルには、次の基本的なセキュリティ定義が含まれています。

- セキュリティコンフィグレーション定義の場所。
- 監査コンフィグレーションファイルの場所。

- adaauth.ini の構造
- トピック：AUTHZ
- 項目：ACTION
- 項目：MODE
- 項目：AUDIT_FILE
- 項目：RBAC_FILE

adaauth.ini の構造

コンフィグレーションファイル `adaauth.ini` には、トピック AUTHZ の単一セクションが含まれています。

セクションは角カッコ（[ ]）で囲まれたトピック名を含む行、`[topic-name]` から始まります。セキュリティ定義に関連するトピックを次に挙げます。

- AUTHZ、および項目
  - ACTION
  - AUDIT_FILE
  - MODE
  - RBAC_FILE

トピック：AUTHZ

トピックAUTHZには、ローカルマシンのセキュリティ構成に使用する情報が含まれています。

トピック AUTHZ の構文は次のとおりです。

```
[AUTHZ]
  ACTION     = <activation of feature>
  MODE       = <source of definitions>
  AUDIT_FILE = <path to adaaudit.ini>
  RBAC_FILE  = <path to adarbac.ini>
[AUTHZ-END]
```

項目：ACTION

項目 ACTION は、Adabas ユーティリティの認可機能をアクティブ化します。

- **YES**
  機能が有効になります。
- **NO**
  機能が無効になります。

デフォルト設定は NO です。

重要: 項目 ACTION は、将来のリリースで廃止されます。

項目：MODE

項目 MODE は、セキュリティ定義のソースを定義します。

- **ADABAS**
  セキュリティ定義は RBAC システムファイルに定義されています。
- **INI**
  セキュリティ定義は、コンフィグレーションファイルに定義されています。

項目：AUDIT_FILE

項目 AUDIT_FILE は、監査処理のコンフィグレーション（監査ログのレイアウトや場所など）が含まれている `adaaudit.ini` ファイルの場所を定義します。

項目：RBAC_FILE

項目 RBAC_FILE は、データベースユーティリティの使用に関するセキュリティ定義が含まれている `adarbac.ini` ファイルの場所を定義します。

adaaudit.ini

コンフィグレーションファイル adaaudit.ini には、そのマシンとすべてのデータベース、すべての製品のインストール、およびマシン上のバージョン 6.5 以上の製品バージョンに適用される情報が含まれています。

このファイルには、次の情報が含まれています。

- 監査ファイル処理の基本的なコンフィグレーション（監査ログのレイアウトや場所など）。

- adaaudit.ini の構造
- トピック：AUDIT
- 項目：FORMAT
- 項目：SEPARATOR
- 項目：LOG_FILE

adaaudit.ini の構造

コンフィグレーションファイル adaaudit.ini には、トピック AUDIT の単一セクションが含まれています。

セクションは角カッコ（[ ]）で囲まれたトピック名を含む行、[*topic-name*] から始まります。セキュリティ定義に関連するトピックを次に挙げます。

- AUDIT、および項目
  - FORMAT
  - LOG_FILE
  - SEPARATOR

トピック：AUDIT

トピック AUDIT は、監査ログのパラメータを定義します。

トピック AUDIT の構文は次のとおりです。

```
[AUDIT]
  FORMAT    = <file layout>
  SEPARATOR = <token separator>
  LOG_FILE  = <log file name>
[AUDIT-END]
```

項目：FORMAT

項目 FORMAT は、監査ログエントリのレイアウトを定義します。

- **TEXT**
  監査エントリ内のすべての値の前にヘッダーが付けられて、空白で区切られます。
- **CSV**
  監査エントリ内のすべての値は、区切り値で区切られます。

項目：SEPARATOR

項目 SEPARATOR は、CSV 形式で値を区切る際に使用する文字を定義します。

SEPARATOR で有効なパラメータ値は次のとおりです。

パラメータ値	説明
`","`	コンマ
`";"`	セミコロン
`"/t"`	タブ
`" "`	空白（デフォルト）

パラメータ値は二重引用符で囲む必要があります。

注意: タブのパラメータ値は文字列 `"/t"` です。

項目：LOG_FILE

項目 LOG_FILE は、監査ログの場所とファイル名を定義します。

ログファイルエントリの項目は、項目 FORMAT に設定した値によって異なります。接頭辞が付けられて空白で区切られるか（`FORMAT=TEXT` の場合）、接頭辞なしで選択した区切り値で区切られます（`FORMAT=CSV` の場合、区切り値は `SEPARATOR` で選択した値）。

接頭辞	値	説明
	`2016-06-02T14:48:19Z`	タイムスタンプ
`HOSTNAME=`	*<hostname>*	マシンのホスト名
`OSVERSION=`	*<operating_system>*	オペレーティングシステムの名前とバージョン
	`%AUTHORIZATION-x`	メッセージインジケータ：（I）情報または（E）エラー
`USER=`	UNIX：*<user>* Windows：*<domain>*/*<user>*	ユーザーアカウントの名前
`OPERATION=`	*<operation>*	試行した処理の名前
`DBID=`	*<number>*	データベース ID

接頭辞	値	説明
AUTHORIZED=	YES|NO	認可要求の結果

adarbac.ini

コンフィグレーションファイル adarbac.ini には、そのマシンとすべてのデータベース、すべての製品のインストール、およびマシン上のバージョン6.5以上の製品バージョンに適用される情報が含まれています。

このファイルには、データベースユーティリティの使用に関する、次のようなセキュリティ定義が含まれています。

- ユーザー／役割の割り当て
- 役割／権限の割り当て
- 権限（オブジェクトで許可されている処理の定義）
- オブジェクト定義
- 処理の定義

- adarbac.ini の構造
- トピック：USER_ROLE
- 項目：USER_ROLE 定義
- トピック：ROLE_PERMISSION
- 項目：ROLE_PERMISSION 定義
- トピック：PERMISSIONS
- サブトピック：PERMISSION 定義
- トピック：OBJECTS
- サブトピック：OBJECT 定義
- トピック：OPERATIONS
- 項目：OPERATIONS 定義
- 制限および制約事項

adarbac.ini の構造

コンフィグレーションファイル adarbac.ini は複数のセクションから構成され、セクションごとにトピックが1つ以上あります。ファイルの各セクションは角カッコ（[]）で囲まれたトピック名を含む行、[*topic-name*] から始まります。関連するトピックは次のとおりです。

トピック	説明
USER_ROLE	ユーザーの定義と役割の割り当て。
ROLE_PERMISSION	役割の定義と権限の割り当て。
PERMISSIONS	権限の定義、および処理とオブジェクト（処理を実行可能なオブジェクト）の割り当て。
OBJECTS	オブジェクトの定義とデータベース ID の割り当て。
OPERATIONS	処理の定義と Adabas ユーティリティの割り当て。

トピック：USER_ROLE

トピック USER_ROLE には、ユーザーアカウントへの役割の割り当てが含まれています。

トピック USER_ROLE の構文は次のとおりです。

```
[USER_ROLE]
  <user_name> = <role_assignments>
[USER_ROLE-END]
```

USER_ROLE トピックには、それぞれが 1 つ以上の役割を *user_name* に割り当てる、1 つ以上の USER_ROLE 定義を含めることができます。

項目：USER_ROLE 定義

USER_ROLE 定義は、*user_name* に 1 つ以上の役割を割り当てます。これは USER_ROLE トピック内の項目です。

USER_ROLE 定義項目の構文は次のとおりです。

```
<user_name> = <role_name [,<role_name>] >
```

user_name の値は次のいずれかになります。

- セッションに関連付けられ、オペレーティングシステムによって検証されたユーザーの名前（Windows の場合は「ドメイン\ユーザー」）
- すべてのユーザーを示すアスタリスク（*）

注意: user_name の値は大文字と小文字が区別されないので、大文字と小文字の違いがあっても user_name の値は等しいと見なされます。

role_name の値は次のいずれかになります。

- ROLE_PERMISSION 定義の名前。
- ROLE_PERMISSION 定義のコンマ区切りリスト。

トピック：ROLE_PERMISSION

トピック ROLE_PERMISSION には、役割に対する権限の割り当てが含まれています。

トピック ROLE_PERMISSION の構文は次のとおりです。

```
[ROLE_PERMISSION]
  <role_name> = <permission_assignments>
[ROLE_PERMISSION-END]
```

ROLE_PERMISSION トピックには、それぞれが 1 つ以上の PERMISSION を *role_name* に割り当てる、1 つ以上の ROLE_PERMISSION 定義を含めることができます。

項目：ROLE_PERMISSION 定義

ROLE_PERMISSION 定義は、*role_name* に 1 つ以上の役割を割り当てます。これは ROLE_PERMISSION トピックの項目です。

ROLE_PERMISSION 定義項目の構文は次のとおりです。

```
<role_name> = <permission_name [,permission_name] >
```

`permission_name` の値は次のいずれかになります。

- PERMISSIONS 定義の名前。
- PERMISSIONS 定義のコンマ区切りリスト。

トピック：PERMISSIONS

トピック PERMISSIONSには、権限に対するオブジェクトと処理の割り当てが含まれています。各エントリは、特定のオブジェクトセットで実行可能な、許可された処理のタプルを定義します。

トピック PERMISSIONS の構文は次のとおりです。

```
[PERMISSIONS]
  [<permission_definition>]
[PERMISSIONS-END]
```

PERMISSIONS トピックは 1 つ以上の PERMISSION 定義を含んでいる場合があります。

サブトピック：PERMISSION 定義

各 PERMISSION 定義は、PERMISSIONS トピックのサブトピックです。固有の名前で識別され、ファイルの適切なセクションで定義されている OBJECT および OPERATION 項目が含まれています。

PERMISSION 定義の名前は、一意にする必要があり、サブトピックの場合のようにカッコで囲む必要があります。

トピック PERMISSION 定義のサブトピックの構文は次のとおりです。

```
[<permission_name>]
  OPERATION = <operation_name>
  OBJECT    = <object_name>
[<permission_name>-END]
```

各 PERMISSION 定義には、OPERATION 項目が 1 つと、OBJECT 項目が 1 つ含まれている必要があります。

- OPERATION 項目の値は、実行可能な処理を定義する OPERATION 定義の名前です。
- OBJECT 項目の値は、処理を実行可能な OBJECT 定義の名前です。

トピック：OBJECTS

トピック OBJECTS は 1 つ以上の OBJECT 定義を含んでいます。各エントリは、1 つ以上の PERMISSION 定義で使用でき、1 つ以上のデータベース ID を *object_name* に割り当てます。

トピック OBJECTS の構文は次のとおりです。

```
[OBJECTS]
  <object_definitions>
[OBJECTS-END]
```

サブトピック：OBJECT 定義

各 OBJECT 定義は、OBJECTS トピックのサブトピックです。固有の名前で識別され、DBID 項目が含まれています。

OBJECT定義の名前は、一意にする必要があります。また、サブトピックの場合のようにカッコで囲む必要があります。

トピック OBJECT 定義のサブトピックの構文は次のとおりです。

```
[<object_name>]
  DBID = <number> [, <number>] [, <number>-<number>]
[<object_name>-END]
```

各 OBJECT 定義には、次のいずれかの DBID エントリが含まれている必要があります。

- 単一のデータベース ID
- データベース ID のコンマ区切りリスト
- ハイフン記号（-）で区切られたデータベース ID の範囲
- データベース ID のリストと範囲の組み合わせ
- すべてのデータベース ID を示すアスタリスク（*）

トピック：OPERATIONS

トピック OPERATIONS は 1 つ以上の OPERATIONS 定義を含んでいます。各エントリは、1 つ以上の PERMISSION 定義で使用でき、1 つ以上のユーティリティ処理を *operation_name* に割り当てます。

トピック OPERATIONS の構文は次のとおりです。

```
[OPERATIONS]
  <operation_definitions>
[OPERATIONS-END]
```

OPERATIONS トピックは 1 つ以上の OPERATIONS 定義を含むことができます。

項目：OPERATIONS 定義

OPERATION 定義は項目です。1 つ以上のユーティリティ処理を固有の名前に割り当てます。各 OPERATION 定義は、PERMISSION 定義で値として使用されます。

OBJECT 定義の名前は、一意にする必要があります。また、サブトピックの場合のようにカッコで囲む必要があります。

トピック OPERATION 定義の構文は次のとおりです。

```
<operation_name> = <utility_name [, <utility_name>]>
```

operation_name の値は次のいずれかになります。

- Adabas ユーティリティの名前
- Adabas ユーティリティの名前のコンマ区切りリスト

有効な Adabas ユーティリティ名を以下に示します。

値	説明
ada.uti.bck	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
ada.uti.dbm	データベース更新
ada.uti.ela	Adabas Analytics の構成
ada.uti.fdu	ファイル定義
ada.uti.frm	新規データベースのフォーマットと作成
ada.uti.opr	オペレータユーティリティ
ada.uti.ord	データベースのリオーダまたはファイルのエクスポート／インポート
ada.uti.rba	RBAC 定義の管理
ada.uti.rec	データベースまたはファイルのリカバリ
ada.uti.rep	データベースレポート
ada.uti.scr	セキュリティ機能の管理および有効化
ada.uti.uld	ファイルのアンロード

注意: *utility_name* の値では大文字と小文字が区別されます。無効なエントリ（無効な権限など）は処理エラーになります。

制限および制約事項

adarbac.ini ファイルのエントリには、次の制限や制約事項が適用されます。

説明	制限／制約事項
行	最大 2064 文字までです。
値	最大 2036 文字までです。
USER の名前	最大 128 文字までです。
ROLE の名前	最大 32 文字までです。
PERMISSION の名前	最大 19 文字までです。
OPERATION の名前	最大 128 文字までです。
OBJECT の名前	最大 19 文字までです。
ファイルパス	最大 255 文字までです。
ファイル名	最大 32 文字までです。
値、空白を含む値	値は引用符（”）で囲む必要があります。
値、値のリスト	複数の値はコンマ（,）で区切ります。
値、値の範囲	アスタリスク文字（*）はすべての有効な値の範囲を示します。この文字の使用は、USER_ROLE および DBID エントリに制限されます。
行、継続文字	使用不可。エントリは 1 行の内容に制限されます。
コメント行	コメントはハッシュ記号（#）で始まります。
行コメント	コメントは最初の空白で始まります。

説明	制限／制約事項
重複するエントリ	以前のエントリ値は、重複するエントリによって上書きされます。最後のエントリ値が使用されます。

サンプル：adarbac.ini

サンプル：無制限アクセス（レガシー）

以下に示すセキュリティ定義では、無制限のアクセスが実装されます。

```
###
###  SAMPLE:   Access Permissions (Unrestricted Access)
###
###  Users:   Generic Definition
###
###  Roles:   DBADMIN          Administrator (Database)
###           FILEADMIN        Administrator (File)
###           DBREPORT         Reporting     (Database)
###           USER             User          (Database)
###
###  Role Assignment:
###           All users are assigned all roles
###           which provides the user with the permissions
###           to execute operations as was in previous product releases
###
###

### Generic definition of Users and the Assignment of Roles
[USER_ROLE]
    * = DBADMIN,FILEADMIN,REPORTER,USER
[USER_ROLE-END]

### Assignment of Permissions to Roles
[ROLE_PERMISSION]
    DBADMIN = DBADMIN_PERM
    FILEADMIN = FILEADMIN_PERM
    REPORTER = REPORTER_PERM
    USER = USER_PERM
[ROLE_PERMISSION-END]

### Definition of Permissions - Tuples of Operations on Objects
[PERMISSIONS]
 [DBADMIN_PERM]
  OBJECT = OBJECT_ANY
  OPERATION = ALL_OPERATIONS
 [DBADMIN_PERM-END]
 [FILEADMIN_PERM]
```

```
  OBJECT = OBJECT_ANY
  OPERATION = FILE_OPERATIONS
 [FILEADMIN_PERM-END]
 [REPORTER_PERM]
  OBJECT = OBJECT_ANY
  OPERATION = REPORT_OPERATIONS
 [REPORTER_PERM-END]
 [USER_PERM]
  OBJECT = OBJECT_ANY
  OPERATION = USER_OPERATIONS
 [USER_PERM-END]
[PERMISSIONS-END]

### Definition of Objects
[OBJECTS]
 [OBJECT_ANY]
  DBID = *
 [OBJECT_ANY-END]
[OBJECTS-END]

### Definition of Operations
[OPERATIONS]
  ALL_OPERATIONS = ↩
ada.uti.bck,ada.uti.dbm,ada.uti.ela,ada.uti.fdu,ada.uti.opr,ada.uti.ord,ada.uti.rba,ada.uti.rec,ada.uti.rep,ada.uti.scr,ada.uti.uld

  FILE_OPERATIONS = ada.uti.fdu,ada.uti.ord,ada.uti.uld
  REPORT_OPERATIONS = ada.uti.rep
  USER_OPERATIONS = ada.uti.none
[OPERATIONS-END]

### EOF
```

サンプル：RBAC セキュリティ

以下に示すセキュリティ定義では、最小限の権限が実装されます。

```
###
###  SAMPLE:  RBAC Security (LEAST AMOUNT OF PRIVILEGES)
###
###  Users:   Explicitly Defined
###
###  Roles:   DBADMIN          Administrator (Database)
###           FILEADMIN        Administrator (File)
###           DBREPORT         Reporting     (Database)
###           USER             User          (Database)
###
###  Role Assignment:
###          Each user is explicitly defined and is assigned the minimum of roles ↩
and privileges.
###           This enables the implementation of a "Least amount of Privileges" ↩
RBAC Security model.
```

```
###

### Explicit definition of Users and the Assignment of Roles
[USER_ROLE]
   USRDBADMIN   = DBADMIN
   USRFILEADMIN = FILEADMIN
   USRDBREPORT  = DBREPORT
   USR0001      = USER
[USER_ROLE-END]

### Assignment of Permissions to Roles
[ROLE_PERMISSION]
   DBADMIN = DBADMIN_PERM
   FILEADMIN = FILEADMIN_PERM
   REPORTER = REPORTER_PERM
   USER = USER_PERM
[ROLE_PERMISSION-END]

### Definition of Permissions - Tuples of Operations on Objects
[PERMISSIONS]
  [DBADMIN_PERM]
    OBJECT = OBJECT_ANY
    OPERATION = ALL_OPERATIONS
  [DBADMIN_PERM-END]

  [FILEADMIN_PERM]
    OBJECT = OBJECT_ANY
    OPERATION = FILE_OPERATIONS
  [FILEADMIN_PERM-END]

  [DBREPORT_PERM]
    OBJECT = OBJECT_ANY
    OPERATION = REPORT_OPERATIONS
  [DBREPORT_PERM-END]

  [USER_PERM]
    OBJECT = OBJECT_ANY
    OPERATION = USER_OPERATIONS
  [USER_PERM-END]
[PERMISSIONS-END]

### Definition of Objects
[OBJECTS]
  [OBJECT_ANY]
    DBID = *
  [OBJECT_ANY-END]
[OBJECTS-END]

### Definition of Operations
[OPERATIONS]
  ALL_OPERATIONS = ↩
ada.uti.bck,ada.uti.dbm,ada.uti.ela,ada.uti.fdu,ada.uti.cpr,ada.uti.ord,ada.uti.rba,ada.uti.rec,ada.uti.rep,ada.uti.scr,ada.uti.uld
```

```
  FILE_OPERATIONS = ada.uti.fdu,ada.uti.ord,ada.uti.uld
  REPORT_OPERATIONS = ada.uti.rep
  USER_OPERATIONS = ada.uti.none
[OPERATIONS-END]

### EOF
```